A Hybrid Model Based on Samples Difficulty for Imbalanced Data Classification

Ao Shan(✉) and Yeh-Ching Chung

The Chinese University of Hong Kong (Shenzhen), Shenzhen 518172, China
18840824080@163.com

Abstract. Imbalanced data classification is a challenging problem with wide applications in machine learning and data mining. Most researchers attempt to solve this problem from the data level or algorithm level. Nevertheless, these methods have their limitations. In addition, most of them focus on dealing with the imbalance in the number of data samples while ignoring the imbalance caused by sample difficulty. Thus, we design a hybrid model to handle this problem. Our model integrates data space improvement, sample selection, sampling strategy, and loss function. To evaluate the performance of our hybrid model, we conduct experiments on several real-world imbalanced datasets. The experimental results prove that our hybrid model is effective.

Keywords: Class imbalance · Machine learning · Imbalanced data

1 Introduction

Imbalanced data classification is challenging [10,13], and it has wide applications in the machine learning field [3,11,19]. The main characteristic of the imbalanced data is its skewed data distribution, which means that most samples belong to one class (the majority class) and the rest belong to the other (the minority class). The skewed data distribution usually leads to conventional machine learning classifiers having poor classification performance.

To address imbalanced data classification, researchers have proposed plenty of methods. Existing methods mainly contain two categories: data-level techniques and algorithm-level techniques. Data-level techniques solve the imbalanced data by changing the data distribution. Algorithm-level techniques increase the importance of the minority class in adjusting the learning or decision process.

However, we notice the weakness of the above existing methods. On the one hand, traditional data-level methods usually do not consider the impact of different types of samples in the imbalanced dataset to train the model. The study [16] indicates that some of the samples are useless and even negatively impact model training. On the other hand, traditional algorithm-level methods [6,8] usually focus on giving a higher loss to the minority class but ignore the impact of sample difficulty.

L. Iliadis et al. (Eds.): ICANN 2023, LNCS 14254, pp. 26–37, 2023.
https://doi.org/10.1007/978-3-031-44207-0_3

This paper aims to remedy the above weaknesses from two aspects. Firstly, this paper introduces the concept of "sample classification importance" to select suitable samples for sampling. Intuitively, classification importance represents the importance of a sample for classifier training. For a dataset, we divide all samples into three kinds, i.e., important informative samples, negative informative samples, and general informative samples. Such sample classification importance can guide the selection of suitable samples for sampling to obtain satisfactory results. Secondly, we propose a loss function that is based on sample difficulty. This loss function can give different costs to different samples according to their sample difficulty.

Then, we further propose a hybrid model to solve imbalanced data classification. Our model integrates data space improvement, sample selection, and loss function based on sample difficulty. Specifically, it contains three blocks: (1) Data space block, which transforms the data space to make samples close to their nearest neighbors belonging to the same class and separates samples from other classes by a large margin. This block can make samples easier to be separated. (2) Sample selection block finds suitable samples for sampling to obtain a balanced dataset. This block aims to find valuable samples. (3) Sample Difficulty block applies a novel loss function that adds larger loss to samples with greater difficulty for training the classifier.

In summary, our contributions lie in the following aspects. (1) Firstly, we propose a new sample selection approach that can use fewer samples but get better classification results. (2) Secondly, we design a novel loss function based on sample difficulty for imbalanced data training. (3) Thirdly, we design a hybrid model that integrates space improvement, sample selection, sampling, and loss function to handle this problem. (4) Finally, experimental results on real-world imbalanced datasets have shown that our hybrid model performs better than competing methods, and each block of our model is valid.

2 Related Work

2.1 Data-Level Methods

Data-level approaches [7] aim to solve imbalanced data by changing the data distribution. They can be further divided into undersampling methods and oversampling methods. Under-sampling methods reduce the number of majority instances from the original dataset to balance the dataset. The simplest undersampling form is random undersampling [10]. This method removes the majority of instances randomly. Unlike undersampling methods, oversampling methods generate minority instances to obtain a balanced dataset. Random oversampling is the most straightforward way that randomly generates minority instances from the original data. In addition, plenty of advanced sampling methods have been designed. SMOTE [5] is the commonly used sampling method that selects close instances, drawing a line between instances and generating a new instance at a point along that line. ADASYN [9], MWMOTE [1], and ADMO [18] are representative sampling methods that generate the minority synthetic instances.

