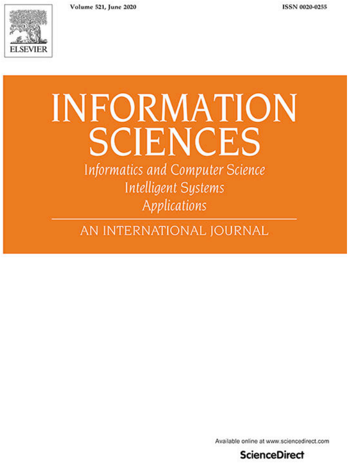
Journal Pre-proofs



An Efficiency Curve for Evaluating Imbalanced Classifiers Considering Intrinsic Data Characteristics: Experimental Analysis

Xiangrui Chao, Gang Kou, Yi Peng, Alberto Fernández

PII:	S0020-0255(22)00640-5
DOI:	https://doi.org/10.1016/j.ins.2022.06.045
Reference:	INS 17675
To appear in:	*Information Sciences*
Received Date:	25 September 2021
Revised Date:	9 June 2022
Accepted Date:	11 June 2022

Please cite this article as: X. Chao, G. Kou, Y. Peng, A. Fernández, An Efficiency Curve for Evaluating Imbalanced Classifiers Considering Intrinsic Data Characteristics: Experimental Analysis, *Information Sciences* (2022), doi: https://doi.org/10.1016/j.ins.2022.06.045

An Efficiency Curve for Evaluating Imbalanced Classifiers Considering Intrinsic Data Characteristics: Experimental Analysis

Xiangrui Chao[a], Gang Kou[b], Yi Peng[c,*] Alberto Fernández[d]

[a]Business School, Sichuan University, Chengdu 610065, China
[b]School of Business Administration, Southwestern University of Finance and Economics, Chengdu 611130, China
[c]School of Management and Economics, University of Electronic Science and Technology of China, Chengdu, 611731, China
[d]Department of Computer Science and Artificial Intelligence, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Granada 18071, Spain

*Corresponding author: Yi Peng, pengyi@uestc.edu.cn.

Abstract: Balancing the accuracy rates of the majority and minority classes is challenging in imbalanced classification. Furthermore, data characteristics have a significant impact on the performance of imbalanced classifiers, which are generally neglected by existing evaluation methods. The objective of this study is to introduce a new criterion to comprehensively evaluate imbalanced classifiers. Specifically, we introduce an efficiency curve that is established using data envelopment analysis without explicit inputs (DEA-WEI), to determine the trade-off between the benefits of improved minority class accuracy and the cost of reduced majority class accuracy. In sequence, we analyze the impact of the imbalanced ratio and typical imbalanced data characteristics on the efficiency of the classifiers. Empirical analyses using 68 imbalanced data reveal that traditional classifiers such as C4.5 and the k-nearest neighbor are more effective on disjunct data, whereas ensemble and undersampling techniques are more effective for overlapping and noisy data. The efficiency of cost-sensitive classifiers decreases dramatically when the imbalanced ratio increases. Finally, we investigate the reasons for the different efficiencies of classifiers on imbalanced data and recommend steps to select appropriate classifiers for imbalanced data based on data characteristics.

Keywords: classification; imbalanced dataset; data intrinsic characteristics; assessment metrics; efficiency.

1. Introduction

When class distribution of a dataset is highly skewed, the dataset becomes imbalanced. The classification of imbalanced data has received increasing attention in the field of data mining (Khorshidi and Aickelin, 2021; Xie et al., 2020). Numerous algorithms have been developed to classify imbalanced data, including data sampling (Kang et al., 2017; Sáez et al., 2015), cost-sensitive learning (Chao and Peng, 2018), and hybrid approaches (Ng et al., 2018). These classification methods, which can improve the classification accuracy in imbalanced data, are called imbalanced classifiers or imbalanced class solutions. One difficulty in imbalanced classification is that most classifiers are biased toward the majority class and perform poorly in the minority class (Thabtah et al., 2020; Luque et al., 2019).

The essential task of binary imbalanced classification is to *improve the accuracy of the minority data while not considerably reducing the accuracy of the majority data* (Thabtah et al., 2020; Chao and Peng, 2018). Wang and Yao (2012) argued that the assessment of the minority class is more important than that of majority class in real-life applications. Many frequently used measures, such as the geometry-mean and F-measure, are not comprehensive indexes that reflect the performance of imbalanced classifiers, because the accuracy of the minority class cannot be accurately represented (He and Garcia, 2009). Therefore, evaluating imbalanced classifiers is an important research direction for imbalanced learning (He and Garcia, 2009; Roy et al., 2019).

