Expert Systems With Applications 213 (2023) 118878

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Benchmarking state-of-the-art imbalanced data learning approaches for credit scoring

Cuiqing Jiang[a,b], Wang Lu[a], Zhao Wang[a,*], Yong Ding[a]

[a] Schoolof Management, Hefei University of Technology, Hefei, Anhui 230009, China
[b] KeyLaboratory of Process Optimization and Intelligent Decision-Making, Ministry of Education, Hefei, Anhui 230009, China

ARTICLE INFO

Keywords:
Credit scoring
Imbalanced classification
Predicting benchmark
GAN

ABSTRACT

The goal of credit scoring is to identify abnormalities, aiding decision making and maintaining the order of financial transactions. Due to the small number of default records, one inevitably faces a class imbalance problem when handling financial data. The class imbalance problem has received a lot of attention because of the economic loss that can occur when one fails to accurately identify default samples. To solve this problem, there are various classic and mature approaches to learning imbalanced data, including resampling approaches, cost-sensitive strategies, and so on. Especially in recent years, generative adversarial networks (GANs) have attracted the attention of researchers to explore these networks' effects as imbalanced data learning tools. However, no attention has been paid to the systematic scoring and comparison of these traditional and state-of-the-art imbalanced data learning approaches in relation to credit scoring. Therefore, choosing several related datasets, we compare the performance of the traditional approaches and GANs in solving the class imbalance problem of credit scoring; at the same time, with the help of benchmark analysis, we provide some suggestions for relevant research.

1. Introduction

Credit risk is the risk of default on a debt that arises from a borrower failing to make required payments. With the occurrence of the subprime mortgage crisis and European debt crisis, credit risk has become more and more important. Credit scoring (Abdou and Pointon, 2011) is the most widely used model-related scoring tool to mitigate credit risk, of which the essence is a classification task. Credit scoring is an important part of maintaining a stable trading environment; a better model is more able to limit the occurrence of defaults and reduce banks' economic losses.

However, the predicting models can easily face the challenge of a class imbalance (CI) problem (He and Garcia, 2009) due to the existence of very few records of default. In the binary CI problem of credit scoring, the majority (non-default) class refers to the class with the large number of samples, compared to the minority (default) class. This differs from the traditional classification problem, which is based on the assumption that the number of training samples in different classes varies very little. The CI problem will make the model produce misleading estimates, biasing the prediction results toward the majority class and failing to accurately identify the minority class samples. At the same time, the performance measures of some traditional classification problems become inapplicable in a CI problem case; for example, if the classifier misjudges a default record as a normal record, even though it gives a high total prediction accuracy, it will also increase the probability of a person defaulting, avoiding repayment, and causing huge losses for the bank.

To solve the CI problem, there are various classic and mature imbalanced data learning approaches (IDLAs): Data-level approaches change the number of samples between different classes by some means to eliminate the CI problem (such as random sampling methods and SMOTEs (Fernández et al., 2018). Algorithm-level approaches make adjustments from the perspective of the model to overcome the deviation of outputs (like cost-sensitive processing). Especially in recent years, generative adversarial networks (GANs) have attracted researchers' attention to handle the CI problem as a sample-synthesizing tool that is similar to SMOTE (Goodfellow et al., 2014; Fiore et al., 2019).

However, there is still a lack of systematic sorting and comparison of the effects of IDLAs in credit scoring. On the one hand, many researchers are committed to comparing the effects of different classification algorithms in credit scoring (Lessmann et al., 2015; Louzada et al., 2016;

* Corresponding author.
E-mail addresses: jiangcuiq@163.com (C. Jiang), winkidslu@gmail.com (W. Lu), xcwangzhao@163.com (Z. Wang), dingyong@hfut.edu.cn (Y. Ding).

https://doi.org/10.1016/j.eswa.2022.118878
Received 30 March 2022; Received in revised form 26 August 2022; Accepted 18 September 2022
Available online 12 October 2022

Brown and Mues, 2012), but there are few comprehensive comparisons of the effects of IDLAs. Also, as for relevant studies, the considered IDLAs are relatively single, of which only resampling methods represented by SMOTEs are widely used. For example, by setting a different imbalanced ratio, Marqués et al. (2013) only compare the performance of resampling approaches based on random sampling (ROS and RUS, defined below) and SMOTEs; similarly, CHEN et al. (2017) select resampling approaches and SMOTEs for experiments, and discuss the applicable conditions of the two types of approaches. In more detail, López et al. (2013) mainly analyze the performance of data-level approaches against algorithm-level approaches, among which cost-sensitive approaches and hybrid approaches are the focus of observation.

On the other hand, GANs have made progress in applications of computer vision such as image generation, style transformation, text-to-image and so on. Different from images in which the pixel values all range from 0 to 255, "tabular data" (rows for records and columns for features) (Lei et al., 2020) has the characteristics of a large amount, a high dimension, and various types of features. Although some scholars have begun to use GANs in the learning and generation of tabular data with success (Fiore et al., 2019; Douzas and Bacao, 2018; Zheng et al., 2020), whether or not GANs can stably cope with such data needs to be verified through experiments.

To make up for the research deficiencies mentioned above, in this paper, we provide a systematic combing and comparison of IDLAs in credit evaluation. First, according to the principle of related IDLAs, we propose a new taxonomy that is intuitive and easily understood (shown in Fig. 1). Second, we select 28 typical IDLAs in total to cover each category in the taxonomy as comprehensively as possible, which have simple principles but represent the processing ideas of solving the CI problem. Third, our work considers the GAN in the benchmark analysis of IDLAs. We choose four representative GAN structures (GAN, CGAN, WGAN, and WGAN-GP, defined below) that can be classified by whether they include additional information in the training process. While introducing the principles of these four GAN structures, we also elaborate on the training process of GAN in processing tabular data. The main purpose of considering GANs is to test its stability and robustness as an IDLA in the scenario of credit scoring and to analyze and provide suggestions for its use. Fourth, we test the statistical significance of the differences among the different approaches and provide some tools for relevant research.

In the following sections, we first summarize the related studies in Section 2. The fundamentals of the approaches used in the benchmark analysis are elaborated on in Section 3. Next, the whole procedure of the experiment, including data pre-processing, analyzing the benchmarking, and discussing the results, is presented in Section 4 and Section 5. Finally, we conclude the paper in Section 6.

2. Background and related work

We spot two streams of previous studies for handling CI problem in credit scoring: one stream of study focuses on proposing new methods to solve CI problem in credit scoring, the other stream of study focuses on benchmarking imbalanced data learning approaches in credit scoring. We expound the background and related work from these two streams.

2.1. Imbalanced data learning methods in credit scoring

From the analysis of various imbalanced data learning frameworks in credit scoring, we find that machine learning models, especially ensemble learning models, have an irreplaceable position in solving CI problem of credit scoring. Several researchers embed imbalanced data processing strategies into ensemble learning models and make increase of credit scoring models' predicting accuracy. For example, Yao et al. (2022) and Niu et al. (2020) apply resampling technique to the training process of ensemble learning models, in their studies, each base classifier is trained with a rebalanced subset. Their frameworks achieve ideal results in imbalanced credit scoring datasets; Zhang and Chi (2021) and Junior et al. (2020) implement adaptively selection method for base classifier according to the scale of imbalanced credit scoring datasets in ensemble models; Carta et al. (2020) design an entropy-based sample selection framework to solve the problem of data insufficient and imbalanced in consumer credit; Mushava and Murray (2022) and Xiao et al. (2020) consider cost-sensitive in ensemble learning. By setting penalty rule for misclassification, models can concentrate on default samples and output unbiased predictions. He et al. (2018) improves a traditional ensemble learning framework by using random forest and extreme gradient boosting method as base classifier and optimize their parameters with particle swarm optimization for best performance.

With the increase of data size and feature dimension in credit scoring datasets, deep learning techniques provide a new solution to handling CI problem. Especially generative adversarial network (GAN), which is first applied by Fiore et al. (2019) as data generating tool in imbalanced

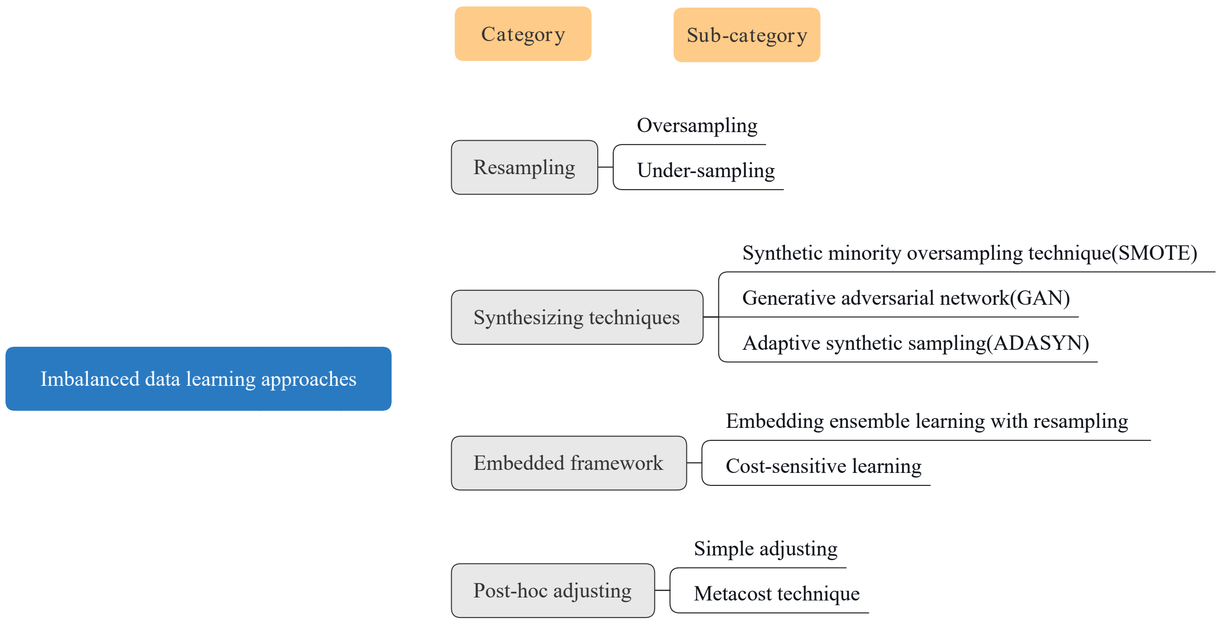
Category
Sub-category
Imbalanced data learning approaches
Resampling
Oversampling
Under-sampling
Synthesizing techniques
Synthetic minority oversampling technique(SMOTE)
Generative adversarial network(GAN)
Adaptive synthetic sampling(ADASYN)
Embedded framework
Embedding ensemble learning with resampling
Cost-sensitive learning
Post-hoc adjusting
Simple adjusting
Metacost technique

Fig. 1. Taxonomy of imbalanced data learning approaches.

credit scoring dataset, gradually gain popularity. Compared with SMOTE's local line segment mechanism, GAN can capture data distribution from a holistic perspective and generate new samples to rebalance credit scoring datasets. Since then, large number of relevant studies are proposed. With numerous credit card transactions, CI problem is common in area of credit card fraud detection owing to the rare of fraud transaction records. Wang and Yao (2022) and Gangwar and Ravi (2019) apply GAN as oversampling method to rebalance dataset and help build robust detection system; Zhu et al. (2022) propose a hybrid GAN-based framework in which a weighted under-sampling method is used to remove the overlapped or noise samples generated by GAN, result shows that the hybrid method perform well in customer credits scoring. Also in consumer credit scoring, Kang et al. (2022) propose a multi-task credit scoring model based on CWGAN-GP, which is adding labels in WGAN-GP model (describe in Section 3) to make the process of sampling generation more controllable, similar, Engelmann and Lessmann (2021) use GAN with label information and find that GAN-based oversampling outperform SMOTE across multiple real-word credit scoring datasets. Lei et al. (2020) designs a generative adversarial fusion network (IGAFN) to preprocess multi-source heterogeneous credit data and generate default samples, the proposed framework has significant increase of predicting accuracy than other machine learning of deep learning models.

According to these previous studies, we find that in handling CI problem of credit scoring, machine learning models still play an important and deep learning models represented by GANs are gradually popular and attract a lot of attention. However, most studies only focus on single method, lacking analysis of multiple imbalanced data learning approaches from a comprehensive perspective, which is meaningful for practical application in credit scoring.

Table 1
Studies related to benchmark analysis of IDLAs in credit scoring.

Related work	Number of datasets	Considered IDLAs					Metrics*	Contributions/Conclusions (imbalanced related) **
		Number of ILDAs	Resampling	Ensemble	Cost-sensitive	Others		
Aswathi et al. (2022)	1	5	√			√	Accuracy, F1, Precision, MCC	1. Applying SMOTE and SMOTE + Tomek produce better results than SMOTE + ENN 2 Majority voting method perform well in handling CI problem.
Lenka et al. (2022)	3	4	√	√			Accuracy, AUC, F1, BS, KS	SMOTE based methods make improvements in most of classifiers
Xiao et al. (2021)	6	10	√				AUC, F1, TPR, G-mean	1. RUS and SMOTE with ENN present best performance in imbalanced processing. 2 RUS + ensemble classifiers are optimal combinations in credit scoring.
Ounacer et al. (2020)	1	3	√				Accuracy, F1, Precision, Recall	Resampling methods perform well in handling CI problem
Rangel-Díaz-de-la Vega et al. (2020)	13	16	√				ACU, TPR, TNR	1. RUS has bad performance when imbalanced ratio is greater than 4.0. 2 Neighbor cleaning-based methods performs well in datasets with moderate; 3 Under-sampling based clustering have bad behavior in handling financial data; 4 SMOTEs has a stable effect imbalance ratio (less than 4.0);
de Melo Junior et al. (2019)	3	8		√	√		AUC	1. Ensemble methods perform well in all imbalanced level. 2 Grid search method helps increase predicting power in imbalanced datasets
Sanabila and Jatmiko (2018)	1	2	√				F1, Precision, Recall, Specificity	Combination of SMOTE and ENN increase classification performance and avoid CI problem.
CHEN et al. (2017)	3	5	√				AUC, F1, TPR, G-mean	1. Resampling methods perform well, and the degree of improvement depend on metrics selection. 2 SMOTE performs best with different classifiers.
Marqués et al. (2013)	5	8	√				AUC	1. Resampling methods consistently improve performance. 2 Over-sampling perform better than any under-sampling approaches.
Brown and Mues (2012)	5	1	√				AUC	1. Random Forest and gradient boosting classifier can well handle CI problem in credit scoring. 2 C4.5 decision tree quadratic discriminant analysis and k-nearest neighbors perform not well when facing with large class imbalance datasets.
García et al. (2012)	5	8	√				AUC	1. Resampling methods perform well in handling CI problem of credit scoring. 2 Random oversampling, one-side selection method and under-sampling based clustering perform worst in considered IDLAs; 3 The performance of random under-sampling, SMOTE-based methods and neighbor cleaning methods are not significantly different.
Crone and Finlay (2012)	2	2	√				GINI coefficient	Sample number recommendation conducted form resampling methods are given to make significant increase in accuracy across algorithms.