However, the weaknesses of data-level methods are apparent: The technique of selecting suitable instances for sampling is still being determined [4].

2.2 Algorithm-Level Methods

Algorithm-level approaches solve imbalanced data by increasing the importance of the minority class in adjusting the learning or decision process. These methods mainly contain cost-sensitive learning and novel loss functions. Cost-sensitive learning approaches modify the cost matrix to reduce bias towards the majority class. However, determining a matrix is difficult for cost-sensitive learning-based methods. Researchers have recently designed several new loss functions [6,8] for training deep neural networks for solving imbalanced data classification. The most widely used loss for imbalanced data is the focal loss [15] that assigns a weight to each instance according to its prediction accuracy in model training.

3 Proposed Method

3.1 Overview

As shown in Fig.1, our model consists of three blocks: (1) Data space block (DSB), which transforms the data space to make samples close to their nearest neighbors with the same class. This block can make samples easier to be separated. (2) Sample selection block (SSB) finds valuable samples and builds up a set based on valuable samples. This block aims to find valuable samples for sampling. (3) Sample Difficulty block (SDB) applies a novel loss function that adds larger loss to samples with higher sample difficulty for the training classifier.

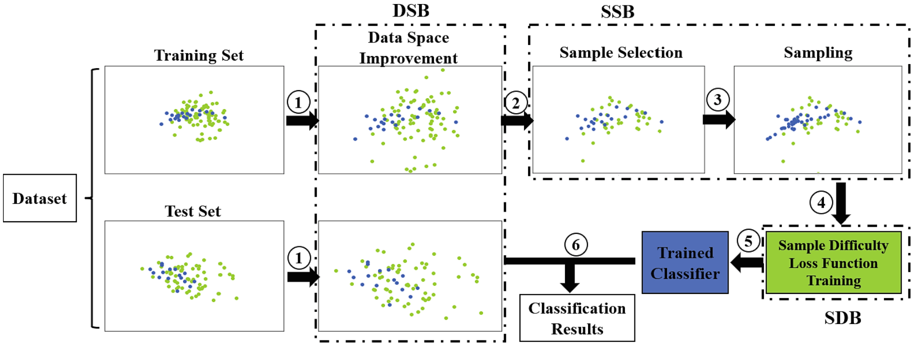

Fig. 1. The architecture of our hybrid model

3.2 Data Space Block

Our model integrates the data space improvement technique to make the imbalanced data easier to be separated. In this work, we use the LMNN [17] technique that builds up an algorithm to pull samples with the same class label close to the target sample and push samples that belong to different class labels away from the target sample, as shown in Fig. 2. The algorithm of the LMNN technique is as follows: $\varphi(H) = (1-\mu)\varphi_{\text{pull}}\,(H) + \mu\varphi_{\text{push}}\,(H)$, where H is the linear transformation of the input space and μ is a positive real number utilized as the weight. The first part of this loss penalizes large distances between the sample and its k nearest neighbors belonging to the same class, which is defined as $\varphi_{\text{pull}}\,(H) = \sum_{p,q\in M(p)} \left\|L\left(x_p - x_q\right)\right\|^2$, where $M(p)$ is the k nearest neighbor of sample p with the same class label as p.

The second part penalizes small distances between the sample and others with different classes, which is defined as:
$\varphi_{\text{push}}\,(H) \quad = \quad \sum_{p,q,l}\left(1-\delta_{pl}\right)\max\left\{1+\left\|H\left(s_p-s_q\right)\right\|^2-\left\|H\left(s_p-s_l\right)\right\|^2,0\right\}$, where δ_{il} is utilized to decide whether samples s_l and s_p belong to different classes or not. If samples belong to different classes, $\delta_{pl}=0$; otherwise, $\delta_{pl}=1$.