Various decision-making approaches that combine multiple assessment metrics have been developed to assess classifiers. Kou et al. (2014) proposed a multi-criteria decision-making method to analyze the performance of classifiers. Peng et al. (2011) proposed a method for determining the weights of these performance metrics. Several curve-based metrics have also been used to evaluate imbalanced classifiers (He and Garcia, 2009), including the cost curve, precision recall (PR), and receiver operating characteristics (ROC). The average performance measures of the majority and minority classes are widely used, such as the area under the ROC curve (AUC) (López et al., 2013). Brzezinsky (2018) proposed a visual technology to analyze the performance differences of 22 measure indices. However, some studies have shown that these metrics are biased in imbalanced datasets (Luque et al., 2019; Mullick et al, 2020).

In addition to the imbalance in the number of data samples, data characteristics such as small disjuncts, overlapping, noisy data, and data shift have a strong influence on the performance (López et al., 2013; 2014). The classifier mechanism is often disturbed by these individual examples in the dataset. For example, in a classical support vector machine (SVM), the classification boundary is more biased toward majority class data and reduces the predictive ability of minority data on an imbalanced overlapping dataset. Because there are differences in the performance of classifiers on different types of data characteristics, choosing a reasonable classification algorithm for these datasets requires a classification of the intrinsic nature of the datasets.

For binary imbalanced classification, an increase in the accuracy of minority data is often accompanied by a decrease in the accuracy of majority data (López et al., 2013; Thai-Nghe et al., 2011). Therefore, the amount of accuracy loss of the majority class that should be allowed to improve the accuracy of minority data should be investigated. For example, suppose that we need to compare two classifiers: the accuracy rates of the majority and minority classes of one classifier are 80% and 75% and those the other classifier are 86% and 69%, respectively. Which classifier is more effective? Answering this question can improve our understanding of the effectiveness of an imbalanced classifier and allow us to compare any imbalanced classifiers. García et al. (2009) proposed an index that combines sensitivity, specificity, and geometric means to measure classifiers.

Although considerable research has been conducted on evaluating classification algorithms, few studies have evaluated imbalanced algorithms from the perspective of classification efficiency. In economics, *efficiency* is the ratio of the benefits to the costs of completing a job. An efficient economic system can achieve more benefits at a lower cost. Motivated by this concept, we treat the accuracy of the majority class as the cost and the accuracy of the minority class as the benefit, and define an algorithm as efficient when it has relative ratios of benefit and cost. Data envelopment analysis (DEA) is a data-driven multi-objective decision-making method that aims to balance different evaluation indicators by constructing an efficiency curve. Based on the nature of the ROC curve, Zheng and Padmanabhan (2007) used DEA method to combine different classifiers to obtain better classification performance. They concluded that the relative efficiency curve can be used to evaluate the classifiers. In this study, we synthesized different evaluation metrics as the output of the model to evaluate the efficiency of a classifier using DEA without explicit inputs (DEA-WEI) (Liu et al., 2011), which is a new attempt to evaluate algorithms.

Recognizing the significant impact of data characteristics on imbalanced classifiers, classifiers were evaluated in this study under typical imbalanced data characteristics in three steps. First, we analyzed the impact of the imbalance ratio and different data characteristics on the efficiency of the classifiers. Second, using real-world examples, we illustrated how the proposed evaluation process can be used to evaluate the effectiveness of a set of classifiers. We also examined the influence of the dimensionality of the dataset, noise intensity, and extreme outliers on the classification efficiency of the artificial datasets. Finally, we investigated why classifiers have different efficiencies on imbalanced data and recommended steps for selecting appropriate classifiers for imbalanced data based on data characteristics.

The contributions of this research are two-fold:

1) A new approach was proposed to evaluate the effectiveness of imbalanced classifiers by developing an efficiency curve that measures the trade-off between benefits and costs;

2) A thorough analysis was conducted to determine the different efficiencies of classifiers and a three-step process was suggested for selecting efficient classifiers for datasets with various imbalance ratios and characteristics.

The remainder of this study is organized as follows: Section 2 introduces imbalanced classification methods, intrinsic characteristics of imbalanced data, and performance metrics. Section 3 proposes a new approach for analyzing and evaluating imbalanced classifiers using the DEA-WEI model. Section 4 presents a large collection of benchmark datasets to illustrate the proposed approach. Section 5 discusses the experimental results and theoretical insights, and Section 6 concludes the study.

2. Related works and preliminary knowledge

This section introduces some classic imbalanced classification methods, intrinsic characteristics of imbalanced data, and performance metrics for imbalanced classifiers.

2.1 Imbalanced classification methods

Several approaches have been developed to classify imbalanced datasets. These approaches can be categorized into three groups.