*Acronym of metrics: F1-score(F1), Matthews Correlation Coefficient (MCC), True Positive Rate (TPR), Kolmogorov-Smirnov statistic (KS), True Negative Rate (TNR).
**Acronym of IDLAs: Random Under-sampling (RUS), Edited Nearest Neighbors (ENN), Synthetic minority over-sampling technique (SMOTE).

2.2. Benchmark studies for imbalanced data learning methods in credit scoring

Table 1 shows the benchmark of imbalanced data learning approaches in credit scoring, several conclusions can be drawn: First, although CI problem is common and inevitable, less work focusing on analyzing the effects and selection of IDLAs in credit scoring compared with that of classifiers (Lessmann et al., 2015; de Melo Junior et al., 2019; Brown and Mues, 2012; Dastile et al., 2020; Louzada et al., 2016).

Second, there is a lack of systematic sorting and comparison of the effects of IDLAs in credit scoring. On the one hand, the types of IDLAs considered in the existing work are relatively single, of which resampling methods represented by SMOTEs are widely used (Aswathi et al., 2022; Lenka et al., 2022; Sanabila and Jatmiko, 2018). Based on the effect of resampling methods, some studies further explore the relationship between the degree of sample balance and improvement of models' prediction performance. For example, Rangel-Díaz-de-la Vega et al. (2020) find that there is a tradeoff of imbalanced ratio (IR) in original dataset that significantly impact the performance of resampling method, Crone and Finlay (2012) obtain the recommended sample size of dataset after using resampling methods to make a significant improvement in accuracy. However, other IDLAs with stable and ideal performance are not considered in relevant studies, like embedded methods (Yao et al., 2022; He et al., 2022) or cost-sensitive methods.

On the other hand, the state-of-the-art IDLAs occur and attract the attention of researchers, which should also be considered in the benchmark analysis. For example, among them, the generative adversarial network (GAN) make great progress in credit scoring scenario after it is booming in the field of Computer Vision (CV). Fiore et al. (2019) first use GAN in credit scoring and find that it has similar effect in handling CI problem with the classical SMOTE, since then, several works examine in detail the effects of various GANs in credit scoring (shown in above subsection), making GANs a research hotspot in handling tabular data like credit scoring datasets.

Third, when evaluating the performance of IDLAs, AUC, precision, recall, F1-score are often chosen as suitable metrics. In Table 1, two studies use both precision and recall in their experiment (Ounacer et al., 2020; Sanabila and Jatmiko, 2018). Precision and recall can respectively evaluate the predictive ability of classifiers for non-default(majority) samples and default(minority) samples, and F1-score is a comprehensive indicator, which is the weighted average result of precision and recall. But for accuracy, it is not an appropriate metrics in CI problem due to the way it is calculated (Loyola-González et al., 2016). Especially in datasets with high imbalanced ratio (defined as the ratio of the number of samples in the majority class to that in the minority class), high accuracy value cannot reflect models' ability to correctly identify default samples.

From the conclusions of literature research, we find that it is necessary to make a com- prehensive analysis of different types of IDLAs through benchmark experiment in credit scoring. Our work make up for the above deficiencies from several aspects: (1) we make a systematic sorting and comparison of the performance of IDLAs in credit scoring by proposing a new taxonomy for basic IDLAs, choosing several basic approaches and carrying out benchmark experiment under three public datasets around credit scoring; (2) we consider four classical and mainstream GAN models in our benchmark under credit scoring scenario; (3) we choose "balanced accuracy"(introduced in Section 4) that can better reflect the real performance of models in imbalanced credit scoring datasets compared with the naive ac- curacy metrics, at the same time, other metrics(F1-score, precision and recall) with high frequency of usage in CI problem are also considered; (4) In the end, we test the statistical significance of differences among the selected IDLAs for further analysis.

3. Methodology

In this section, we briefly introduce the theory of IDLAs based on the idea of solving a CI problem, we propose a new taxonomy for these approaches (shown in Fig. 1) in which IDLAs are divided into the categories of resampling approaches, synthesizing techniques, embedded framework approaches, and post-hoc adjusting. To distinguish the source of samples added to the rebalanced datasets, we separate "synthesizing techniques" from "resampling approaches". For synthesizing techniques, the instances of the "rebalanced dataset" are generated and are not from the subset of the original, as compared to resampling approaches, in which SMOTEs, GANs, and the adaptive synthetic sampling method (ADASYN) are included.

3.1. Resampling approaches

Resampling can directly change the proportion of the samples in various classes by adding or eliminating original instances from datasets to overcome the CI problem at the data level. Random sampling approaches and advanced sampling approaches are described below.

Random sampling approaches. Random sampling approaches highlight the random selection process for datasets. Random over-sampling (ROS) and random under-sampling (RUS) approaches can change the sample weight by randomly selecting minority class samples or deleting majority class samples, respectively. Owing to the random operation, no complex steps are involved and the approaches are easy to perform. This kind of method reflects the basic idea of handling imbalanced class datasets, but due to a lack of filtering, RUS is apt to cause a loss of key information, and ROS faces the challenge of overfitting.

Advanced sampling approaches. These approaches can be seen as the "updated versions" of simple random sampling approaches. For example, Nearmiss (Mani and Zhang, 2003) proposes adding the idea of nearest neighbors into the procedure of under-sampling to reduce the information loss of RUS; only the majority class samples that meet the standards will be selected. Combining this idea with the idea of "noise cleaning", the Edited Nearest Neighbors (ENN) method (Wilson, 1972), which removes the majority class samples that do not belong to the same class as most of its k-nearest neighbors(k is a hyperparameter) samples; and Tomek-link removal method (Tomek, 1976), which cleans up noise by deleting Tomek-link that two samples of different classes are the nearest neighbors to each other, are two classical "noise cleaning" process, and they are often used as a supplement to other sampling approaches.

3.2. Synthesizing techniques approaches

Synthesizing techniques can create and add new default samples to the original training set to balance the weights of samples in different classes. Compared with resampling approaches, the biggest difference is the mechanism of sample generation, such as adversarial game (GANs), line segment (SMOTEs), and self-adaptation (ADASYN). The latter two are classic and stable approaches with quite a high frequency of use in solving CI problems.

Synthetic minority over-sampling technique. Over-sampling approach can directly increase the weight of the minority class by the process of random selection. However, models can easily face the dilemma of overfitting by adding a large number of duplicate samples. To overcome this, based on the "line segment" rule, SMOTE (Chawla et al., 2002) can create new minority class samples according to the local information of the reference data point. For each minority class sample x_i, a new sample x_g can be obtained as follows:

$$x_g = x_i + r\cdot(x_i - x_{in}) \tag{1}$$

where r is a random number chosen in (0, 1), and x_{in} represents one of the k neighbors of x_i that is in the same category as x_i (k is usually set to

5).

Still, the sample synthesis process of SMOTE is still random, which has the risk of introducing extraneous data points to blur the classification boundary and increasing the difficulty of the classifier learning procedure. To overcome this, various variants of SMOTE have been proposed, like SMOTE-ENN (Batista et al., 2004) and SMOTE-Tomek (Batistaet al., 2003), which combine the idea of cleaning with the original SMOTE process to alleviate the interference of noise samples. Also, Han et al. (2005) considers the influence of the boundary during the SMOTE process. In their method, samples around the borderline are divided into three categories: noise, danger, and safe, and SMOTE is driven to focus on catching the samples around the decision boundary, which are more easily misclassified than those far from the boundary. By choosing and synthesizing samples, Borderline-SMOTE reduces the calculation cost while ensuring the quality of the synthesis samples. As of today, SMOTEs are still the hotspot in machine learning, and they have made great progress in a variety of applications and in continuously promoting the research of imbalanced classification (Fernández et al., 2018).

Generative adversarial network. Compared with SMOTE's interpolation strategy to rebalance datasets, GANs (Goodfellow et al., 2014) view the distribution of data from a global perspective and use the idea of a distribution approximation to overcome the lack of prior probability. GAN methods can synthesize samples via the adversarial training of the network's two basic components: the generator(G) and discriminator (D), which are both multi-layer neural networks with different learning tasks. G learns a distribution and synthesizes samples as close as possible to the actual data; interference network judgment can result. D has to do its best to distinguish the source of samples—accepting real data and rejecting the fake data from G. Connected by game theory, both components gradually reduce their errors and improve their performance. Finally, when reaching "Nash Equilibrium" can make D output a fuzzy judgement result. The optimization function of GAN is defined as follows:

$$\begin{aligned} \underset{G}{min}\underset{D}{max}V(G,D) &= \mathbb{E}_{x\sim p_{data}(x)}\log D(x) + \mathbb{E}_{x\sim p_z(z)}\log(1-D(G(z))) \\ G: minV(G) &= \mathbb{E}_{x\sim p_z(z)}\log(1-D(G(z))) \end{aligned} \tag{2}$$

where $\mathbb{E}$ represents the expectation value, p_{data} and p_z indicate real data distribution and noise data distribution (like Gaussian Distribution) and V is the objective function.

Since the idea of GANs was proposed, researchers have faced some serious problems in the training phase. First, the training process of a GAN lacks stability; if D reaches the optimal state before G, G's gradient required to update its parameter will not be obtained from D (i.e., gradient vanishing). Second, in the original GAN, G tends to fall into the trap of mode collapse (Arjovsky and Bottou, 2017) during the training process, which means that the output of G is uncontrollable and without diversity. Due to the unreasonableness of the Jensen-Shannon (JS) divergence used in GAN, a new kind of structure called a WGAN is proposed, which employs Wasserstein distance instead of JS divergence to measure the difference between the real and fake distributions, changing the optimization objective as follows:

$$W(\mathbb{P}_{real},\mathbb{P}_g) = \underset{\gamma\in\Pi(\mathbb{P}_{real},\mathbb{P}_g)}{inf}\mathbb{E}_{(x,y)\sim\gamma}[\|x-y\|] \tag{3}$$

where $\Pi(\mathbb{P}_{real},\mathbb{P}_g)$ is the joint distribution, and the marginals of it are distribution P_{real} and P_g. This formula is intuitively understood as trying to find the optimal "route planning" among all possible solutions in each of which the "consumption" that moving Preal to Pg should be calculated. Also, there is another "updated version" of WGAN called WGAN-GP (Gulrajani et al., 2017), which uses gradient penalty instead of weight clipping in WGAN to satisfy Lipschitz constraint.

For the sack of control, Mirza (Mirza and Osindero, 2014) add the conditional information(labels) t0 to GAN and form the Conditional GAN(CGAN):

$$\underset{G}{min}\underset{D}{max}V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(\boldsymbol{z}|\boldsymbol{y})))] \tag{4}$$

where information y is associated with inputs of both G and D. After training, G can create samples as required according to the given conditional information. CGAN uses small changes to overcome the randomness of the generated results, and it provides meaningful reference for subsequent GAN frameworks.

GAN has made great progress in Computer Vision, with a wide range of usage scenarios such as image generation, style transformation, text-t o-image and so on. In contrast to an image, in tabular data, rows and columns represent records and features, respectively. Take credit scoring datasets as an example. This kind of data has the following characteristics: a large amount of records, various feature categories, and independent relationships. Some researchers have explored the application of GANs to tabular data: Douzas and Bacao (2018) made progress on imbalanced classification by using CGAN as a data augmenting tool on several datasets with different imbalanced ratios. Combining the advantages of CGAN and WGAN-GP, Zheng et al. (2020) proposed a comprehensive framework named CWGAN-GP; by applying it to several UCI datasets and some real-word records, they proved that this new structure is better than single GANs.

Adaptive synthetic sampling method. The adaptive synthetic sampling method (ADASYN) (He et al., 2008) is another kind of synthesizing method to deal with class imbalance problem. The most obvious characteristic of it is the self-adaption mechanism, in which the number of synthetic samples for minority is analyzed and calculated by the following distribution Γ rather than constructing a 1:1 balanced dataset between classes.

$$\Gamma = \frac{\delta_i/K}{Z} \tag{5}$$

here for minority class sample x_i, δ_i is the number of majority samples of x_i's K nearest neighbors, and Z is a normalization factor to make Γ form a distribution. Then along with $G = \left(m_{majority} - m_{minority}\right) \times \beta(\beta \in [0,1])$, where m the number of samples and β is a parameter to indicate the balance level after synthesizing($\beta = 1$ means a fully balanced set), the synthesizing number g_i for each minority sample is defined as:

$$g_i = \delta_i \times G \tag{6}$$

The adaptive mechanism of ADASYN can be interpreted as that for a random minority sample x_i, distribution Γ gives a diverse weight according to their "data positions", the more majority samples in the K nearest neighbors of x_i, the more minority samples will be generated around it. However, this will cause ADASYN to be insensitive to abnormal minority data points whose neighbors are full of samples with diverse class.

3.3. Embedded framework approaches

Embedded framework approaches are series of integrated solutions which combine specific strategies into the existing solutions to make the hybrid model more robust, to deal with a variety of complex application scenarios. In the following section, two kinds of specific solutions are introduced: the "sampling-embedded" method, in which the sampling or synthesizing approaches mentioned above are embedded into the procedure of ensemble learning, and "cost-embedded" method, which considers class weight or misclassification cost in the ensemble framework or the training process of the original classifiers. More details are given below.

Sampling-embedded. This kind of strategy aims to embed sampling approaches into the framework of ensemble learning. Ensemble learning is a kind of effective trick, in which several classifiers (also called base classifiers) are trained, and their outputs are combined to give a better performance than that of a single classifier. Bagging and boosting are two mainstream approaches of this technique. The main difference between them is the combination strategy of the base classifiers: for bagging, the base classifiers are trained in parallel, and the final results are gotten through a voting mechanism. Bagging can significantly improve the performance of a model and reduce variance. Compared to bagging, in boosting, a weak classifier is gradually transformed to a stronger one with low bias by weight allocating and updating among the samples and classifiers. However, traditional ensemble models lack the ability to tackle the imbalanced class problem and to improve it. Several IDLAs should be used, such as by embedding the sampling approaches mentioned above into the process of the ensemble technique.