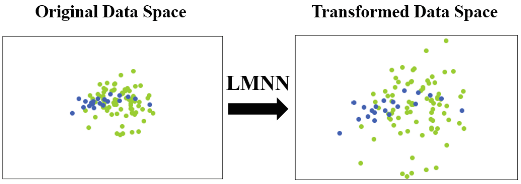

Fig. 2. Using the LMNN algorithm, the comparison between original data space and transformed data space

3.3 Sample Selection Block

Traditional data-level methods usually select all samples for sampling to obtain a balanced dataset. However, study [16] has indicated that not all samples are useful for model training. Thus, it is necessary to distinguish the types of samples and select suitable samples for sampling. In this part, we first introduce the definition of sample classification importance and propose a method to finish sample selection based on this definition.

Definition: Sample classification importance represents the importance of a sample for the classifier training.

Intuitively, we divide samples into three kinds, i.e., important informative samples, negative informative samples, and general informative samples, as shown in Fig. 3 .

Important informative samples: They are the most informative samples during the classifier training. For instance, as we can see in Fig.3, important informative instances are usually located close to the classification boundary of the classifier. Improving the importance of these instances is helpful in improving the performance of the classification [16].

Negative informative samples: By contrast, negative informative samples harm the model training. For example, negative informative samples are in Fig. 3 are usually caused by indistinguishable noise, which could lead the model to serious overfitting. Thus, we need to reduce the impact of these samples.

General informative samples: Most of the samples are general informative samples that the model can correctly classify, as shown in Fig.3. Each general informative sample only contributes minor importance. However, the overall contribution is enormous because of its large number. For this type of sample, we only need a small part of them to remain their " skeleton " to prevent overfitting, then remove most of them.

We evaluate sample classification importance based on the kNN method [2]. If all neighbors of a sample belong to a different class, then it is a negative informative sample. On the contrary, if all neighbors of a sample and itself belong to the same class, then it is a general informative sample. In other cases, the sample can be seen as an important informative sample, which means that it will have a large value when a sample locates on the borderline between different classes. Then, we introduce the sample selection method. Given a dataset, it can be divided into three parts: negative informative set, important informative set, and general informative set according to sample classification importance. We do not use negative informative samples to sample since they have negative impacts on the classifier training. We focus on sampling important informative samples because they are essential in finding the classification boundary. In addition, we only use small parts of general informative samples to sample because we only need a small part of them to retain their "skeleton". Based on the above analysis, our sample selection method is shown in Algorithm 1 in detail.

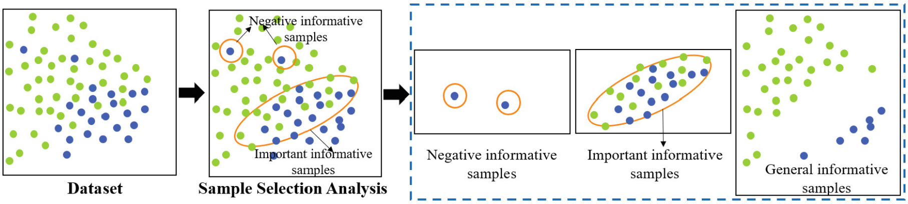
Dataset
Sample Selection Analysis
Negative informative samples
Important informative samples
Negative informative samples
Important informative samples
General informative samples

Fig. 3. Illustration of types of samples

3.4 Sample Difficulty Block

This block applies a new loss function based on sample difficulty to train the classifier with the imbalanced data. We first introduce the sample difficulty and then propose our loss function. Based on the analysis in the sample selection part, finding suitable samples that can learn the classification boundary as precisely as possible is important. In addition, we also notice that different suitable samples

Algorithm 1. Sample Selection

```
Input: Dataset $D$, the quantity of samples $N$, the parameter of kNN method $k$, the percent of general informative samples $m$.

for $i \leftarrow 1$ to $N$ do
    Use the kNN method to calculate the number of its neighbors that have different labels with itself: $kNN\left(x_{i,j}, D - D_j\right)$;
    if $kNN\left(x_{i,j}, D - D_j\right) = k$ then
        $x_{i,j}$ is a negative informative sample;
    else if $kNN\left(x_{i,j}, D - D_j\right) = 0$ then
        $x_{i,j}$ is a general informative sample;
        Add $x_{i,j}$ to the set of general informative samples $D_{general}$;
    else
        $x_{i,j}$ is an important informative sample;
        Add $x_{i,j}$ to the set of important informative samples $D_{important}$;
    end if
end for
Based on $D_{general}$, use random undersampling to obtain $m$ percent of general informative samples $D_{sampledgeneral}$.
$D_{selection} = D_{important} \cup D_{sampledgeneral}$
Output: The dataset after Sample Selection $D_{selection}$
```

may also have different difficulties in model training. Thus, we propose a method to calculate the level of sample difficulty.