Data preprocessing: To address imbalanced data characteristics, three types of sampling methods have been used to ensure balanced training datasets (Kang et al., 2017): oversampling (Fernández et al., 2018), under-sampling (Tsai et al., 2019), feature reduction (Sun et al., 2022) and hybrid sampling (Razavi-Far et al., 2017). One drawback of sampling techniques is the structural change in the original datasets, which may cause overfitting in the classification process. In imbalanced classification, ensemble methods (Song et al., 2018; Ng et al., 2018), which integrate different classifiers to improve the classification results, are often combined with data preprocessing techniques. Chouhan and Rathore (2021) proposed an oversampling method based on a generative adversarial network and used it for software aging-related bug prediction. Du et al. (2021) proposed a label enhancement method in which a numerical label is introduced instead of the original logical label. Sowah et al. (2021) proposed a new hybrid sampling technique that uses a combination of the cluster undersampling technique to undersample the majority instances, and an oversampling technique derived from sigma nearest oversampling based on convex combination for minority instances.

Algorithmic modification: Many standard classification algorithms, such as decision trees (Lomaxand and Vadera, 2013), have been adapted for imbalanced datasets using the concept of misclassification cost (Chao and Peng, 2018; Chao et al., 2021). Maurya and Toshniwal (2018) studied class-imbalanced learning using large-scale sparse data in a distributed setting. Richhariya and Tanveer (2020) introduced prior information regarding data characteristics into SVMs and proposed a reduced universum twin SVM. Their method provided balanced data by prior information, and the algorithm time complexity was relatively lower. Fu et al. (2022) proposed a system by combining the advantages of ν-SVM and asymmetric LINEX loss function. This function is used to allocate different costs to each instance. Elyan et al. (2021) converted an imbalanced binary classification into a balanced multiclass classification system. Their approach clusters the multiclass data into multiple subcategories, in which each subcategory is equivalent to the few-class data to reduce the dataset imbalance.

Cost-sensitive learning: Cost-sensitive learning is a combination of the former two methods. A cost is introduced to the traditional classification, such as the imbalance ratio (Chao and Peng, 2018), misclassification costs (Thai-Nghe et al, 2011), test costs (Lomax and Vadera, 2013), and penalty coefficients (Veganzones and Séverin, 2018). These cost-sensitive classifiers use a cost ratio, which measures the misclassification of the majority and minority classes, as an input parameter. One disadvantage of cost-sensitive classifiers is the difficulty in determining the optimal cost ratio. Chen et al. (2021) introduced cost-sensitive positive and unlabeled learning, and imposed different misclassification costs on different classes. Wang et al. (2021) concluded that existing methods are unable to determine precise misclassification cost values. They employed the F-measure to compute cost information and proposed a cost-sensitive hypergraph structure learning method and F-measure optimization to address imbalance problems. Li et al. (2021) proposed an adaptive weighted cross-entropy loss in conjunction with the Jaccard distance to address the exceedingly foreground-background imbalance in road crack detection.

2.2 Intrinsic characteristics of imbalanced datasets

Imbalanced datasets have been widely used in real-world applications. The characteristics of imbalanced datasets and the performance of algorithms on such datasets have been intensively studied in the field of data mining. Thabtah et al. (2020) measured the functional relationship between accuracy and data imbalance ratio. Using four UCI datasets, they concluded that when the ratio of data instances of majority and minority data was 50%:50%, algorithms always performed the best on evaluation metrics. Luque et al. (2019) empirically analyzed the functional relationship between the imbalance ratio and performance metrics, and proposed a new measure of the imbalance ratio. Barella et al. (2021) proposed complexity measures for imbalanced datasets. They used the distribution of data at the borders and overlapping regions as a measure of dataset complexity. Further, they proposed some guidance for selecting data preprocessing methods to reduce the complexity of the data. Vuttipittayamongkol et al. (2021) investigated the effect of class overlap on classification accuracy and found that the performance of the algorithm deteriorated with varying degrees of class overlap.

Because the intrinsic characteristics of imbalanced datasets have a significant impact on the performance of classifiers (López et al., 2013), their effects on different categories of classifiers should be analyzed (Chao and Peng, 2018). Through visualization technology (e.g., t-distributed stochastic neighbor embedding (t-SNE) dimensionality reduction proposed by Maaten and Hinton (2008)), this study summarizes the characteristics of imbalanced data into scarcity and disjunct, overlapping, and noisy (López et al., 2013; López et al., 2014). Scarcity refers to a rare class of data that most classifiers cannot recognize. Disjunction is a special case of scarcity, characterized by scattered minority data surrounded by majority data. Overlapping refers to a situation in which two classes are evenly mixed and distributed in one area. Noisy refers to data points that are far away from the concentrated area to which they belong, and the two classes have a clear boundary. In Appendix A, 12 KEEL and UCI datasets are used as examples to illustrate these three types of imbalanced data characteristics.

He and Ma (2013) and Fernández et al. (2018) provided detailed descriptions of the characteristics of imbalanced data and different approaches for improving classifiers learned from imbalanced data. Moreover, methods can be used for the division of data characteristics, such as k-nearest examples and kernel functions (Napierala and Stefanowski, 2016).