The principle of the preprocessing-embedded method is, in the bagging or boosting framework, combining a data preprocessing technique into the construction of subsets in each iteration, to overcome the difficulties of the original ensemble learning approaches in handling the CI problem. For example, by handling random under-sampling to remove majority class samples in each iteration before training, the RUSBoost (Seiffert et al., 2009) algorithm combines the advantages of preprocessing approaches and a boosting strategy, not only increasing the diversity of data between categories, but also improving the prediction ability of the model compared to a single classifier. Similarly, SMOTEBoost is a combination of SMOTE and boosting. Another advanced hybrid solution, the EasyEnsemble (Liu et al., 2008) approach integrates the Under-Bagging (combining under-sampling and bagging) processing approach and an Adaboost-like training approach to construct more base classifiers without increasing the cost of calculation. In this paper, to explore the performance of approaches mentioned above, several typical techniques are chosen in the experiment.

Cost-embedded. Sampling approaches can directly change the weights of different classes by adding or removing instances, a balanced dataset means classifiers can equally treat all samples. Similarly, cost-sensitive learning aims to learn a new classifier by reweighting the original instances without quantitative changes. Its advantage is obvious, it keeps the original distribution of datasets as complete as possible and reduces the cost of calculation. To achieve this, the cost matrix should be well-designed (shown in Table 2), with the restriction of cost value, when an instance is misclassified, the classifier will be punished accordingly. Owing to the goal of optimization, classifiers will readjust the learning strategy and pay attention to the misclassified instances.

In cost matrix, $C(class_A, class_B)$ represents the cost value caused by wrongly classifying an instance of $class_B$ as $class_A$. $C(0,0)$ and $C(1,1)$ are generally set to zero. For some scenarios of imbalanced classification, the C_{FN} should be set larger than the C_{FP}, which means that minority class samples should be treated more carefully, or they will cause great loss, in reality. Take rare disease diagnosis as an example, misclassifying positive samples means losing the chance of treatment and brings a huge cost to both medical institutions and patients. To analyze its performance, several different ratios of C_{FN} and C_{FP}, which means the class weight of minority and majority class are set in Section 4.

3.4. Post-hoc adjusting

The post-hoc adjusting method is more like “wrapper” techniques, which means post modification such as adjusting a classifier’s output by recalculating the probability threshold or modifying the attributes of instances after each iteration. These two solutions correspond to threshold adjusting and MetaCost, the basic ideas of which are described below.

Table 2
Cost matrix.

	Actual negative	Actual positive
Predict negative	$C(0,0)/C_{TN}$*	$C(0,1)/C_{FN}$
Predict positive	$C(1,0)/C_{FP}$	$C(1,1)/C_{TP}$

*TP represents the number of correctly predicted; TN represents the number of correctly predicted; FP represents the number of wrongly predicted as positive; FN represents the number of wrongly predicted as negative.

Threshold adjusting. This kind of method can also be seen as “cost-sensitive classification”. The threshold is adjusted according to the given cost matrix to fit the CI problem. The main idea of this cost-sensitive method is that samples belonging to different categories are treated differently (Ling and Sheng, 2008), and the optimization goal is to minimize the total cost R, defined as follows:

$$R(i|x) = \sum_{j=1}^{n} P(j|x)\cdot C(i,j) \tag{7}$$

here $P(j|x)$ represents the predicting probability. Considering the binary classification issue, if an instance is judged as positive, it should satisfy $R(1|x) \le R(0|x)$, and namely,

$$P(0|x)\cdot C(0,1) + P(1|x)\cdot C(1,1) \le P(0|x)\cdot C(0,0) + P(1|x)\cdot C(0,1) \tag{8}$$

Combining with $P(0|x) = 1 - P(1|x)$, we can get the adjusted threshold:

$$Threshold = P(1|x) = \frac{C(1,0)}{C(1,0)+C(0,1)} = \frac{C_{FP}}{C_{FP}+C_{FN}} \tag{9}$$

The principle can be explained as follows: When facing the CI problem, a classifier may get a high FN (defined in Table 2) under the default threshold setting of 0.5; minority class samples should be treated differently in order to reduce the cost, because in reality, they will cause a huge loss. By setting a different cost value in the cost matrix, classifiers will reconstruct the predicting result, in which FN will get lower, and more minority class samples will be correctly classified. Although this result will be obtained at the expense of increasing FP (defined in Table 2), considering the problems in the real world, it is still significant.

MetaCost. The Metacost algorithm (Domingos, 1999) works like a wrapper method, which is not limited by the choice of classification algorithm and the output of models. It is a flexible framework and has a fast speed of response. As long as the cost matrix and classification algorithm are given, Metacost can quickly implement cost-sensitive classification and help researchers validate their ideas. The basic principle of the Metacost method is to minimize the cost function; the samples of the training set are relabeled according to their cost values, and then classifiers are trained on these “new samples”. In this way, classifiers can overcome the problem of biased output caused by the CI problem, and using this form of training, models can finally achieve a cost-sensitive classification process and reduce the impact of the CI problem on the prediction results.

4. Experimental setup

4.1. Datasets

There are three credit scoring datasets chosen for our experiment, with a high frequency of usage in related works (Rangel-Díaz-de-la Vega et al., 2020; CHEN et al., 2017; Marqués et al., 2013). The Taiwanese credit card dataset (*Taiwan*) and German credit dataset (*Germany*) are from the UCI Machine Learning Repository; the “Give Me Some Credit” (*GMSC*) dataset is from Kaggle. In order to preserve the original structure of these datasets, only some basic preprocesses are done, such as removing missing or duplicated values and normalization (scaled to [–1,1]). One thing we need to explain is that the datasets mentioned above all involve a binary CI problem, and the targets are all mapped to the numeral 1 for positive (minority class) and 0 for negative (majority class). More details are shown in Table 3.

4.2. Parameter setting and experimental environment

According to the proposed taxonomy in Fig. 1, we select 28 approaches for our experiment and benchmark analysis; all the approaches are shown in Table 4. Note that because of the large number of IDLAs

Table 3
Credit scoring datasets for benchmark experiment.

Datasets	Instance	Features	Minority class	IR	Positive Rate	Cross Validation (Fold*Times)	Source
Germany*	1000	24	300	2.33	30 %	5*20	UCI
Taiwan**	29,932	30	6631	3.51	22.15 %	5*20	UCI
GMSC***	120,170	10	8355	13.38	6.95 %	5*20	Kaggle

* https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).
** https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients).
*** https://www.kaggle.com/c/GiveMeSomeCredit.

Table 4
Imbalanced data learning approaches involved in experimental setup.

Category	Sub-category	Imbalanced data learning approaches	Acronym	Models
Resampling approaches	Over-sampling	Random over-sampling	ROS	1
	Under-sampling	Random under-sampling	RUS	1
		Neighbor-based under-sampling	Nearmiss (1/2/3)	3
		Edited nearest neighbors	ENN	1
		Tomek-link removing	Tomek	1
Synthesizing techniques	Synthetic minority over-sampling technique	Standard SMOTE	SMOTE	1
		SMOTE with borderline considering	BS(1/2)	2
		SMOTE with ENN	SE	1
		SMOTE with Tomek-link removing	ST	1
	Generative adversarial network	Standard GAN	GAN	1
		GAN with conditional information	CGAN	1
		GAN with Wasserstein distance (weight clipping)	WGAN	1
		GAN with Wasserstein distance (gradient penalty)	WGAN-GP	1
	Adaptive synthetic sampling	–	ADASYN	1
Embedded framework approaches	Embedding ensemble learning with resampling	RUS + Adaboost	EasyEn	1
		RUS + Boosting	RUSB	1
	Cost-sensitive learning	Cost-sensitive learning	CSL	3
Post-hoc adjusting	Simple adjusting	Cost-sensitive classification	CSC	3
	Metacost technique	Metacost technique	MC	3
				Total:28

with different strategies, it is not possible to describe all approaches in detail, so we select only the typical approaches under each subcategory. For the sake of presentation, we take only binary CI as an example and apply these chosen approaches to training sets in our experiment.

In the following context, name of IDLAs is all represented by the Acronym column of Table 4. In the following, we set parameters for the 28 IDLAs and three classification algorithms: logistic regression (LR), decision tree (DT), and random forest (RF). All sampling approaches and most of the synthesizing approaches except the GANs are from the imblearn (Lemaître et al., 2017) API of Python, used to process the training sets with a default strategy of balance (1:1). To make the results of the experiment repeatable, a fixed random seed value is also set in some typical approaches such as random sampling approaches, SMOTEs, and so on. Three classification algorithms are all from the scikit-learn (Pedregosa et al., 2011) API with default parameters. And the number of base estimators in ensemble-like approaches (EasyEnsemble and RUSBoost) is set as 10.

For the four kinds of GANs chosen in our experiment (GAN, CGAN, WGAN, and WGAN-GP), each network structure is trained 1000 times. In the four GANs, both generators and discriminators are all three-layer, fully connected neural networks without batch normalization layers and dropout layers. The activation functions we used include leaky relu (an updated version of the relu function) and tanh, which are shown below:

$$leaky\_relu(x) = \begin{cases} \alpha \cdot x & ,x < 0, \alpha \in (0,1) \\ x & ,x \geq 0 \end{cases} \tag{10}$$

$$relu(x) = \begin{cases} 0, x < 0 \\ x, x \geq 0 \end{cases} \tag{11}$$

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{12}$$

we use cross entropy as loss function and Adam as optimizing algorithm (setting $\beta_1 = 0.5$, $\beta_2 = 0.9$ in Adam). Random noise z is sampled from Gaussian Distribution with default setting in NumPy package. An additional hyperparameters λ is set as 10 in WGAN-GP structure. the basic layer structure design is shown in Fig. 2.

For the cost-sensitive process, three different ratio values of C_{FP} and C_{FN} are given with 1:2, 1:4 and 1:6, which correspond to threshold values of 0.33, 0.2, and 0.14 according to Section 3.4. On the one hand, the ratio of C_{FP} and C_{FN} is adjusted in cost-sensitive learning by setting the parameter of "class weight" to train the cost-sensitive classifiers; on the other hand, the given threshold values play a role by modifying the outputs of models in the cost-sensitive classification. Approaches related to cost-sensitive learning have the following problems in practical application. First, the cost matrix is domain related; it is usually provided by experts in relevant fields. Typical cost-sensitive approaches assume a constant cost for each type of error, which is called a class-dependent approach. The use of class-dependent approaches is limited in real-world applications owing to the acquisition difficulty of the cost matrix. Second, cost values are also sample related in some scenarios,

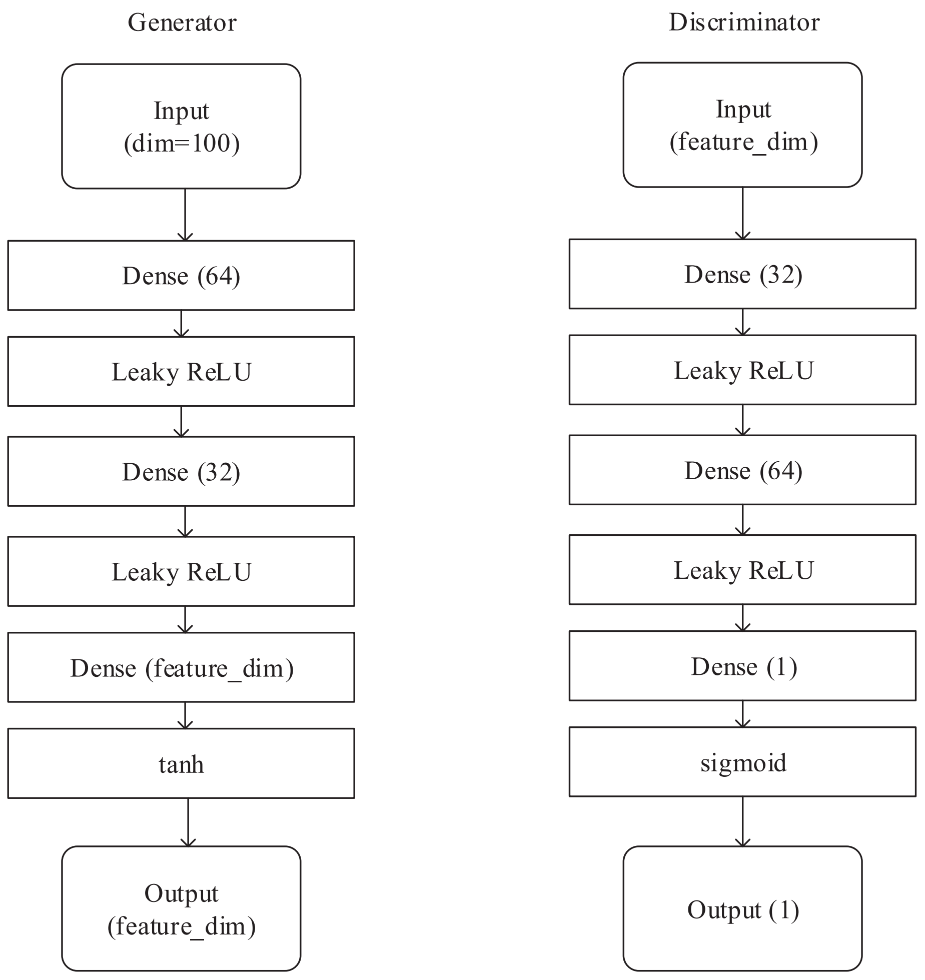

Fig. 2. Network structure of GANs.

because of the varying degrees of consequence when failing to detect the default events.[1] In this paper, the values of the cost matrix are given subjectively because we want only to observe the effect of cost-sensitive approaches, not to obtain a better performance. Similarly, based on the idea of post-hoc adjusting, the MetaCost technique can also produce a cost-sensitive result by reallocating training samples according to the ratio given above.

All experiments are done on the platform Windows-10 Professional Edition (64 bit) with AMD Ryzen 5 3600X 6-Core Processor and 16.0 GB RAM. The four GAN structures (GAN, CGAN, WGAN, and WGAN-GP) are constructed and trained with the deep learning structure of TensorFlow 2.1.0 and GPU of NVIDIA RTX 2070 super (8.0 GB).