Intuitively, a sample with more nearest neighbors with different class labels will have a high sample difficulty level. Based on this, we provide formula (1) to evaluate the sample difficulty (SD), where k is the number of nearest neighbors. $kNN(x_{i,j}, D - D_j)$ is the number of k nearest neighbors of sample $x_{i,j}$ that do not belong to class j.

$$\mathrm{SD}\left(x_{i,j}\right) = \frac{kNN\left(x_{i,j}, D - D_j\right)}{k} \tag{1}$$

Then, We introduce our novel loss starting from the cross-entropy (CE) loss for classification. For a classification of p categories, the CE loss is defined as:

$$L_{\mathrm{CE}} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p} y_{i,j} \log \hat{y}_{i,j} \tag{2}$$

where n is the sample size. $y_{i,j} \in \{1, 0\}$ specifies the ground truth sample, and $\hat{y}_{i,j} \in [0, 1]$ is the model's estimated probability for the sample with ground truth i, j.

Based on the CE loss, we add a factor that can consider the different types of samples in a dataset, as mentioned in the sample selection block. The parameter $w_{i,j}$ is related to the sample difficulty. We use formulas (1) and (3) to calculate the value of $w_{i,j}$. Then we define our sample difficulty loss function as formula

(4). We notice the property of our proposed loss function. The parameter $w_{i,j}$ gives samples that are more difficult to train a large loss.

$$w_{i,j} = \log(1 + SD(x_{i,j})) \tag{3}$$

$$L_{\mathrm{SD}} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p} w_{i,j} y_{i,j} \log \hat{y}_{i,j} \tag{4}$$

4 Experiments

4.1 Data Description and Compared Methods

We employ several real-world imbalanced datasets by imblearn toolbox [14] (These datasets are from UCI, LIBSVM, and KDD repository.) to test the performance of our hybrid model. These datasets have different characteristics in terms of the number of samples, IR (Imbalance Ratio), and features. The detailed information on datasets is shown in Table 1. Besides, we randomly split datasets into training sets (60%), valid sets (20%), and test sets (20%).

Table 1. Summary of imbalanced datasets

Datasets	Samples	Features	IR
optical-digits	5620	64	9.1
satimage	6435	36	9.3
pen-digits	10992	16	9.4
abalone	4177	10	9.7
sick-euthyroid	3163	42	9.8
spectrometer	531	93	11
isolet	7797	617	12
us-crime	1994	100	12
yeast-ml8	2417	103	13
scene	2407	294	13
thyroid-sick	3772	52	15
coil-2000	9822	85	16
arrhythmia	452	278	17
oil	937	49	22
car-eval-4	1728	21	26
wine-quality	4898	11	26
abalone-19	4177	10	130

We compare our hybrid model with the following methods, including data-level methods: Random oversampling (ROS), MWMOTE [1], ADASYN [9], SMOTE [5], and AMDO [18]; algorithm-level methods: Focal loss [15], Class-balanced loss [6], and DWE loss [8].

4.2 Evaluation Metrics

We employ commonly used metrics, G-mean and AUC [12], to evaluate the performance of imbalanced data classification. Let FN, FP, TP, and TN be false negative, false positive, true positive, and true negative. TNR and TPR measure the number of correctly classified positive instances and negative instances, respectively. G-mean combines TNR and TPR . AUC is the area under the receiver operating characteristic curve that reflects the relationship between the false positive and true positive ratios. This area describes the trade-off between incorrectly classified positive and correctly classified negative instances.

$$TNR = \frac{TN}{TN + FP} \tag{5}$$

$$TPR = \frac{TP}{TP + FN} \tag{6}$$

$$G - \text{ Mean } = \sqrt{TPR \times TNR} \tag{7}$$

4.3 Implementation Details

We select Multilayer perception (MLP) as the classifier and a batch size of 32 to train it for 100 epochs based on the TensorFlow framework. The classifier utilizes Adam as the optimizer, with a learning rate is 0.001. We ran all experiments ten times and took the average of ten times as the final result to obtain a reliable result. Our model finds suitable samples and evaluates the sample difficulty level based on the kNN method (k = 7).