2.3 Performance metrics for imbalanced classification

Ferri et al. (2009) summarized the most widely used performance metrics for classification, including the true-positive, true-negative, false-positive, and false-negative rates, ROC value, and F-measure. In imbalanced classification, the increase in the accuracy rate of the minority class is usually accompanied by a decrease in the accuracy rate of the majority class. Thus, the AUC and mean (He and Garcia, 2009; López et al., 2013) have been utilized to assess the overall performance of imbalanced classifiers. Specialized metrics based on different insights have also been introduced (Wang and Yao, 2012; Thai-Nghe et al., 2011).

In this study, the minority class is defined as the positive class, and the majority class is defined as the negative class. N and P denote the total number of negative and positive examples, respectively. Table 1 lists the confusion matrix for the four classification outcomes.

Table 1 Standard binary confusion matrix.

		Predicted results	
		Positive	*Negative*
True class	*Positive(P)*	True Positive (TP)	False Negative (FN)
	Negative(N)	False Positive (FP)	True Negative (TN)

The following definitions are commonly used in performance metrics for classification algorithms: $Sensitivity = Recall = \frac{TP}{TP+FN}$ and $Specificity = \frac{TN}{TN+FP}$;

$Precision = \frac{TP}{TP+FP}$.

Several composite metrics have been developed based on these metrics. For instance, F-measure is a harmonic average of precision and recall, that is, $F - measure = \frac{(1+\beta)^2 \cdot Precision \cdot Recall}{\beta^2 (Precision + Recall)}$, where β is a coefficient that adjusts the relative importance of precision and recall. Other commonly used composite metrics include ROC and PR curves. The ROC curve is represented and plotted using the TP and TN rates computed by setting different thresholds of the parameters. It provides a visual representation of the trade-off between the TP and FP rates. The PR curve is plotted by precision and recall, and is sensitive to data characteristics (He and Garcia, 2009).

The average indices are often considered robust when assessing the overall performance of imbalanced classifiers, such as *Gmean*.

$$Gmean = \sqrt{Specificity \times Sensitivity} .$$

Gmean is a function of $\frac{N}{P}$ and $\frac{P-FN}{P}$ (Maurya and Toshniwal, 2018; Brzezinsky, 2018).

The AUC was approximately estimated using an algebraic mean of the sensitivity and specificity. The ROC can provide a visual representation of the relative trade-offs between the benefits (reflected by true positives) and costs (reflected by false positives) of classification at different parameter thresholds. This metric can be regarded as a “benefit-cost” curve. The AUC value is often calculated as the mean value of the accuracy of two classes and serves as the threshold for the average performance of a classification algorithm.

In imbalanced classification, misclassifying a positive record as a negative one is normally more expensive than misclassifying a negative one. For example, in credit card fraud detection, misclassifying fraud behavior as a normal transaction causes a financial institution to misclassify normal behavior as fraud. The cost of misclassification includes balancing and domain costs (Siers and Islam, 2018). Appendix B provides the definitions of balancing and domain costs.

In fact, the maximum *AUC* is equal to the minimum total misclassification cost under certain conditions (see lemma in Appendix C). This shows that there is a linear relationship between the total misclassification cost and AUC, that is, the algebraic mean of the specificity and sensitivity. Thus, we selected one of them as the evaluation indicator in our model (subsection 4.3).

García et al. (2009) proposed a combination index, *IBA*, measured as $IBA = (1 + Sensitivity - Specificity) \cdot Gmean^2$. Conversely to IBA, this study uses the *efficiency curve* between basic indicators instead of mixing them to evaluate imbalanced classifiers. The two methods are compared in Subsections 5.3.

3. **Novel approach to assess the performance of imbalanced classifiers using DEA**

In this study, a theoretical framework was constructed to explain why some classifiers outperform others in imbalanced classification problems. As aforementioned, improving the accuracy of the minority class sacrifices the accuracy of the majority class, and it is difficult to determine the optimal level of accuracy for the minority class. Based on this observation, we propose to use the DEA model to construct an efficiency curve based on the performance of selected algorithms and determine the efficiency of an algorithm based on the distance between the algorithm and the efficiency curve.

DEA (Charnes et al., 1978) is a data-driven approach for assessing efficiency in a wide range of applications. It evaluates the effectiveness of comparable units using linear programming based on the number of cost (input) and benefit (output) indicators. Its ability to analyze efficiency can be applied to evaluation of classifiers. The challenge here is to determine the inputs and outputs of a DEA model. This section illustrates the process of establishing a DEA model to analyze and evaluate imbalanced classifiers.

3.1 DEA

The units measured by DEA are called decision-making units (DMUs). The algorithms to be evaluated are the DMUs. DEA treats each DMU as a target or alternative and assesses its performance on multiple criteria (i.e., performance metrics in imbalanced learning in Subsection 2.3). The inputs are expenditures or resources (costs) that are put into a production process and the outputs are goods or services (benefits) that are obtained from the production process. These inputs and outputs are set according to the different evaluation objectives for different applications.