4.3. *Training process of GANs*

We elaborate on the selection of the training set in each GAN structure in our experiment, shown in Fig. 3 and Fig. 4, because the details of this part of the experiment have not been mentioned or described comprehensively in previous studies.

The process of GAN training in handling CI problem is divided into two types: with label information and without label information. Training a GAN with label information means that the whole training set is used, and class labels (0 and 1) are added to the input of the generator and discriminator; when finished training, the generator is used to augment the imbalanced data and the extra synthetic minority class samples are generated (input random noise and label of positive class to the generator) and added to the original training set. CGAN is trained using the strategy above (Fig. 3). The rest of the GAN models (GAN, WGAN, and WGAN-GP) are trained using the without-label-information strategy (Fig. 4), in which only the minority class samples are used as the training set, and there is no need to input label information; after training, the generating strategy is similar to that described above.

4.4. *Performance measures*

To evaluate the outputs of the different models and in particular to observe the model's ability to discriminate minority class samples from the other samples in the datasets, several metrics are chosen that are popular in classification tasks (Sokolova and Lapalme, 2009). Table 5 shows the evaluation indicators we selected in the benchmark analysis (Measures related to confusion matrix (TP, TN, FP and FN) are defined in Table 2).

Precision(P) and recall(R) are classic indicators used in imbalanced classification task for the observation of models to accurately identify negative and positive samples respectively. Precision represents the proportion of samples that are correctly predicted among all samples with positive prediction results, and recall indicates the proportion of correctly predicted samples among all real positive samples. Recall receives more attention in many application scenarios, because it can help to reduce loss, such as via credit card fraud detection (Makki et al., 2019), rare disease diagnosis (Haixiang et al., 2017) and so on. The F1 score is a comprehensive indicator that considers both precision and recall. It is also a special care of F_β when $\beta = 1$. F_β is defined as follows:

[1] Costcla: https://albahnsen.github.io/CostSensitiveClassification/.

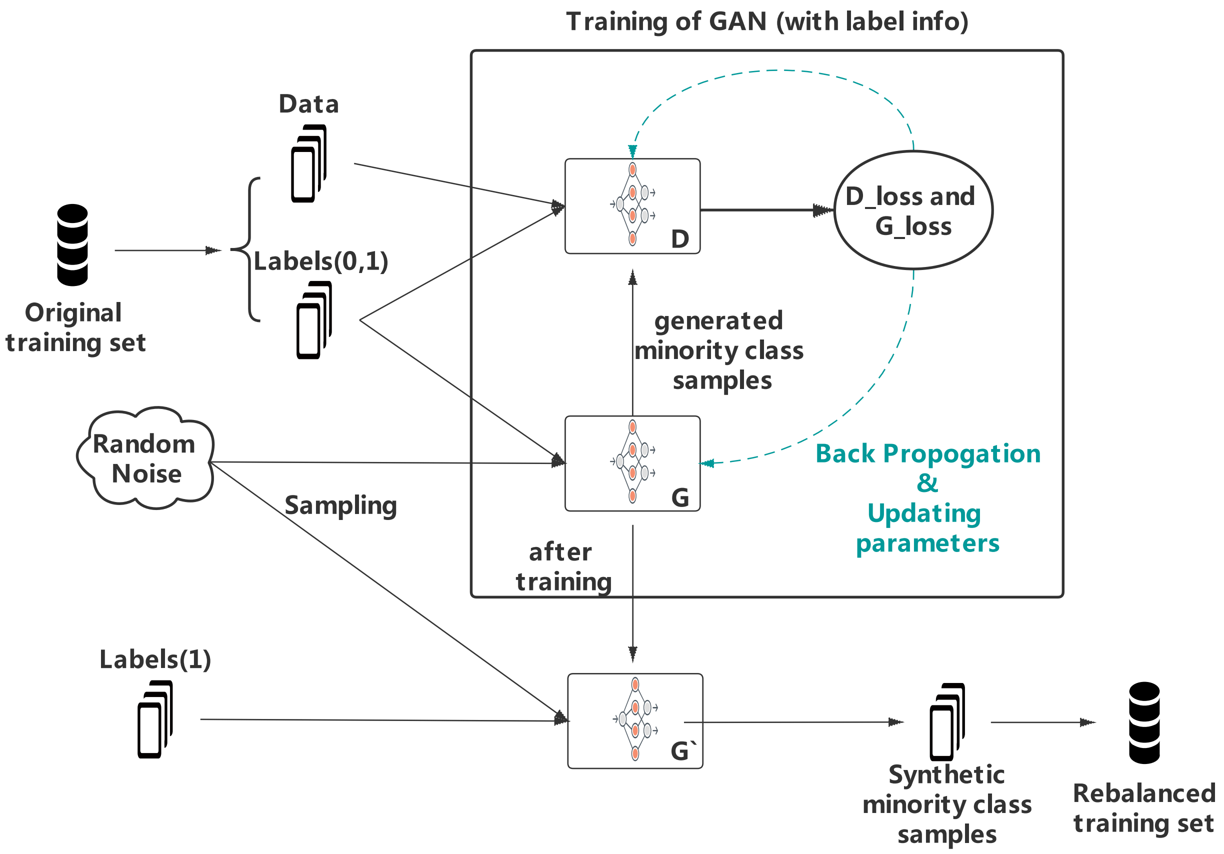
Training of GAN (with label info)
Data
Labels(0,1)
Original training set
D
D_loss and G_loss
generated minority class samples
Random Noise
G
Back Propogation & Updating parameters
Sampling
after training
Labels(1)
G`
Synthetic minority class samples
Rebalanced training set

Fig. 3. Training process of GANs with label information.

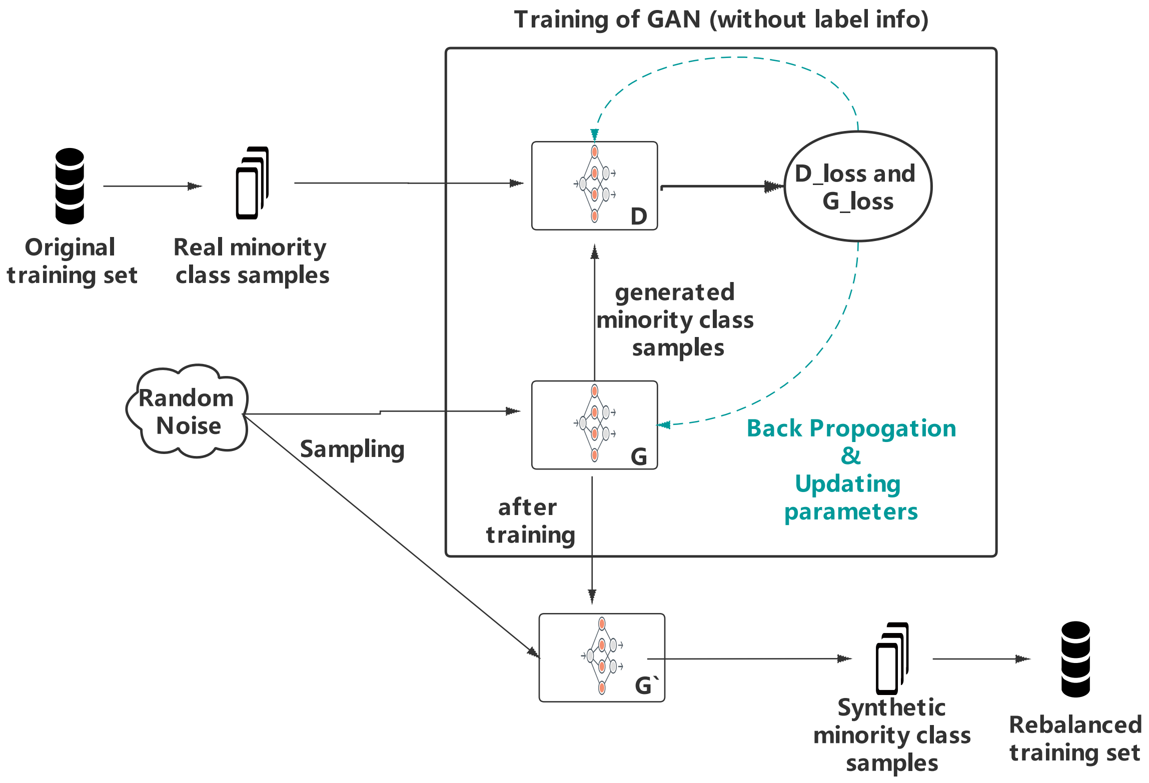
Training of GAN (without label info)
Original training set
Real minority class samples
D
D_loss and G_loss
generated minority class samples
Random Noise
G
Sampling
Back Propogation & Updating parameters
after training
G`
Synthetic minority class samples
Rebalanced training set

Fig. 4. Training process of GANs without label information.

Table 5
Performance metrics of benchmark analysis.

Measure	Equation*
balanced accuracy	$bacc = (TPR + TNR)/2$**
F-measure (F1 score)	$F1 = (2 \cdot Precision \cdot Recall)/(Precision + Recall)$
Precision	$Precision = (TP)/(TP + FP)$
Recall	$Recall = (TP)/(TP + FN)$

*Measures related to confusion (TP, TN, FP and FN) are defined in Table 2.
**True Positive Rate (TPR) = TP/(TP + FN), True Negative Rate (TNR) = TN/(TN + FP).

$$F_\beta = \frac{(1+\beta^2)\cdot P \cdot R}{\beta^2 \cdot P + R} \tag{13}$$

when $\beta = 1$, F_β can be defined as the harmonic mean of precision and recall. F1 score has a widely range of use in statistical analysis and it is a classical metrics in evaluating the effect of binary classification models.

Because of the issue of imbalance, accuracy plays a misleading role in the evaluation process: models with a quite high accuracy score still lack the ability to recognize minority class samples because of the majority class samples' leading role in calculating accuracy. Instead, bacc is defined as the arithmetic mean of sensitivity (TPR) and specificity (TNR). In a classic classification process, it is the same as accuracy; but in a CI problem, it can overcome the defect of accuracy very well, reflecting the classification capabilities of models without providing misleading information.

4.5. *Experimental process*

Fig. 5 illustrates the whole process of the benchmark analysis. Details of the whole experimental process are listed below.

1. **Splitting training and testing sets:** To ensure that training set has a comparable number of default samples, we use repeated fivefold cross validation to split the datasets. To achieve this, we use fivefold cross validation for default and non-default samples and combine the corresponding parts to form the training and testing sets. For each dataset, we repeat the above step 20 times and get 100 different split forms in total.
2. **Handling the CI problem:** The 28 kinds of IDLAs listed in Table 4 are used in the training sets obtained in step 1. With the default balancing strategy (same number of samples between majority and minority class), the rebalanced training sets are obtained at the end of step 2.
3. **Training classifiers:** Three classifiers (LR, DT and RF) are fitted with the rebalanced training sets.
4. **Evaluating and ranking:** Using the performance measures mentioned in Table 5 for evaluating, scores are calculated and ranked in ascending order of the results.

5. Empirical results

The empirical results consist of the performance of the 28 IDLAs across the three credit scoring datasets under the four performance measures. In the subsections below, we first summarize the experimental results and compare them with the relevant results obtained by our predecessors. Then we expand upon the observed results and uncover the causes.

5.1. *Benchmark analysis*

The IDLAs aim to adjust the weight between the minority class and majority class from the perspective of quantity of samples or algorithm, to avoid biased prediction results from classification models. This adjustment is reflected in the actual credit scoring scenario as models' enhanced ability to recognize minority(default) samples. To indicate the results of imbalanced data learning approaches, we choose four performance measures — bacc, F1, recall, and precision. We mainly focus on the first three metrics.

The ranking of the performance of all IDLAs in the three credit scoring datasets are presented in Table 6, Table 7, and Table 8. The smaller the ranking or score of a certain IDLA under a certain metric, the better its effect, which means that the best approach receives a ranking of 1 and the worst approach receives a 28. Bold face in each table indicates the best imbalanced learning method. The penultimate column *AvgR* is the average result of each row of data, and the last column Score is the ranking score of *AvgR*. We test the statistical significance of the differences among the different approaches; the values in brackets represent the p-values corresponding to the pairwise comparison between an imbalanced learning method and the best imbalanced learning approach in the row beneath, for each evaluating metrics. We use an underscore to show that the relevant p- value is significant at a 95 percent confidence level. And the last row is the $\chi 2$ value and its p-value per column.

In order to highlight the performance of the different IDLAs in the dimension of performance measure, we fuse the results of the three datasets. Table 9 shows the ranking of the average of the scores in Table 6, Table 7, and Table 8 per performance metric, and the meaning

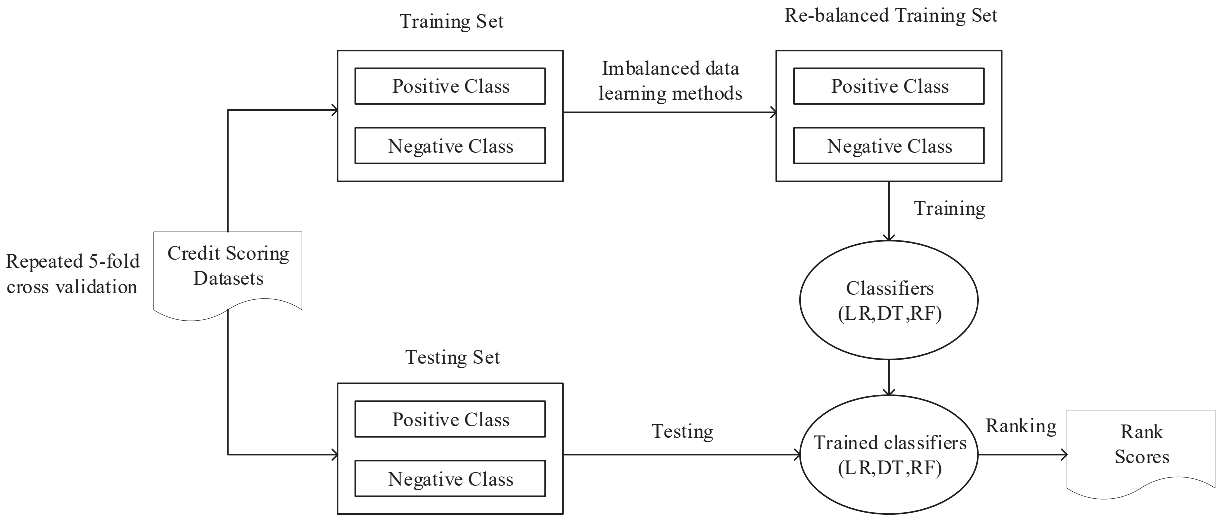

Fig. 5. Benchmark process.