4.4 Experimental Results

Tables 2 and 3 reports AUC and G-mean values on imbalanced datasets. From the experimental results, we find that no single method can achieve the best performance on all datasets. In contrast, our hybrid model achieves decent performance in most cases. The reasons that our model can perform well lie in the following aspects.

First, we use a data space block to make samples easier to be classified. Second, unlike traditional imbalance resolution methods, we select suitable samples based on sample selection for model training. This method retains the critical classification information. Third, our sample difficulty loss function gives each sample a loss corresponding to its sample difficulty. This loss function fully considers the impact of sample difficulty and offers a higher loss to the samples with higher sample difficulty and more challenging to distinguish. Combining the findings above, our model is effective for imbalanced data classification.

Table 2. Valus of AUC on 17 real-world imbalanced datasets

Dataset	MWMOTE	ADASYN	SMOTE	AMDO	ROS	Focal	DWE	BCE	Our model
optical-digits	0.9792	0.9772	0.9810	0.9747	0.9861	0.5000	0.5000	0.9826	**0.9940**
satimage	0.7931	0.7946	0.7985	0.5302	0.8060	0.5000	0.5000	0.7964	**0.8437**
pen-digits	0.9951	0.9963	0.9977	0.9956	0.9985	0.5000	0.5000	0.9952	**0.9985**
abalone	0.7206	**0.7389**	0.7122	0.4990	0.7362	0.5048	0.5416	0.6504	0.6700
sick-euthyroid	0.9224	**0.9404**	0.8988	0.9006	0.9201	0.5000	0.5000	0.9283	0.9092
spectrometer	**0.9948**	0.9928	0.9726	0.9231	0.9574	0.5000	0.5000	0.9716	0.9776
isolet	0.9637	0.9581	0.9480	0.9621	0.9734	0.6846	0.5000	0.9617	**0.9937**
us-crime	0.6874	0.6685	0.6527	0.6870	0.6905	0.6555	0.6858	0.6947	**0.8011**
yeast-ml8	0.5193	0.5126	0.5196	0.4964	0.5126	0.5136	0.5102	0.5195	**0.5916**
scene	0.5924	0.6036	0.5850	0.5658	0.5841	0.5767	0.5000	0.5809	**0.7827**
thyroid-sick	0.9098	0.8941	0.8831	0.8536	0.8968	0.5000	0.5000	0.9098	**0.9102**
coil-2000	0.5572	0.5492	0.5641	0.5252	0.5581	0.5290	0.5331	0.5787	**0.5923**
arrhythmia	0.6101	0.6112	0.6100	0.6066	0.6089	0.5000	0.5000	0.6712	**0.9965**
oil	0.6799	0.8132	0.6443	0.6367	0.6028	0.5000	0.5000	0.7282	**0.8475**
car-eval-4	0.9072	0.9267	0.9170	0.9725	0.9470	0.9079	0.9023	0.8970	**0.9880**
wine-quality	0.6512	0.6522	**0.6819**	0.5438	0.6781	0.5000	0.5000	0.6532	0.6542
abalone-19	0.4892	0.5018	0.4896	0.4996	0.4898	0.4995	0.5000	0.5018	**0.5818**