Model (1) is a DEA model that evaluates the production efficiency of DMUs with multiple inputs and outputs (Liu et al., 2011) :

$$\begin{aligned} &Max \quad e_{j_0} \\ &S.t. \quad e_j < 1 \end{aligned} \qquad (1)$$

where $e_j = \frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_r x_{rj}}$, $j_0 \in \{1, 2, ..., n\}$ is a DMU with s number of outputs and m number of inputs; x_{rj} and y_{rj} are inputs and outputs of the DEA model, respectively; v_r and u_r are unknown weight variables of inputs and outputs, respectively, which should be optimized. This model aims to determine whether a DMU can obtain a higher output with as few inputs as possible.

Model (1) can be transformed into model (2) using the Charnes–Cooper transformation (Charnes et al., 1978).

$$\begin{aligned} &Max \quad \sum_{r=1}^{s} u_r y_{rj_0} \\ &S.t. \quad \sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} w_i x_{ij} \le 0 ; \sum_{i=1}^{m} w_i x_{ij0} = 1 \end{aligned} \qquad (2)$$

Model (2) can be solved using dual programming (3):

$$\begin{aligned} &Min \quad \theta \\ &S.t. \quad \sum_{j=1}^{n} \lambda_j y_{rj} \ge y_{rj_0} \ ; r = 1, 2, ..., s; \\ &\qquad \sum_{j=1}^{n} \lambda_j x_{ij_0} \le x_{ij_0} \ ; i = 1, 2, ..., m; \\ &\qquad \lambda_j \ge 0 ; j = 1, 2, ..., n. \end{aligned} \qquad (3)$$

Generally, the goal of a DEA evaluation is input- or output-driven, and the objective function of a

DEA can be set based on the evaluation goal. The input-driven approach tends to maximize input savings under certain output constraints, whereas the output-driven approach maximizes the output under certain input constraints.

3.2 DEA-WEI and benchmarking model

The original DEA model was developed for a production process in which inputs and outputs are certain. For instance, the inputs can be labor compensation, raw materials, and fixed assets, and the outputs are products or services. In some situations, datasets are provided without explicit inputs, or the original input–output data cannot be easily identified. Liu et al. (2011) proposed DEA-WEI for performance evaluation. The DEA-WEI model requires no calculation of the weights of outputs, which are often controversial (Liu et al., 2011). There are no explicit inputs in an imbalanced classification evaluation. The performance metrics can be treated as outputs and different sets of performance metrics can be selected based on the purpose of the classifier. One of the challenging issues is determining the weights of different metrics. DEA-WEI is can solve the problem because it does not require to determine the weights of the outputs.

Let $\{Y_j \mid j = 1, 2, ..., n\}$ be the group of data in $\Re_+^s$. The smallest closed convex and free-disposal attainable set (AS) that contains observations is defined by Liu et al. (2011):

$$AS = \left\{ Y \middle| Y \leq \sum_{j=1}^{n} \lambda_j Y_j, \sum_{j=1}^{n} \lambda_j = 1, \lambda_j \geq 0 \right\}$$

The DEA-WEI model (4) can be used to measure the relative efficiency of the observation (Y_0) based on the elements in the AS:

$$\begin{aligned} \theta^* &= Max \quad \theta \\ S.t. \quad & \sum_{j=1}^{n} \lambda_j Y_j \geq \theta Y_0; \\ & \sum_{j=1}^{n} \lambda_j = 1, \lambda_j \geq 0; \\ & j = 1, 2, ..., n. \end{aligned} \qquad (4)$$

Compared to the classic DEA model (model (1)), the DEA-WEI model (model (4)) does not explicitly include the input variables in the attainable set. In the DEA-WEI model, a bounded convex set is constructed as follows:

Let $P=\{(X,Y)\}$ be a bounded production possibility set (PPS), which is a free-disposal and closed convex technology set. The projection of all outputs is:

ASI= $\{Y$: There is X such that $(X,Y)\in P\}$ defines a bounded closed convex and free-disposal attainable set.

Let $\{(X_i, Y_i) | i = 1, 2, ..., n\}$ represent a group of input and output data. Subsequently, a bounded closed convex and free-disposal attainable set ASII can be defined as follows:

$$\text{ASII}=\left\{ F \leq \sum_{j=1}^{n} \lambda_j \frac{Y_j}{X_j}, \sum_{j=1}^{n} \lambda_j = 1, \lambda_j \geq 0 \right\}$$

where Y_j / X_j is the ratio of input and output variables. Thus, the variables in the DEA-WEI model are ratios rather than the raw input and output data.

$$
\begin{aligned}
&\theta^{*} = Max \quad \theta \\
&S.t. \quad \sum_{j=1}^{n} \lambda_j Y_{rj} \geq \theta Y_{r0};\, r \in E \\
&\qquad \sum_{j=1}^{n} \lambda_j = 1, \lambda_j \geq 0; \\
&\qquad j = 1, 2, \ldots, n.
\end{aligned}
\tag{5}
$$

where $Y_{rj} = \frac{Y_j}{X_r}$ are quotients.