Table 6
Average ranks of *Germany* for different performance measures.

		LR_BACC	LR_F1	LR_P	LR_R	DT_BACC	DT_F1	DT_P	DT_R	RF_BACC	RF_F1	RF_P	RF_R	AvgR	Score
Traditional classification	Original	20.48 (0.000)	22.37 (0.000)	**1.49** (/)	27.72 (0.000)	14.17 (0.000)	16.33 (0.000)	**1.40** (/)	24.81 (0.000)	12.97 (0.000)	14.73 (0.000)	**4.29** (/)	20.97 (0.000)	15.14	13
Resampling approaches	ROS	8.66 (0.219)	8.96 (0.305)	12.35 (0.000)	15.03 (0.000)	5.96 (0.868)	6.03 (0.790)	9.16 (0.000)	8.03 (0.000)	8.60 (0.062)	8.91 (0.004)	14.45 (0.000)	14.04 (0.000)	10.01	4
	RUS	8.87 (0.162)	9.16 (0.233)	14.13 (0.000)	13.46 (0.000)	6.07 (0.800)	6.10 (0.746)	10.95 (0.000)	6.81 (0.000)	6.95 (0.621)	7.00 (0.197)	17.67 (0.000)	8.64 (0.000)	**9.65**	1
	ENN	8.67 (0.217)	9.02 (0.284)	17.87 (0.000)	8.94 (0.000)	6.00 (0.845)	6.10 (0.746)	15.71 (0.000)	2.96 (0.152)	8.37 (0.093)	8.19 (0.023)	21.60 (0.000)	5.28 (0.000)	9.89	2
	Tomek	14.92 (0.000)	15.89 (0.000)	2.77 (0.288)	25.10 (0.000)	9.81 (0.000)	10.52 (0.000)	2.17 (0.523)	19.35 (0.000)	8.86 (0.037)	9.56 (0.001)	6.80 (0.037)	19.41 (0.000)	12.09	11
	Nearmiss1	23.76 (0.000)	23.02 (0.000)	24.49 (0.000)	14.72 (0.000)	17.79 (0.000)	16.15 (0.000)	25.38 (0.000)	7.72 (0.000)	19.98 (0.000)	17.66 (0.000)	26.83 (0.000)	8.02 (0.000)	18.79	22
	Nearmiss2	22.36 (0.000)	21.80 (0.000)	23.51 (0.000)	14.53 (0.000)	15.96 (0.000)	14.45 (0.000)	23.80 (0.000)	7.51 (0.000)	20.24 (0.000)	17.97 (0.000)	26.64 (0.000)	8.09 (0.000)	18.07	21
	Nearmiss3	12.66 (0.000)	13.18 (0.000)	14.00 (0.000)	17.84 (0.000)	8.75 (0.013)	8.58 (0.017)	11.07 (0.000)	10.06 (0.000)	9.64 (0.006)	9.53 (0.001)	17.50 (0.000)	11.39 (0.000)	12.01	10
Synthesizing techniques	SMOTE	8.52 (0.268)	8.68 (0.428)	11.48 (0.000)	15.79 (0.000)	5.84 (0.947)	**5.71** (/)	8.54 (0.000)	8.62 (0.000)	8.94 (0.031)	9.29 (0.001)	14.15 (0.000)	14.67 (0.000)	10.02	5
	BS1	8.73 (0.199)	9.10 (0.255)	12.72 (0.000)	14.64 (0.000)	6.07 (0.800)	6.05 (0.774)	9.56 (0.000)	7.69 (0.000)	8.15 (0.135)	8.65 (0.008)	13.84 (0.000)	13.74 (0.000)	9.91	3
	BS2	8.28 (0.361)	8.67 (0.435)	13.90 (0.000)	12.67 (0.000)	**5.76** (/)	**5.71** (/)	10.73 (0.000)	6.10 (0.000)	9.44 (0.010)	9.75 (0.000)	15.53 (0.000)	13.82 (0.000)	10.03	6
	SE	14.18 (0.000)	13.79 (0.000)	21.96 (0.000)	6.88 (0.000)	9.80 (0.001)	9.34 (0.003)	22.24 (0.000)	**1.30** (/)	10.51 (0.001)	10.19 (0.000)	23.27 (0.000)	4.65 (0.004)	12.34	12
	ST	8.72 (0.202)	8.90 (0.329)	11.91 (0.000)	15.80 (0.000)	6.05 (0.813)	5.91 (0.865)	8.87 (0.000)	8.59 (0.000)	9.69 (0.006)	10.04 (0.000)	14.13 (0.000)	15.07 (0.000)	10.30	8
	ADASYN	9.95 (0.022)	9.93 (0.068)	12.77 (0.000)	15.88 (0.000)	6.89 (0.348)	6.62 (0.447)	9.57 (0.000)	8.72 (0.000)	7.59 (0.305)	7.98 (0.035)	13.90 (0.000)	13.22 (0.000)	10.25	7
	GAN	**7.18** (/)	**7.73** (/)	13.57 (0.000)	12.47 (0.000)	12.44 (0.000)	11.07 (0.000)	15.11 (0.000)	12.48 (0.000)	**6.35** (/)	**5.45** (/)	17.73 (0.000)	7.29 (0.000)	10.74	9
	CGAN	15.06 (0.000)	15.21 (0.000)	17.35 (0.001)	17.49 (0.000)	24.70 (0.000)	26.35 (0.000)	11.59 (0.000)	27.41 (0.000)	10.48 (0.001)	10.20 (0.000)	13.05 (0.000)	14.69 (0.000)	16.96	19
	WGAN	22.70 (0.000)	24.28 (0.000)	4.84 (0.005)	27.16 (0.000)	20.08 (0.000)	20.11 (0.000)	16.92 (0.000)	20.31 (0.000)	22.25 (0.000)	24.14 (0.000)	6.95 (0.027)	24.73 (0.000)	19.54	25
	WGAN-GP	18.65 (0.000)	19.28 (0.000)	13.59 (0.000)	23.40 (0.000)	23.21 (0.000)	22.94 (0.000)	20.43 (0.000)	22.23 (0.000)	23.89 (0.000)	24.73 (0.000)	7.43 (0.009)	25.39 (0.000)	20.43	27
Embedded framework approaches	EasyEn	22.83 (0.000)	24.39 (0.000)	5.09 (0.003)	27.16 (0.000)	20.22 (0.000)	19.98 (0.000)	17.35 (0.000)	20.38 (0.000)	23.60 (0.000)	24.29 (0.000)	7.33 (0.012)	25.25 (0.000)	19.82	26
	RUSB	24.28 (0.000)	25.70 (0.000)	5.27 (0.002)	27.63 (0.000)	21.44 (0.000)	21.21 (0.000)	17.57 (0.000)	21.40 (0.000)	24.82 (0.000)	24.88 (0.000)	7.73 (0.004)	25.04 (0.000)	20.58	28
	CSL12	10.12 (0.015)	10.25 (0.036)	9.60 (0.000)	19.73 (0.000)	21.64 (0.000)	21.56 (0.000)	17.52 (0.000)	21.46 (0.000)	14.35 (0.000)	14.11 (0.000)	19.74 (0.000)	15.22 (0.000)	16.27	16
	CSL14	14.90 (0.000)	14.27 (0.000)	23.44 (0.000)	5.34 (0.002)	21.64 (0.000)	21.56 (0.000)	17.52 (0.000)	21.46 (0.000)	15.42 (0.000)	13.90 (0.000)	25.54 (0.000)	4.75 (0.003)	16.64	18
	CSL16	23.96 (0.000)	21.35 (0.000)	27.51 (0.000)	2.24 (0.642)	21.64 (0.000)	21.56 (0.000)	17.52 (0.000)	21.46 (0.000)	22.64 (0.000)	18.19 (0.000)	28.70 (0.000)	**1.21** (/)	19.00	23
Post-hoc adjusting	CSC12	8.14 (0.425)	8.36 (0.601)	8.13 (0.000)	19.43 (0.000)	22.63 (0.000)	22.90 (0.000)	18.31 (0.000)	23.34 (0.000)	24.57 (0.000)	25.70 (0.000)	6.04 (0.147)	26.12 (0.000)	17.80	20

(*continued on next page*)

Table 6 (*continued*)

	LR_BACC	LR_F1	LR_P	LR_R	DT_BACC	DT_F1	DT_P	DT_R	RF_BACC	RF_F1	RF_P	RF_R	AvgR	Score
CSC14	14.61	14.19	23.28	4.96	23.46	23.92	18.51	24.15	25.26	26.44	6.69	26.72	19.35	24
	(0.000)	(0.000)	(0.000)	(0.006)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.046)	(0.000)		
CSC16	23.91	21.24	27.88	**1.68**	23.02	23.57	17.77	24.19	26.04	27.04	7.23	27.14	20.89	29
	(0.000)	(0.000)	(0.000)	(/)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.015)	(0.000)		
MC12	9.46	9.87	8.47	20.09	18.95	20.71	16.79	20.22	21.47	22.80	7.04	23.32	16.60	17
	(0.059)	(0.076)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.022)	(0.000)		
MC14	15.66	14.70	23.70	5.16	18.51	18.18	19.50	16.07	11.07	11.06	17.87	14.23	15.47	14
	(0.000)	(0.000)	(0.000)	(0.004)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)		
MC16	24.85	21.77	28.02	2.12	16.58	15.89	23.51	10.26	12.94	12.68	25.41	2.96	15.65	15
	(0.000)	(0.000)	(0.000)	(0.715)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.147)		
Friedman 0.05	1541.481	1427.82	2381.93	2489.28	1877.833	1969.87	1399.47	2281.18	1830.358	1905.62	2129.767	2943.404		
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)		

of the last two columns remains unchanged.

First, resampling approaches are effective, although they have some potential disadvantages such as easily causing overfitting (ROS) or losing information (RUS). According to Table 9, ROS has the best performance with the score of 1.5, and the performance of ROS is better than that of RUS (3.0), which confirms previous findings (Japkowicz and Stephen, 2002; Marqués et al., 2013; García et al., 2012). However, the performance of the clustering- based resampling approaches is not ideal. For example, in Table 9, Nearmiss2 shows the worst performance with the score of 29, and both Nearmiss1 and Nearmiss2 perform worse than Nearmiss3.

Second, the sophisticated imbalanced data learning approaches do not necessarily im- prove the predicting accuracy for positive samples. On the one hand, for state-of-the-art GANs, when using datasets with similar structure (tabular datasets) as done in past research (Douzas and Bacao, 2018; Zheng et al., 2020), results show that a simple GAN performs better than the sophisticated versions of CGAN, WGAN and WGAN-GP. In Table 9, GAN, CGAN, WGAN, and WGAN-GP show scores of 1.5, 22, 24, and 15, respectively. GAN, CGAN, and WGAN-GP alleviate the CI problem, but WGAN does not play a simi- lar role. From the perspective of network principles, compared with GAN, CGAN has made the following improvements: using the entire training set for training to ensure sufficient data sources, and adding label information to make the generated results of the network more directional. At the same time, WGAN and WGAN-GP use training strategies of weight clipping and gradient penalty to overcome the shortcomings of mode collapse and the unstable training process of GAN, but among the four structures, the simplest GAN structure has gained the greatest performance improvement. Therefore, when dealing with tabular datasets, the advanced level of GANs is not a determinant factor to the performance results.

On the other hand, approaches with an embedded framework, which mainly combine an imbalanced classification process with an ensemble learning framework, do not play a better role compared with concise approaches. From Table 9, we find that RUSB, a boosting method integrated with an under-sampling process, gains a score of 25.5; this performance is even worse than that of a single under-sampling. Compared with the results without imbalanced data learning process, RUSB does not improve the prediction bias caused by the CI problem. In principle, the ensemble learning approach with imbalanced processing can improve the predictive ability of the model under the premise of avoiding bias and the occurrence of overfitting; but from the results point of view, this effect has not been achieved. Similarly, the comprehensive approach EasyEn is also not as effective as the single ones.

Third, the performance of the SMOTE family is stable, and the effect of SMOTE variants are not necessarily better than that of traditional SMOTE. Among all the approaches with a score of less than 10 (ROS (1.5), RUS (3.0), ENN (5.5), SMOTE (5.5), BS1 (8.5), BS2 (5.5), SE (10.0), ST (5.5), ADASYN (8.5), and GAN (1.5)), SMOTE and its variants occupy-six of them. Through in-depth analysis, the performance of techniques in the SMOTE family is related to the sample distribution of specific datasets. For example, in the results of the GMSC dataset, the hybrid approaches SE and ST perform better than the original SMOTE approach in most cases. When using a logistic regression classifier, SENN even achieved the best results under the BACC and F1 indicators, but with the Taiwan dataset, this kind of negative impact is insignificant, and a good effect can be achieved by using ordinary SMOTE. That means that in the GMSC dataset, class-overlap caused by noise samples is the main issue affecting the classification results. More analysis of the internal effect will be presented below.

Fourth, compared with the original approach, which does not use an imbalanced data learning process, cost-sensitive approaches (cost-sensitive classification, cost-sensitive learning and MetaCost) have played a positive role in promoting the classification effect to varying degrees, and with the given cost value, results show that MetaCost performs best in related approaches. The results related to cost-sensitive theory are based on the cost matrix, the value of which is intuitively given and

Table 7
Average ranks of *Taiwan* for different performance measures.