Table 3. Valus of G-mean on 17 real-world imbalanced datasets

Dataset	MWMOTE	ADASYN	SMOTE	AMDO	ROS	Focal	DWE	BCE	Our model
optical-digits	0.9788	0.9769	0.9807	0.9743	0.9860	0.0000	0.0000	0.9825	**0.9940**
satimage	0.7893	0.7892	0.7968	0.2582	0.8032	0.0000	0.0000	0.7936	**0.8425**
pen-digits	0.9951	0.9963	0.9977	0.9956	0.9985	0.0000	0.0000	0.9952	**0.9985**
abalone	0.7080	**0.7347**	0.7029	0.0000	0.7292	0.0605	0.2951	0.6129	0.6338
sick-euthyroid	0.9221	**0.9403**	0.8953	0.8980	0.9185	0.0000	0.0000	0.9281	0.8952
spectrometer	**0.9948**	0.9928	0.9721	0.9120	0.9558	0.0000	0.9372	0.9710	0.9445
isolet	0.9632	0.9574	0.9451	0.9615	0.9732	0.3844	0.0000	0.9612	**0.9937**
us-crime	0.6247	0.5936	0.5639	0.6118	0.6284	0.5646	0.6176	0.6375	**0.7849**
yeast-ml8	0.2287	0.2254	0.2412	0.0755	0.2302	0.1884	0.1939	0.2830	**0.4853**
scene	0.4736	0.4964	0.4527	0.3955	0.4483	0.4323	0.0000	0.4511	**0.7652**
thyroid-sick	0.9098	0.8904	0.8786	0.8436	0.8930	0.0000	0.0000	0.9072	**0.9089**
coil-2000	0.3921	0.3770	0.4210	0.2545	0.4098	0.2623	0.2994	0.4637	**0.5298**
arrhythmia	0.4924	0.4930	0.4924	0.4907	0.4918	0.0000	0.0000	0.6055	**0.9965**
oil	0.5201	0.8091	0.4144	0.3503	0.3683	0.0000	0.0000	0.7282	**0.8383**
car-eval-4	0.9031	0.9439	0.9212	0.9087	0.9236	0.9026	0.8954	0.8917	**0.9880**
wine-quality	0.6070	0.6059	**0.6454**	0.3003	0.6390	0.0000	0.0000	0.5883	0.6365
abalone-19	0.0000	0.0696	0.0000	0.0000	0.0000	0.0000	0.0000	0.1380	**0.4926**

5 Discussion

5.1 The Impact of Important Informative Samples

In our model, we select suitable samples to train the classifier because samples are essential for finding the classification boundary. Thus, we run experiments on both original and datasets that only contain important informative samples to further illustrate the impact of important informative samples. From Table 4, we observe that training the classifier with datasets containing only important informative samples can obtain better results than training the classifier with original datasets, which verifies the effectiveness of important informative samples. In addition, we also noticed that by selecting suitable samples for training, we improved the classification results while reducing the number of samples used for model training. In summary, selecting suitable samples to deal with imbalanced data classification is a new perspective, which can both reduce the number of samples used for the classifier training and improve the performance of the classifier.

5.2 The Impact of Parameters

To analyze the impact of parameter k in our model, we conduct experiments with varying k from 1 to 13 on three real-world imbalanced datasets. From the experimental results in Fig. 4, we find that the performance of our model is stable with the change of k and when $k = 7$ achieves the best performance.

Table 4. The Impact of Important Informative Samples

Dataset	Original Samples		Suitable Samples	
	AUC	G-mean	AUC	G-mean
pen-digits	0.9976	0.9976	0.9985	0.9985
abalone	0.6561	0.6207	0.6700	0.6338
yeast-ml8	0.5330	0.2956	0.5916	0.4853

5.3 Ablation Study

Our model consists of three blocks: Data Space Block (DSB), Sample Selection Block (SSB), and Sample Difficulty Block (SDB). To analyze the effectiveness of each block, we build some variants of our hybrid model: (1) DSB, which is our model without DSB; (2) SSB, which is our model without SSB; (3) SDB, which is our model without SDB. Fig. 5 shows experimental results on abalone-19 and us-crime datasets. We find that all of these variants perform worse than our model on both datasets, which illustrates that our model effectively integrates three blocks to take advantage of each. Moreover, we find that SSB performs the worst, which demonstrates that SSB has a more critical impact among all blocks.