For N DMUs, let $X^r = (x_1^{\,r}, x_2^{\,r}, \ldots, x_m^{\,r})$ and $Y^r = (y_y^{\,r}, y_2^{\,r}, \ldots, y_s^{\,r})$ be the inputs and outputs of the r^{th} DMU, respectively. Let $e_{ij}^{r} = y_i^r / x_j^r$ be the evaluation index. Then, the DEA-WEI model can be formulated as the following performance score model (6):

$$
\begin{aligned}
&Max \quad \sum w_{ij} e_{ij}^{0} \\
&S.t. \quad \sum w_{ij} e_{ij}^{r} \leq 1, \quad r = 1, \ldots, n \\
&\qquad w_{ij} \geq 0, \quad i = 1, 2, \ldots, m. \quad j = 1, 2, \ldots, s.
\end{aligned}
\tag{6}
$$

where w_{ij} denotes the performance score. The traditional multiplier output-oriented DEA model can be transformed into two DEA-WEI models.

The DEA benchmark is an efficiency curve that provides a reference on where and by how much an inefficient DMU should be improved to achieve full efficiency. If a DMU is on the efficiency curve, it is efficient or relatively efficient; otherwise, it is inefficient, and a DEA benchmark model can be used to determine the closest point on the curve as the direction for the efficiency improvement of the DMU (Cook et al., 2019).

3.3 DEA-WEI for imbalanced classifier evaluation

This section introduces a DEA-WEI benchmark model for an imbalanced classifier evaluation.

Let $\partial(AS)$ be a technically efficient AS bound. The characterization of $\partial(AS)$ used in the formulation of the benchmarking models is (Ruiz et al., 2015):

$$
\partial(AS) = \left\{ (X, Y) \in \Re_{+}^{m+s} \left| \begin{array}{l}
Y = \sum_{k \in E} \lambda_k Y_k, \sum_{k \in E} \lambda_k = 1, \lambda_k \geq 0; \\
uY + u_0 + d_k = 0; k \in E; \\
u_r y_{rj} \geq 1; r = 1, 2, \ldots, s; \\
d_k \leq M b_k; k \in E; \\
\lambda_k \leq M(1 - b_k); k \in E; \\
d_k, \lambda_k \geq 0; b_k \in [0,1]; k \in E; u_0 \in \Re; M \; is \quad big
\end{array} \right. \right\}
\tag{7}
$$

Without loss of generality, we assume that p_r is one of the output indicators in the DEA-WEI model and that a DMU is inefficient at this index. If this DMU wants to achieve relative efficiency, p_r should be increased or decreased to $p_r^{\,g}$ in $\partial(AS)$. $p_r^{\,g}$ is the target of p_r. For the jth DMU, the objective function of the optimal $p_r^{\,g}$ is the L_1 norm, as follows.

$$
\begin{aligned}
&Min \quad \sum_{j \in E} \sum_{r=1}^{s} \left\| p_{rj} - p_{rj}^{\,g} \right\|_1 \\
&S.t. \quad p_{rj}^{g} \in [0,1]
\end{aligned}
\tag{8}
$$

Let $Y_j = (p_{1j}, p_{2j}, \ldots, p_{sj})^T$ be the output index vector for the j^{th} DMU (where $j \in G$ is the total number of DMUs). The DEA-WEI benchmarking model can be described as follows:

$$\begin{array}{lll}
Min & \sum\limits_{r\in E}\sum\limits_{j=1}^{s}\left\| p_{rj} - p_{rj}{}^{g} \right\|_1 & \\
S.t. & \sum\limits_{k\in E}\lambda_k p_{rk} = p_{rj} + x_{rj};\, j\in G; r = 1,2,\ldots,s & (9-1) \\
 & \sum\limits_{k\in E}\lambda_k = 1; & (9-2) \\
 & uY_k + u_0 + d_k = 0; k\in E; & (9-3) \\
 & u_r p_{rj} \geq 1; j\in G; r = 1,2,\ldots,s; & (9-4) \\
 & d_k \leq Mb_k; k\in E; & (9-5) \\
 & \lambda_r \leq M(1-b_k); k\in E; & (9-6) \\
 & b_k \in [0,1]; p_{rj}^{g} \in [0,1] & \\
 & d_k \geq 0; k\in E; \lambda_k \geq 0; r = 1,2,\ldots,s; u_0, x_{rj} \in \Re; & \\
 & M \quad is \quad big \quad positive. &
\end{array} \qquad (9)$$

where E is the set of extremely efficient DMUs (in the Pareto sense) of the AS and M is a large positive number.