		LR_BACC	LR_F1	LR_P	LR_R	DT_BACC	DT_F1	DT_P	DT_R	RF_BACC	RF_F1	RF_P	RF_R	AvgR	Score
Traditional classification	Original	22.46 (0.000)	27.48 (0.000)	**1.40** (/)	28.37 (0.000)	24.55 (0.000)	27.70 (0.000)	**1.00** (/)	28.70 (0.000)	16.66 (0.000)	15.94 (0.000)	**1.28** (/)	20.87 (0.000)	18.03	21
Resampling approaches	ROS	6.26 (0.07)	8.30 (0.000)	11.93 (0.000)	14.09 (0.000)	3.49 (0.325)	3.81 (0.021)	15.70 (0.000)	7.73 (0.000)	4.97 (0.024)	5.94 (0.001)	18.23 (0.000)	10.48 (0.000)	9.24	3
	RUS	7.17 (0.01)	8.76 (0.000)	11.73 (0.000)	15.44 (0.000)	3.96 (0.171)	4.21 (0.008)	15.46 (0.000)	8.67 (0.000)	4.09 (0.185)	5.55 (0.004)	18.15 (0.000)	10.13 (0.000)	9.44	4
	ENN	3.38 (0.746)	**1.88** (/)	7.09 (0.000)	22.24 (0.000)	2.33 (0.983)	**1.03** (/)	3.00 (0.097)	15.50 (0.000)	3.86 (0.129)	**2.09** **(/)**	13.06 (0.000)	15.59 (0.000)	**7.59**	1
	Tomek	18.98 (0.000)	18.68 (0.000)	3.07 (0.164)	26.92 (0.000)	13.31 (0.000)	13.21 (0.000)	2.00 (0.406)	27.56 (0.000)	14.62 (0.000)	12.07 (0.000)	5.43 (0.001)	19.33 (0.000)	14.60	16
	Nearmiss1	26.62 (0.000)	25.30 (0.000)	25.06 (0.000)	12.05 (0.000)	27.65 (0.000)	24.45 (0.000)	27.98 (0.000)	5.92 (0.000)	27.83 (0.000)	26.15 (0.000)	28.00 (0.000)	**1.32** (/)	21.53	27
	Nearmiss2	29.00 (0.000)	28.54 (0.000)	29.00 (0.000)	18.23 (0.000)	28.84 (0.000)	28.38 (0.000)	29.00 (0.000)	11.34 (0.000)	29.00 (0.000)	28.49 (0.000)	29.00 (0.000)	1.84 (0.666)	24.22	29
	Nearmiss3	11.27 (0.000)	11.26 (0.000)	12.35 (0.000)	18.14 (0.000)	6.95 (0.000)	6.84 (0.000)	16.15 (0.000)	10.82 (0.000)	19.37 (0.000)	19.20 (0.000)	25.92 (0.000)	10.43 (0.000)	14.06	13
Synthesizing techniques	SMOTE	9.46 (0.000)	10.78 (0.000)	15.00 (0.000)	11.89 (0.000)	5.66 (0.005)	6.11 (0.000)	20.61 (0.000)	5.85 (0.000)	7.62 (0.000)	7.87 (0.000)	19.10 (0.000)	11.38 (0.000)	10.94	6
	BS1	14.99 (0.000)	15.04 (0.000)	17.83 (0.000)	8.78 (0.000)	10.68 (0.000)	10.93 (0.000)	24.97 (0.000)	2.96 (0.104)	10.99 (0.000)	12.30 (0.000)	23.45 (0.000)	5.58 (0.000)	13.21	11
	BS2	15.52 (0.000)	15.61 (0.000)	18.79 (0.000)	7.59 (0.000)	11.01 (0.000)	11.32 (0.000)	25.69 (0.000)	2.11 (0.357)	7.02 (0.000)	5.99 (0.001)	15.89 (0.001)	14.08 (0.000)	12.55	10
	SE	19.41 (0.000)	19.25 (0.000)	22.98 (0.000)	4.00 (0.013)	13.27 (0.000)	13.23 (0.000)	26.95 (0.000)	**1.00** (/)	10.56 (0.000)	12.35 (0.000)	23.98 (0.000)	4.54 (0.007)	14.29	15
	ST	8.30 (0.000)	9.80 (0.000)	13.74 (0.000)	13.08 (0.000)	4.71 (0.046)	5.22 (0.001)	18.87 (0.000)	6.99 (0.000)	7.83 (0.000)	8.11 (0.000)	19.02 (0.000)	11.01 (0.000)	10.55	5
	ADASYN	12.95 (0.000)	13.04 (0.000)	16.37 (0.000)	9.93 (0.000)	8.94 (0.000)	9.16 (0.000)	23.77 (0.000)	3.95 (0.014)	10.44 (0.000)	11.69 8.11 (0.000)	22.80 (0.000)	6.77 (0.000)	12.48	8.5
	GAN	6.80 (0.002)	8.25 (0.000)	11.53 (0.000)	14.88 (0.000)	**2.31** (/)	2.43 (0.247)	5.10 (0.001)	10.94 (0.000)	2.82 (0.000)	5.12 (0.012)	19.06 (0.000)	8.07 (0.000)	8.11	2
	CGAN	21.59 (0.000)	21.23 (0.000)	23.90 (0.000)	17.22 (0.000)	24.34 (0.000)	23.64 (0.000)	14.80 (0.000)	22.98 (0.000)	25.70 (0.000)	26.75 (0.000)	9.36 (0.000)	27.13 (0.000)	21.55	28
	WGAN	**2.99** (/)	4.07 (0.068)	8.29 (0.000)	20.82 (0.000)	20.32 (0.000)	20.27 (0.000)	14.63 (0.000)	20.74 (0.000)	21.36 (0.000)	21.32 (0.000)	6.90 (0.000)	23.62 (0.000)	15.43	17
	WGAN-GP	23.19 (0.000)	25.03 (0.000)	16.53 (0.000)	24.09 (0.000)	20.61 (0.000)	20.52 (0.000)	14.65 (0.000)	20.68 (0.000)	21.51 (0.000)	21.84 (0.000)	6.86 (0.000)	23.96 (0.000)	19.95	25
Embedded framework approaches	EasyEn	3.58 (0.621)	4.57 (0.000)	8.99 (0.000)	20.19 (0.000)	20.17 (0.000)	20.04 (0.000)	14.81 (0.000)	20.49 (0.000)	21.52 (0.000)	21.27 (0.000)	6.03 (0.000)	23.52 (0.000)	15.44	18
	RUSB	22.75 (0.000)	27.80 (0.000)	1.68 (0.816)	28.60 (0.000)	20.66 (0.000)	20.47 (0.000)	15.21 (0.000)	20.81 (0.000)	21.91 (0.000)	21.77 (0.000)	7.26 (0.000)	23.77 (0.000)	19.39	24
	CSL12	4.79 (0.135)	2.48 (0.618)	5.92 (0.000)	23.55 (0.000)	20.31 (0.000)	20.22 (0.000)	14.99 (0.000)	20.36 (0.000)	13.42 (0.000)	12.26 (0.000)	15.02 (0.000)	17.87 (0.000)	14.26	14
	CSL14	16.78 (0.000)	16.87 (0.000)	20.37 (0.000)	6.79 (0.000)	20.31 (0.000)	20.22 (0.000)	14.99 (0.000)	20.36 (0.000)	13.67 (0.000)	14.89 (0.000)	23.37 (0.000)	9.83 (0.000)	16.54	19
	CSL16	25.46 (0.000)	22.40 (0.000)	26.08 (0.000)	2.21 (0.315)	20.31 (0.000)	20.22 (0.000)	14.99 (0.000)	20.36 (0.000)	18.31 (0.000)	18.57 (0.000)	26.98 (0.000)	2.88 (0.194)	18.23	22
Post-hoc adjusting	CSC12	7.72 (0.000)	4.50 (0.029)	4.56 (0.009)	25.47 (0.000)	21.66 (0.000)	22.08 (0.000)	12.66 (0.000)	24.00 (0.000)	24.34 (0.000)	24.55 (0.000)	5.44 (0.001)	26.54 (0.000)	16.69	20

(*continued on next page*)

Table 7 (*continued*)

	LR_BACC	LR_F1	LR_P	LR_R	DT_BACC	DT_F1	DT_P	DT_R	RF_BACC	RF_F1	RF_P	RF_R	AvgR	Score
CSC14	15.61	15.70	19.65	5.55	21.67	22.24	11.13	24.58	25.14	25.57	5.05	27.36	18.27	23
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.002)	(0.000)		
CSC16	27.14	23.20	27.97	**1.00**	21.58	22.35	9.08	25.44	25.63	26.28	4.27	27.97	20.16	26
	(0.000)	(0.000)	(0.000)	(/)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.013)	(0.000)		
MC12	6.48	3.55	4.75	24.77	17.03	17.10	11.09	17.14	16.45	15.75	8.35	19.96	13.53	12
	(0.004)	(0.164)	(0.005)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)		
MC14	17.89	18.04	21.59	6.83	11.29	11.00	6.13	14.23	6.17	2.14	11.95	17.09	11.99	7
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.001)	(0.964)	(0.000)	(0.000)		
MC16	26.55	23.64	26.90	2.79	7.13	6.67	9.65	12.88	**2.26**	3.15	15.96	12.18	12.48	8.5
	(0.000)	(0.000)	(0.000)	(0.137)	(0.000)	(0.000)	(0.000)	(0.000)	(/)	(0.379)	(0.000)	(0.000)		
Friedman 0.05	2666.865	2732.371	2729.896	2770.470	2563.691	2558.438	2253.760	2646.156	2688.993	2678.885	2662.858	2686.254		
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)		

without careful design. Since the cost value cannot be determined by training an iterative optimization, we believe that comprehensive business research helps to design a more reasonable cost matrix, which can promote cost-sensitive methods to exert better performance.

5.2. *Extended analysis of the CI problem*

Based on the analysis of the experimental results, in the following section, we further explore the internal causes of CI problems. Intuitively, the number gap of instances between the majority and minority classes is the most prominent feature of the CI problem, but it is not the determining factor of the imbalanced classification issue (Sun et al., 2009; Japkowicz and Stephen, 2002). Instead, other reasons cause the imbalanced classification challenge, such as class-overlapping, the disturbance of noise, and the existence of subcategories (Ali et al., 2013; Thabtah et al., 2020; Haixiang et al., 2017).

First, class-overlapping, in which majority and minority samples have fuzzy boundaries in spatial distribution and are chaotic, which makes identification of the classifier difficult, is a problem that cannot be ignored when dealing with the CI problem. Class-overlapping is related only to the structure and sample distribution of the dataset itself and has nothing to do with the gap in the number of samples between classes. When the decision boundary is clear enough, the classifier can also accurately recognize samples from a different class with the imbalanced class problem (Liu et al., 2019). In this case, the imbalanced ratio (IR) becomes meaningless. From the experimental results of this article, we have confirmed the statement above that class-overlapping complicates the imbalanced classification task. By analyzing the results of the Germany (IR = 2.33) and Taiwan (IR = 3.51) datasets, we see that the results in Table 6 shows that the Borderline-SMOTE method performs better than SMOTE based on score (SMOTE = 5, BS1 = 3, BS2 = 6), but in Table 7, SMOTE's performance is better than that of Borderline-SMOTE (SMOTE = 6, BS1 = 11, BS2 = 10). This means that although the gap of sample size between classes in the Germany dataset is smaller than that of the Taiwan ($IR_{Germany} \leq IR_{Taiwan}$), for the former, class-overlapping is indeed the main cause of imbalance between classes. The purpose of Borderline-SMOTE is more clear than ordinary SMOTE; it can summarize positive sample points close to the border and selectively synthesize minority samples. When class-overlapping exists, if SMOTE is directly used to increase positive samples, irrelevant sample points may be introduced, and the final effect will be limited; at this time, Borderline-SMOTE will exert a greater positive effect. The above results not only confirm that class-overlapping will affect the effect of an imbalanced data learning process and has nothing to do with the imbalance ratio between classes, but also show that there is no absolute difference in performance between SMOTE and its "updated" versions.

Second, noise points can also play a certain disruptive role in an imbalanced data learning process. Noise points usually appear away from the sample cluster area of existing clusters of heterogeneous samples. In Table 8, SE performs better than SMOTE in terms of the F1 score; this shows that noise cleaning indeed plays a positive role for the *GMSC* dataset. If SMOTE is directly used to increase positive samples, more irrelevant sample points will be introduced, and the discrimination ability of the classifier will be limited. However, with a noise cleaning strategy, SE will identify the positive samples near the boundary and remove them. In addition to using IDLAs with noise cleaning, one can also use data preprocessing, to reduce the interference of noise points, to a certain extent.

Third, since the occurrence of default events may be caused by multiple behavior pat- terns, samples representing different behavior patterns exhibit the characteristics of different subcategories, in terms of the samples' distribution. Under the CI problem, the number of default events caused by different behavior patterns is scarce, and the model is not conducive to capturing different patterns (sample clusters). This is especially prominent when using GANs as data synthesizing tools. During the training step of a GAN, the issue of mode collapse may easily

Table 8
Average ranks of *GMSC* for different performance measures.