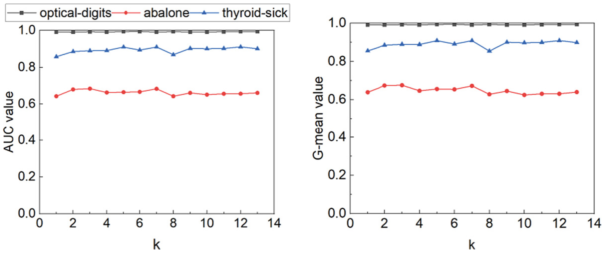

Fig. 4. Impact of parameter k in our model

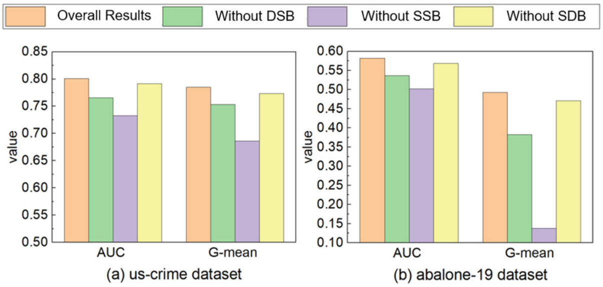

Fig. 5. Ablation Study

6 Conclusion

We aim to overcome the weakness of existing imbalanced learning methods from perspectives of sample selection and sample difficulty. First, we divide samples into different types in an imbalanced dataset according to their impacts on imbalanced data classification. Based on this, we can select suitable samples for sampling. Then, we propose a loss function based on sample difficulty. After that, we design a hybrid model to solve imbalanced data classification. To the best of our knowledge, this is the first model that integrates data space improvement, sample selection, and loss function into imbalanced data classification. Experiments on real-world imbalanced datasets have shown that our hybrid model performs better than competing methods. The ablation study verifies that each model block is valid.

References

1. Barua, S., Islam, M.M., Yao, X., Murase, K.: Mwmote-majority weighted minority oversampling technique for imbalanced data set learning. IEEE Trans. Knowl. Data Eng. **26**(2), 405–425 (2012)
2. Borsos, Z., Lemnaru, C., Potolea, R.: Dealing with overlap and imbalance: a new metric and approach. Pattern Anal. Appl. **21**(2), 381–395 (2018)

3. Bugnon, L.A., Yones, C., Milone, D.H., Stegmayer, G.: Deep neural architectures for highly imbalanced data in bioinformatics. IEEE Trans. Neural Netw. Learn. Syst. **31**(8), 2857–2867 (2019)
4. Cao, P., Zhao, D., Zaïane, O.R.: A PSO-based cost-sensitive neural network for imbalanced data classification. In: Li, J., et al. (eds.) PAKDD 2013. LNCS (LNAI), vol. 7867, pp. 452–463. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40319-4_39
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. J. Artifi. Intell. Res. **16**, 321–357 (2002)
6. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9268–9277 (2019)
7. Das, B., Krishnan, N.C., Cook, D.J.: Racog and wracog: two probabilistic oversampling techniques. IEEE Trans. Knowl. Data Eng. **27**(1), 222–234 (2014)
8. Fernando, K.R.M., Tsokos, C.P.: Dynamically weighted balanced loss: class imbalanced learning and confidence calibration of deep neural networks. IEEE Trans. Neural Netw. Learn. Syst. (2021)
9. He, H., Bai, Y., Garcia, E.A., Li, S.: Adasyn: adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322–1328. IEEE (2008)
10. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**(9), 1263–1284 (2009)
11. Hu, Y., Zhang, Y., Gong, D., Sun, X.: Multi-participant federated feature selection algorithm with particle swarm optimizaiton for imbalanced data under privacy protection. IEEE Trans. Artifi. Intell. (2022)
12. Johnson, J.M., Khoshgoftaar, T.M.: Survey on deep learning with class imbalance. J. Big Data **6**(1), 1–54 (2019)
13. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Progress Artifi. Intell. **5**(4), 221–232 (2016)
14. Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning. J. Mach. Learn. Res. **18**(1), 559–563 (2017)
15. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
16. Liu, Z., et al.: Self-paced ensemble for highly imbalanced massive data classification. In: 2020 IEEE 36th International Conference on Data Engineering (ICDE), pp. 841–852. IEEE (2020)
17. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. **10**(2) (2009)
18. Yang, X., Kuang, Q., Zhang, W., Zhang, G.: Amdo: an over-sampling technique for multi-class imbalanced problems. IEEE Trans. Knowl. Data Eng. **30**(9), 1672–1685 (2017)
19. Zhao, H., Wang, R., Lei, Y., Liao, W.H., Cao, H., Cao, J.: Severity level diagnosis of parkinson's disease by ensemble k-nearest neighbor under imbalanced data. Expert Syst. Appli. **189**, 116113 (2022)