In model (9), constraints (9-1) and (9-2) ensure that the optimal solution is within AS. The supporting hyperplanes contain facets of the Pareto frontier of AS, whose coefficients are strictly positive by means of (9-3) and (9-4). Constraints (9-5) and (9-6) guarantee that the benchmark is on the Pareto-efficient frontier of $\partial(AS)$. As far as a DMU is effective, it is distributed on the same facet of the AS Pareto frontier, because all effective DMUs are located at a common supporting hyperplane of the AS.

For a linear solution, the objective function of model (9) can be replaced by a linear transformation $\left\| p_{rj} - p_{rj}{}^{g} \right\|_1 = h_{rj} + l_{rj}, p_{rj} - p_{rj}{}^{g} = h_{rj} - l_{rj}, j\in G; r=1,2,\ldots,s$, where h_{rj} and l_{rj} are non-negative real numbers.

$$\begin{array}{lll}
Min & \sum\limits_{r\in E}\sum\limits_{j=1}^{s} h_{rj} + l_{rj} & \\
S.t. & p_{rj}{}^{g} + h_{rj} - l_{rj} = p_{rj};\, j\in G; r = 1,2,\ldots,s & (10-1) \\
 & \sum\limits_{k\in E}\lambda_k p_{kj} = p_{rj} + x_{rj};\, j\in G; r = 1,2,\ldots,s & (10-2) \\
 & \sum\limits_{k\in E}^{s}\lambda_k = 1; & (10-3) \\
 & uY_k + u_0 + d_k = 0; k\in E; & (10-4) \\
 & u_r y_{rj} \geq 1; r = 1,2,\ldots,s; & (10-5) \\
 & d_k \leq Mb_k; k\in E; & (10-6) \\
 & \lambda_k \leq M(1-b_k); k\in E; & (10-7) \\
 & b_j \in [0,1]; p_{rj}^{g} \in [0,1] & \\
 & h_{rj}, l_{rj}, d_j \geq 0; j\in E; \lambda_r \geq 0, r = 1,2,\ldots,s; u_0, x_{rj} \in \Re; & \\
 & M \quad is \quad big \quad positive. &
\end{array} \qquad (10)$$

Model (11) reformulates model (10) by replacing condition (10-1) with $p_{rj}{}^{g} + h_{rj} - l_{rj} = p_{rj}, j\in G; r=1,2,\ldots,s.$:

$$
\begin{aligned}
& Min \quad \sum_{r \in E} \sum_{j=1}^{s} h_{rj} + l_{rj} \\
& S.t. \quad \sum_{k \in E} \lambda_k p_{kj} = p_{rj}{}^{g} + h_{rj} - l_{rj} + x_{rj}; j \in G; r = 1, 2, ..., s && (11-1) \\
& \qquad \sum_{k \in E}^{s} \lambda_k = 1; && (11-2) \\
& \qquad uY_k + u_0 + d_k = 0; k \in E; && (11-3) \qquad (11) \\
& \qquad u_r y_{rj} \geq 1; r = 1, 2, ..., s; && (11-4) \\
& \qquad d_k \leq Mb_k; k \in E; && (11-5) \\
& \qquad \lambda_k \leq M(1 - b_k); k \in E; && (11-6) \\
& \qquad b_k \in [0,1]; p_{rj}^{g} \in [0,1] \\
& \qquad h_{rj}, l_{rj}, d_k \geq 0; k \in E; \lambda_k > 0, r = 1, 2, ..., s; u_0, x_{rj} \in \Re; \\
& \qquad M \quad is \quad big \quad positive.
\end{aligned}
$$

Model (11) can be solved by replacing the LP problem with special-ordered sets in the CPLEX optimizer. It can also be solved using commonly used LP software such as LINGO.

In this study, we defined the outputs as sensitivity and specificity, which indicate the overall accuracy of the minority and majority classes, respectively. Thus, the solution of model (11) is a benchmark for the accuracy of minority and majority classes. The experimental design is as follows.

A classic DEA model includes three steps to address application problems. Our experiment adopted these steps to evaluate imbalanced classifiers.

Step 1: Use the visualization t-SNE method of the dataset to depict the two-dimensional plane distribution of the data and divide the data characteristics.

Step 2: Define the DMUs according to the research objectives. Different categories of classification methods were considered as the DMUs in the experiment. Then, we measured the efficiency of each algorithm by applying the DEA-WEI model (5) to datasets with different types of characteristics and identified the change tendencies of efficiency.

Step 3: Determine the performance metrics. Many metrics have been introduced to assess imbalanced classifiers. Some consider the accuracy of both positive and negative classes, such as the AUC or ROC (He and Garcia, 2009; López et al., 2013). Some indices were designed based on the probability distribution of the negative class (Thai-Nghe et al., 2011; Wang and Yao, 2012). In this experiment, we selected several frequently used metrics as outputs of the DEA-WEI model, including sensitivity, specificity, AUC, geometric mean (GM), and F-measure.