		LR_BACC	LR_F1	LR_P	LR_R	DT_BACC	DT_F1	DT_P	DT_R	RF_BACC	RF_F1	RF_P	RF_R	AvgR	Score
Traditional classification	Original	23.25 (0.000)	25.22 (0.000)	5.84 (0.049)	25.36 (0.000)	26.75 (0.000)	28.49 (0.000)	2.24 (0.558)	28.49 (0.000)	21.03 (0.000)	19.66 (0.000)	**1.06** (/)	23.56 (0.000)	19.24	25
Resampling approaches	ROS	3.55 (0.038)	4.61 (0.030)	18.63 (0.000)	8.25 (0.000)	3.87 (0.017)	7.31 (0.000)	21.68 (0.000)	8.10 (0.000)	2.27 (0.292)	11.16 (0.000)	22.16 (0.000)	5.56 (0.000)	9.76	3
	RUS	10.08 (0.000)	10.71 (0.000)	24.19 (0.000)	4.04 (0.036)	10.63 (0.000)	22.96 (0.000)	26.00 (0.000)	4.33 (0.008)	3.77 (0.022)	9.36 (0.000)	20.54 (0.000)	7.71 (0.000)	12.86	10
	ENN	17.03 (0.000)	18.96 (0.000)	**3.47** (/)	18.97 (0.000)	25.05 (0.000)	26.99 (0.000)	**1.53** (/)	26.97 (0.000)	15.86 (0.000)	3.18 (0.070)	8.43 (0.000)	18.56 (0.000)	15.42	15
	Tomek	23.55 (0.000)	25.22 (0.000)	5.84 (0.049)	25.36 (0.000)	26.75 (0.000)	28.49 (0.000)	2.24 (0.558)	28.49 (0.000)	18.35 (0.000)	17.12 (0.000)	1.94 (0.465)	20.47 (0.000)	18.62	23
	Nearmiss1	28.77 (0.000)	16.39 (0.000)	28.76 (0.000)	1.78 (0.829)	28.78 (0.000)	25.72 (0.000)	28.78 (0.000)	2.50 (0.266)	15.45 (0.000)	26.99 (0.000)	27.00 (0.000)	**1.04** (/)	19.33	26
	Nearmiss2	25.40 (0.000)	15.91 (0.000)	28.20 (0.000)	12.23 (0.000)	27.55 (0.000)	25.26 (0.000)	28.22 (0.000)	11.42 (0.000)	29.00 (0.000)	29.00 (0.000)	29.00 (0.000)	2.17 (0.350)	21.95	29
	Nearmiss3	14.42 (0.000)	14.57 (0.000)	26.98 (0.000)	10.60 (0.000)	23.94 (0.000)	23.99 (0.000)	27.00 (0.000)	10.10 (0.000)	26.55 (0.000)	28.00 (0.000)	28.00 (0.000)	7.82 (0.000)	20.16	28
Synthesizing techniques	SMOTE	5.29 (0.000)	7.39 (0.000)	21.09 (0.000)	5.78 (0.000)	5.45 (0.000)	11.69 (0.000)	23.98 (0.000)	5.74 (0.000)	7.29 (0.000)	12.06 (0.000)	21.94 (0.000)	8.92 (0.000)	11.38	7
	BS1	6.90 (0.000)	4.96 (0.014)	17.99 (0.000)	11.21 (0.000)	7.02 (0.000)	7.54 (0.000)	21.04 (0.000)	10.80 (0.000)	5.13 (0.001)	9.10 (0.000)	19.98 (0.000)	8.95 (0.000)	10.88	6
	BS2	5.25 (0.000)	4.07 (0.086)	17.61 (0.000)	9.84 (0.000)	5.54 (0.000)	6.72 (0.000)	20.64 (0.000)	9.53 (0.000)	6.53 (0.000)	7.31 (0.000)	18.72 (0.000)	10.55 (0.000)	10.19	4
	SE	**1.05** (/)	**2.00** (/)	15.83 (0.000)	4.90 (0.005)	2.00 (0.046)	4.14 (0.010)	18.83 (0.000)	5.05 (0.001)	3.79 (0.021)	13.34 (0.000)	23.65 (0.000)	5.24 (0.000)	**8.32**	1
	ST	2.93 (0.118)	5.30 (0.006)	19.50 (0.000)	7.01 (0.000)	3.28 (0.059)	8.02 (0.000)	22.49 (0.000)	6.86 (0.000)	7.11 (0.000)	12.15 (0.000)	22.20 (0.000)	8.60 (0.000)	10.45	5
	ADASYN	7.70 (0.000)	8.60 (0.000)	22.28 (0.000)	**1.52** (/)	7.86 (0.000)	20.54 (0.000)	24.98 (0.000)	2.43 (0.292)	8.14 (0.000)	13.23 (0.000)	22.92 (0.000)	8.67 (0.000)	12.40	8
	GAN	10.18 (0.000)	10.70 (0.000)	24.26 (0.000)	3.66 (0.076)	**1.00** (/)	2.61 (0.187)	18.14 (0.000)	**1.16** (/)	**1.00** (/)	14.75 (0.000)	25.61 (0.000)	2.80 (0.145)	9.65	2
	CGAN	12.14 (0.000)	13.21 (0.000)	25.92 (0.000)	13.03 (0.000)	12.12 (0.000)	5.55 (0.000)	6.79 (0.000)	15.00 (0.000)	15.47 (0.000)	9.35 (0.000)	14.93 (0.000)	17.80 (0.000)	13.44	12
	WGAN	10.93 (0.000)	10.34 (0.000)	17.91 (0.000)	14.44 (0.000)	16.94 (0.000)	15.08 (0.000)	12.27 (0.000)	20.01 (0.000)	22.56 (0.000)	21.57 (0.000)	7.22 (0.000)	24.07 (0.000)	16.11	18
	WGAN-GP	11.42 (0.000)	7.34 (0.000)	15.88 (0.000)	15.59 (0.000)	16.99 (0.000)	15.20 (0.000)	12.12 (0.000)	19.98 (0.000)	22.56 (0.000)	21.59 (0.000)	6.96 (0.000)	24.19 (0.000)	15.82	17
Embedded framework approaches	EasyEn	5.72 (0.000)	5.08 (0.011)	17.64 (0.000)	13.04 (0.000)‘	16.85 (0.000)	14.86 (0.000)	11.75 (0.000)	19.60 (0.000)	22.09 (0.000)	21.27 (0.000)	7.47 (0.000)	24.29 (0.000)	14.97	14
	RUSB	24.33 (0.000)	25.91 (0.000)	8.08 (0.000)	25.79 (0.000)	16.70 (0.000)	14.54 (0.000)	11.75 (0.000)	19.71 (0.000)	22.73 (0.000)	21.53 (0.000)	7.43 (0.000)	24.42 (0.000)	18.58	22
	CSL12	23.45 (0.000)	25.35 (0.000)	6.07 (0.007)	25.36 (0.000)	17.14 (0.000)	15.43 (0.000)	12.26 (0.000)	19.87 (0.000)	14.99 (0.000)	4.45 (0.004)	13.53 (0.000)	17.41 (0.000)	16.33	19
	CSL14	20.63 (0.000)	21.92 (0.000)	8.52 (0.000)	21.79 (0.000)	17.14 (0.000)	15.43 (0.000)	12.26 (0.000)	19.87 (0.000)	12.01 (0.000)	6.00 (0.000)	15.99 (0.000)	14.52 (0.000)	15.50	16
	CSL16	15.78 (0.000)	17.81 (0.000)	13.11 (0.000)	17.83 (0.000)	17.14 (0.000)	15.43 (0.000)	12.26 (0.000)	19.87 (0.000)	10.00 (0.000)	7.81 (0.000)	17.21 (0.000)	13.00 (0.000)	14.77	13

(*continued on next page*)

Table 8 (*continued*)

		LR_BACC	LR_F1	LR_P	LR_R	DT_BACC	DT_F1	DT_P	DT_R	RF_BACC	RF_F1	RF_P	RF_R	AvgR	Score
Post-hoc adjusting	CSC12	23.45 (0.000)	25.35 (0.000)	6.70 (0.007)	25.36 (0.000)	19.76 (0.000)	17.33 (0.000)	12.08 (0.000)	23.23 (0.000)	25.10 (0.000)	23.81 (0.000)	6.54 (0.000)	26.93 (0.000)	19.64	27
	CSC14	17.85 (0.000)	19.78 (0.000)	4.83 (0.257)	19.78 (0.000)	21.31 (0.000)	18.76 (0.000)	12.32 (0.000)	24.69 (0.000)	26.36 (0.000)	25.03 (0.000)	6.97 (0.000)	27.96 (0.000)	18.80	24
	CSC16	13.57 (0.000)	12.37 (0.000)	8.73 (0.000)	16.60 (0.000)	21.99 (0.000)	19.36 (0.000)	11.37 (0.000)	25.41 (0.000)	27.01 (0.000)	25.52 (0.000)	6.75 (0.000)	28.58 (0.000)	18.10	21
	MC12	23.25 (0.000)	25.22 (0.000)	5.84 (0.049)	25.36 (0.000)	14.89 (0.000)	12.62 (0.000)	10.15 (0.000)	17.93 (0.000)	18.66 (0.000)	17.73 (0.000)	5.31 (0.000)	20.54 (0.000)	16.46	20
	MC14	23.25 (0.000)	25.22 (0.000)	5.84 (0.049)	25.36 (0.000)	11.57 (0.000)	3.96 (0.015)	5.37 (0.001)	14.80 (0.000)	13.24 (0.000)	2.00 (0.406)	11.95 (0.000)	16.24 (0.000)	13.23	11
	MC16	24.26 (0.000)	25.53 (0.000)	8.89 (0.000)	25.24 (0.000)	9.04 (0.000)	**1.02** (/)	4.54 (0.013)	13.11 (0.000)	11.00 (0.000)	**1.00** (/)	13.62 (0.000)	14.50 (0.000)	12.64	9
Friedman 0.05		2643.183 (0.000)	2639.37 (0.000)	2561.241 (0.000)	2733.252 (0.000)	2679.773 (0.000)	2581.736 (0.000)	2614.017 (0.000)	2703.313 (0.000)	2747.889 (0.000)	2680.858 (0.000)	2681.960 (0.000)	2723.932 (0.000)		

occur (Arjovsky and Bottou, 2017), which means that to meet the needs of an iterative optimization, the neural network learns significant data distributions, but fails to capture insignificant data distributions. Correctly identifying different sample clusters and performing data generation greatly affect the learning ability and accuracy of the GAN, especially when dealing with this kind of tabular data. How to complete the learning task with high quality, without wasting resources, and how to avoid the reverse promotion of the CI problem, is a problem that needs to be considered when adopting a GAN.

When facing a CI problem under the scenario of credit scoring, it is not advisable to directly adopt a balancing approach at the data or algorithm level. The gap in the number of samples between the classes is not the decisive factor of a CI problem; a CI problem is caused by a variety of reasons, including but not limited to the ones discussed in this section. By using three datasets of credit scoring data to conduct the experiments, some causes of the CI problem have been proved. In fact, although there are many applications of credit scoring, adequate research of financial businesses and analysis of behavioral patterns have always been the most basic and important part of the whole process, which will facilitate the development of feature engineering and can greatly reduce the negative impact of the CI problem.

5.3. Discussion

By analyzing 28 imbalanced data learning approaches in credit scoring. Our work confirms the conclusions of previous studies: First, simple but useful, the random sampling methods (ROS and RUS) and SMOTE are efficient in credit scoring, which confirms the conclusions in previous studies (Xiao et al., 2021; Marqués et al., 2013; García et al., 2012). In Table 10, the rank scores of these methods are all less than 10.0, which means that the performance of them is quite stable in handling CI problem of credit scoring.

Second, our work confirms the pervious finding that the RUS and neighbor cleaning- based sampling method are practical under the circumstances of "IR less than 4.0″(Rangel-Díaz- de-la Vega et al., 2020), which means that they are not universal in credit scoring. RUS get the rank score of 1 and 4 in Germany (IR = 2.33) and Taiwan (IR = 3.51) dataset respectively, however, in GMSC with IR = 13.38, its performance has a significant decline with rank score of 10, and the similar situation also appear on ENN and Tomek. From the experimental result, we can find that in handling CI problem of credit scoring, the effect of sampling methods is related to the size of dataset and IR is a basis of IDLAs selection which should pay more attention in application.

The most revealing finding from our benchmark study is that there is no absolute correlation between the complexity of IDLAs and performance of credit scoring models. In other words, it's not that the more "advanced" the IDLAs are, the better their performance. From GANs' perspective, the credit scoring datasets we use in benchmark are tabular dataset and are similar with(Kang et al., 2022; Zheng et al., 2020; Douzas and Bacao, 2018), but Table 9 shows that GAN(Final_score = 1.5) perform best in all 28 ID- LAs and better than the other three "sophisticated versions" of CGAN(Final_score = 20.0), WGAN(Final_score = 16.0) WGAN-GP(Final_score = 25.5); on the other hand, updated versions of SMOTE(like SMOTE with cleaning strategy or BorderlineSMOTE) not always perform better than naive SMOTE: in Taiwan, naive SMOTE perform better than other "update versions", but in other two datasets, we get opposite conclusion, which prove our finding and make suggestion for the framework design in credit scoring.

Our work first explores the issue of GAN as IDLA, and we verify its practicability and stability through benchmark experiment. Several questions worth exploring in future research are about the stability and effectiveness of a GAN in handling the CI problem of a tabular dataset. First, from the benchmark analysis, GAN performs better than the updated versions (CGAN, WGAN, and WGAN-GP). Therefore, we suggest that in future, related work, the complexity of the GAN need not be overly considered, and the focus should be placed on business research,

Table 9
Final rank score among the results of three credit scoring datasets.

		LR_BACC	LR_F1	LR_P	LR_R	DT_BACC	DT_F1	DT_P	DT_R	RF_BACC	RF_F1	RF_P	RF_R	AvgR	Score
Traditional classification	Original	25.0	28.0	**1.0**	28.0	25.0	29.0	**1.0**	29.0	18.0	18.0	**1.0**	22.0	19	23.0
Resampling approaches	ROS	**1.0**	**1.0**	10.0	16.0	**1.0**	2.0	18.0	9.0	3.0	7.0	16.5	12.0	8	**1.5**
	RUS	5.0	6.0	17.0	10.0	5.0	10.0	21.0	5.0	2.0	4.0	20.0	6.0	9	3.0
	ENN	7.0	8.0	7.0	20.0	11.0	11.0	3.0	15.0	10.0	**1.0**	12.0	16.0	10	5.5
	Tomek	22.0	23.0	2.0	27.0	14.0	15.0	2.0	28.0	12.0	15.0	2.0	19.0	15	13.5
	Nearmiss1	29.0	25.0	28.0	5.0	29.0	27.0	29.0	3.0	20.0	25.0	28.0	**1.0**	21	27.5
	Nearmiss2	28.0	26.0	29.0	17.0	28.0	28.0	28.0	11.0	24.0	27.0	29.0	2.0	23	29.0
	Nearmiss3	13.5	16.0	22.0	18.0	12.0	13.0	23.0	12.0	17.0	20.0	26.0	10.5	17	18.0
Synthesizing techniques	SMOTE	3.0	4.0	13.0	11.0	4.0	4.0	22.0	6.0	7.0	8.0	18.5	14.0	10	5.5
	BS1	9.0	7.0	15.0	13.0	8.0	7.0	24.0	7.0	4.0	9.0	21.0	7.0	11	8.5
	BS2	6.0	5.0	18.0	6.0	6.0	6.0	25.0	4.0	8.0	5.0	15.0	15.0	10	5.5
	SE	11.0	11.0	23.0	**1.0**	9.0	8.0	27.0	**1.0**	6.0	14.0	25.0	3.0	12	10.0
	ST	2.0	2.0	11.0	14.0	2.0	3.0	18.0	8.0	9.0	10.0	18.5	13.0	β	5.5
	ADASYN	8.0	9.0	20.0	4.0	7.0	12.0	26.0	2.0	5.0	12.0	22.0	8.0	11	8.5
	GAN	4.0	3.0	16.0	9.0	3.0	**1.0**	9.0	10.0	**1.0**	6.0	23.0	5.0	8	**1.5**
	CGAN	18.0	17.0	27.0	19.0	23.0	18.0	5.0	24.0	13.0	17.0	11.0	20.0	18	20.0
	WGAN	12.0	15.0	8.0	22.0	17.0	17.0	12.0	18.0	22.0	22.0	8.0	23.0	16	16.0
	WGAN-GP	20.0	19.0	12.0	23.0	22.0	23.0	19.0	23.0	25.0	23.5	9.0	26.0	20	25.5
Embedded framework approaches	EasyEn	10.0	10.0	9.0	21.0	16.0	16.0	13.0	17.0	23.0	21	6.5	24.5	16	16.0
	RUSB	26.0	29.0	3.0	29.0	18.0	19.0	14.0	22.0	26.0	23.5	10.0	24.5	20	25.5
	CSL12	13.5	12.0	6.0	24.0	20.0	21.0	16.0	20.0	16.0	11.0	14.0	18.0	16	16.0
	CSL14	19.0	20.0	21.0	12.0	20.0	21.0	16.0	20.0	15.0	13.0	24.0	9.0	18	20.0
	CSL16	24.0	24.0	26.0	3.0	20.0	21.0	16.0	20.0	19.0	16.0	27.0	4.0	18	20.0
Post-hoc adjusting	CSC12	16.0	13.0	5.0	26.0	24.0	24.0	11.0	25.0	27.0	26.0	3.0	27.0	19	23.0
	CSC14	17.0	18.0	14.0	8.0	26.0	25.0	10.0	26.0	28.0	28.0	5.0	28.0	19	23.0
	CSC16	23.0	21.0	25.0	2.0	27.0	26.0	8.0	27.0	29.0	29.0	4.0	29.0	21	27.5
	MC12	15.0	14.0	4.0	25.0	15.0	14.0	7.0	16.0	21.0	19.0	6.5	21.0	15	13.5
	MC14	21.0	22.0	19.0	15.0	13.0	9.0	4.0	14.0	11.0	2.0	12.0	17.0	13	11.0
	MC16	27.0	27.0	24.0	7.0	10.0	5.0	6.0	13.0	14.0	3.0	16.5	10.5	14	12.0

especially on data preprocessing. Classifiers will positively affect the performance of a dataset that has been processed by filtering and cleaning. Similarly, we believe that on the dataset after the above processing steps, GANs used for data synthesis can better avoid the interference of noise points and exert their powerful fitting effect.

Second, the research on GANs' ability is meaningful and should be paid attention to. Without transforming the feature space of the datasets, as happens with feature engineering, a GAN can autonomously discover and pay attention to the features that have a greater impact on the classification results during training, while at the same time diluting the insignificant features. In this way, the learning ability and efficiency of a GAN will be guaranteed. Also, when considering practical issues like running time or computing resources, whether it still worth using needs to be explored in depth.

Third, the relationship of the authenticity and validity of generated samples is worth studying. For example, in the training process of WGAN, the Wasserstein distance (W), which is shown in Equation (3), can be seen as an indicator of the training process. The smaller W is, the closer the generated data distribution, p_g, is to the real data distribution, p_{data}. However, it is still unknown whether the closer the generated sample distribution is to the true distribution, the more such samples can improve the classification accuracy (especially for minority class samples) of the final classifiers. We believe that this relationship has far-reaching significance for the application of GANs in tabular data. By observing the experimental results of this article, we foresee that GANs have broad application prospects as an imbalanced data learning tool in credit scoring.

6. Conclusion

In the benchmark analysis of this article, we summary imbalanced data learning approaches comprehensively by proposing a new taxonomy according to their working principle. At the same time, we also consider the relatively novel GAN approaches that appear recently. With the experiment of repeated fivefold cross validation, we observe the difference in performance of these approaches solving CI problem and provide a reference for applications of imbalanced data learning approaches in credit scoring.

The contributions of our work are three-folds: (1) We propose a new taxonomy of state-of-the-art IDLAs according to their mechanism to comprehensively analyze various IDLAS in credit scoring. (2) To our best knowledge, we first consider GANs as sample generating tools to handle CI problem of credit scoring in our benchmark experiment. (3) We conduct large-scale benchmark experiment of 28 IDLAs across three real-world credit scoring datasets and draw many conclusions about their application and internal factors that cause CI problem in credit scoring.

Our work has several limitations, which may be addressed in further research. First, the techniques of handling class imbalance problem of credit scoring are developing rapidly, besides GANs, there are also some up-and-coming techniques (e.g., transfer learning and natural language processing technique). Further research may pay attention to these techniques to continuously update our relevant work and make more practical suggestions. Second, as our benchmark are based on public credit scoring datasets from UCI machine learning repository, performance of the selected IDLAs in other datasets from real scenario are not explored. Further research may experiment with IDLAs on more credit scoring datasets collected from other data sources to validate the generalizability of our findings.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was funded by the National Natural Science Foundation of China (Grant Nos. 71731005, 72101073, and 71571059) and the Fundamental Research Funds for the Central Universities (Grant No. JZ2021HGTA0130).

References

Abdou, H. A., & Pointon, J. (2011). Credit scoring, statistical techniques and evaluation crite- ria: A review of the literature. *Intelligent systems in accounting, finance and management, 18*, 59–88.

Ali, A., Shamsuddin, S. M., & Ralescu, A. L. (2013). *Classification with class imbalance problem* (p. 5). Appl: Int. J. Advance Soft Compu.

Arjovsky, M., Bottou, L., 2017. Towards principled methods for training generative adver- sarial networks. arXiv preprint arXiv:1701.04862 .

Aswathi, M., Ghosh, A., & Namboothiri, L. V. (2022). Borda count versus majority voting for credit card fraud detection. *Ubiquitous Intelligent Systems. Springer*, 319–330.

Batista, G. E., Bazzan, A. L., Monard, M. C., et al. (2003). *Balancing training data for automated annotation of keywords: A case study* (pp. 10–18). in: WOB.

Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter, 6*, 20–29.

Brown, I., & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications, 39*, 3446–3453.

Carta, S., Ferreira, A., Recupero, D. R., Saia, M., & Saia, R. (2020). A combined entropy-based approach for a proactive credit scoring. *Engineering Applications of Artificial Intelligence, 87*, Article 103292.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic mi- nority over-sampling technique. *Journal of artificial intelligence research, 16*, 321–357.

Chen, J., Xie, L., Liu, D. H., & XIAO4d, J.,. (2017). Effect analysis of resampling tech- niques on the performance of customer credit scoring models. *DEStech Transactions on Computer Science and Engineering, 12*.

Crone, S. F., & Finlay, S. (2012). Instance sampling in credit scoring: An empirical study of sample size and balancing. *International Journal of Forecasting, 28*, 224–238.

Dastile, X., Celik, T., & Potsane, M. (2020). Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing, 91*, Article 106263.

Domingos, P., 1999. Metacost: A general method for making classifiers cost-sensitive, in: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 155–164.

Douzas, G., & Bacao, F. (2018). Effective data generation for imbalanced learning using con- ditional generative adversarial networks. *Expert Systems with applications, 91*, 464–471.

Engelmann, J., & Lessmann, S. (2021). Conditional wasserstein gan-based oversampling of tabular data for imbalanced learning. *Expert Systems with Applications, 174*, Article 114582.

Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). Smote for learning from im- balanced data: Progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research, 61*, 863–905.

Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2019). Using generative adver- sarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences, 479*, 448–455.

Gangwar, A. K., & Ravi, V. (2019). Wip: Generative adversarial network for oversampling data in credit card fraud detection. *International Conference on Information Systems Security, Springer.*, 123–134.

García, V., Marqués, A. I., & Sánchez, J. S. (2012). Improving risk predictions by preprocessing imbalanced credit data. *International conference on neural information processing, Springer.*, 68–75.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. Advances in neural information pro- cessing systems 27.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C., 2017. Improved training of wasserstein gans. Advances in neural information processing systems 30.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert systems with applications, 73*, 220–239.

Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-smote: A new over-sampling method in imbalanced data sets learning. *International conference on intelligent computing, Springer.*, 878–887.

He, F., Zhang, W., & Yan, Z. (2022). A novel multi-stage ensemble model for credit scoring based on synthetic sampling and feature transformation. *Journal of Intelligent & Fuzzy Systems*, 1–16.

He, H., Bai, Y., Garcia, E.A., Li, S., 2008. Adasyn: Adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), IEEE. pp. 1322–1328.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowl- edge and data engineering, 21*, 1263–1284.

He, H., Zhang, W., & Zhang, S. (2018). A novel ensemble method for credit scoring: Adaption of different imbalance ratios. *Expert Systems with Applications, 98*, 105–117.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intel- ligent data analysis, 6*, 429–449.

Junior, L. M., Nardini, F. M., Renso, C., Trani, R., & Macedo, J. A. (2020). A novel approach to define the local region of dynamic selection techniques in imbalanced credit scoring problems. *Expert Systems with Applications, 152*, Article 113351.

Kang, Y., Chen, L., Jia, N., Wei, W., Deng, J., & Qian, H. (2022). A cwgan-gp-based multi-task learning model for consumer credit scoring. *Expert Systems with Applications, 117650*.

Lei, K., Xie, Y., Zhong, S., Dai, J., Yang, M., & Shen, Y. (2020). Generative adversarial fusion network for class imbalance credit scoring. *Neural Computing and Applications, 32*, 8451–8462.

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. URL *Journal of Machine Learning Research, 18*, 1–5 http://jmlr.org/papers/v18/16-365.

Lenka, S.R., Bisoy, S.K., Priyadarshini, R., Sain, M., 2022. Empirical analysis of ensemble learning for imbalanced credit scoring datasets: A systematic review. Wireless Commu- nications and Mobile Computing 2022.

Lessmann, S., Baesens, B., Seow, H. V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research, 247*, 124–136.

Ling, C. X., & Sheng, V. S. (2008). Cost-sensitive learning and the class imbalance problem. *Encyclopedia of machine learning, 2011*, 231–235.

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

Liu, Y., Li, Z., Zhou, C., Jiang, Y., Sun, J., Wang, M., et al. (2019). Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering, 32*, 1517–1528.

López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classifi- cation with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information sciences, 250*, 113–141.

Louzada, F., Ara, A., & Fernandes, G. B. (2016). Classification methods applied to credit scoring: Systematic review and overall comparison. *Surveys in Operations Research and Management Science, 21*, 117–134.

Loyola-González, O., Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A., & García-Borroto, M. (2016). Study of the impact of resampling methods for contrast pattern based classifiers in imbalanced databases. *Neurocomputing, 175*, 935–947.

Makki, S., Assaghir, Z., Taher, Y., Haque, R., Hacid, M. S., & Zeineddine, H. (2019). An exper- imental study with imbalanced classification approaches for credit card fraud detection. *IEEE Access, 7*, 93010–93022.

Mani, I., & Zhang, I. (2003). knn approach to unbalanced data distributions: A case study in- volving information extraction. *Proceedings of workshop on learning from imbalanced datasets, ICML.*, 1–7.

Marqués, A. I., García, V., & Sánchez, J. S. (2013). On the suitability of resampling techniques for the class imbalance problem in credit scoring. *Journal of the Operational Research Society, 64*, 1060–1070.

de Melo Junior, L.S., Nardini, F.M., Renso, C., de Macêdo, J.A.F., 2019. An empirical com- parison of classification algorithms for imbalanced credit scoring datasets, in: 2019 18th IEEE international conference on machine learning and applications (ICMLA), IEEE. pp. 747–754.

Mirza, M., Osindero, S., 2014. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 .

Mushava, J., & Murray, M. (2022). A novel xgboost extension for credit scoring class-imbalanced data combining a generalized extreme value link and a modified focal loss function. *Expert Systems with Applications, 202*, Article 117233.

Niu, K., Zhang, Z., Liu, Y., & Li, R. (2020). Resampling ensemble model based on data distribution for imbalanced credit risk evaluation in p2p lending. *Information Sciences, 536*, 120–134.

Ounacer, S., Jihal, H., Bayoude, K., Daif, A., & Azzouazi, M. (2020). Handling imbalanced datasets in the case of credit card fraud. *International Conference on Advanced Intelligent Systems for Sustainable Development, Springer.*, 666–678.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Sanabila, H.R., Jatmiko, W., 2018. Ensemble learning on large scale financial imbalanced data, in: 2018 International Workshop on Big Data and Information Security (IWBIS), IEEE. pp. 93–98.

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2009). Rusboost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 40*, 185–197.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for clas- sification tasks. *Information processing & management, 45*, 427–437.

Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International journal of pattern recognition and artificial intelligence, 23*, 687–719.

Thabtah, F., Hammoud, S., Kamalov, F., & Gonsalves, A. (2020). Data imbalance in classifi- cation: Experimental evaluation. *Information Sciences, 513*, 429–441.

Tomek, I. (1976). Two modifications of cnn. IEEE Trans. *Systems, Man and Cybernetics, 6*, 769–772.

Rangel-Díaz-de-la Vega, A., Villuendas-Rey, Y., Yanez-Marquez, C., Camacho-Nieto, O., & Lopez-Yanez, I. (2020). Impact of imbalanced datasets preprocessing in the performance of associative classifiers. *Applied Sciences, 10*, 2779.

Wang, J., Yao, L., 2022. Unrolled gan-based oversampling of credit card dataset for fraud detection, in: 2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), IEEE. pp. 858–861.

Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, 408–421.

Xiao, J., Wang, Y., Chen, J., Xie, L., & Huang, J. (2021). Impact of resampling methods and classification models on the imbalanced credit scoring problems. *Information Sciences, 569*, 508–526.

Xiao, J., Zhou, X., Zhong, Y., Xie, L., Gu, X., & Liu, D. (2020). Cost-sensitive semi-supervised selective ensemble model for customer credit scoring. *Knowledge-Based Systems, 189*, Article 105118.

Yao, G., Hu, X., Zhou, T., & Zhang, Y. (2022). Enterprise credit risk prediction using supply chain information: A decision tree ensemble model based on the differential sampling rate, synthetic minority oversampling technique and adaboost. *Expert Systems, e12953*.

Zhang, T., & Chi, G. (2021). A heterogeneous ensemble credit scoring model based on adaptive classifier selection: An application on imbalanced data. *International Journal of Finance & Economics, 26*, 4372–4385.

Zheng, M., Li, T., Zhu, R., Tang, Y., Tang, M., Lin, L., et al. (2020). Conditional wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification. *Information Sciences, 512*, 1009–1023.

Zhu, B., Pan, X., & vanden Broucke, S., Xiao, J.,. (2022). A gan-based hybrid sampling method for imbalanced customer classification. *Information Sciences*.