Step 4: Analyze the outcomes using the DEA-WEI model. In this experiment, we first used the DEA-WEI model (5) to evaluate the effectiveness of the imbalanced classifiers on different data characteristics. Then, we applied the DEA-WEI benchmarking model (11) to determine the efficiency curve for a given dataset using sensitivity and specificity, and obtained a set of effective algorithms.

The proposed method obtained an optimal solution through linear optimization. The time complexity of the solution method, the interior point method, was approximately $O(n^{3.5}L^{2})$, where *n* denotes the number of variables and *L* is the number of bits in the problem code. The pseudocode for the proposed steps is presented in Appendix D.

4. Empirical study

In this section, the models from Subsection 3.3 are used to evaluate the efficiency of classifiers on the three types of imbalanced data characteristics. The empirical research was designed to answer the following questions: How does the efficiency of an imbalanced learning approach change under different data characteristics; how does the imbalance level affect the efficiency of an imbalanced learning approach; and how to choose an appropriate algorithm for imbalanced data?

4.1 Datasets

The 68 datasets used in the experiment were obtained from the KEEL machine-learning repository. Imbalance ratio (IR) is the number of records of the majority class to the minority class. Multi-class datasets were converted into binary class datasets using certain random classes versus other classes (Thai-Nghe et al., 2011). Records with missing values were excluded from analysis. Table 2 lists the datasets used in these experiments. The IRs of these data ranged from 2.01 to 129.43. We grouped the datasets into overlapping, disjunct, and noisy datasets based on their characteristics. The data type column in Table 2 indicates their characteristics: SD (scarcity or disjunct data), OD (overlapping data), and ND (noisy data).

Table 2 Summary of the datasets.

Datasets	Instances	Features	IR	Data type	Datasets	Instances	Features	IR	Data type
Wine1	178	13	2.01	OD	Ecoli067vs5	220	6	10.00	ND
Echo	131	13	2.04	OD,ND	Vowel0	988	13	10.10	SD
Glass0	214	9	2.06	SD	Glass016vs2	192	9	10.29	SD
Yeast1	1484	8	2.46	SD	Glass2	214	9	10.39	SD
Vehicle1	846	18	2.52	SD	Ecoli0147vs2356	336	7	10.59	ND
Haberman	306	3	2.68	SD	Led7digit02456789vs1	443	7	10.97	SD
Glass_Non-window	214	10	3.18	ND	Glass06vs5	108	9	11.00	OD
Glass0123vs456	214	9	3.19	ND	Ecoli01vs5	240	6	11.00	ND
Vehicle0	846	18	3.23	SD	Glass0146vs2	205	9	11.06	SD,ND
Ecoli1	336	7	3.36	OD	Pageblock 2vs3	358	10	11.34	SD
Hepatitis	155	19	3.84	SD	Ecoli0147vs56	332	6	12.28	ND
New-thyroid2	215	5	4.92	OD	Cleveland0vs4	173	13	12.30	SD,ND
New-thyroid1	215	5	5.14	ND	Ecoli0146vs5	280	6	13.00	SD,ND
Ecoli2	336	7	5.46	ND	Ecoli4	336	7	13.84	OD
Segment0	2308	19	6.01	SD	Yeast1vs7	459	8	13.87	SD
Glass 6	214	9	6.37	SD	Shuttle0vs4	1829	9	13.87	OD
Yeast3	1484	8	8.11	ND	Glass4	214	9	15.47	SD,ND
Ecoli3	336	7	8.19	OD	Abalone9vs18	731	8	16.40	SD
Page-blocks0	5472	10	8.77	SD	Page-blocks 13vs4	472	10	16.44	SD
Ecoli034vs5	200	7	9.00	ND	Zoo_3	101	16	19.20	SD
Yeast2vs4	514	8	9.08	ND	Glass016vs5	184	9	19.44	OD
Ecoli067vs35	222	7	9.09	ND	Shuttle2vs4	129	9	20.50	OD
Ecoli0234vs5	202	7	9.10	ND	Yeast1458vs7	693	8	22.10	SD
Glass015vs2	172	9	9.12	SD	Glass5	214	9	22.81	SD
Yeast0359vs78	506	8	9.12	OD	Yeast2vs8	482	8	23.10	ND
Yeast02579vs368	1004	8	9.14	ND	Yeast4	1484	8	28.41	SD,ND
Yeast0256vs3789	1004	8	9.14	ND	Yeast1289vs7	947	8	30.56	SD,ND
Ecoli046vs5	203	6	9.15	OD,ND	Yeast5	1484	8	32.78	OD
Ecoli01vs235	244	7	9.17	OD,ND	Ecoli0137vs26	281	7	39.15	SD,OD

Ecoli0267vs35	224	7	9.18	ND	Yeast6	1484	8	39.15	,ND SD,ND
Glass04vs5	92	9	9.22	OD	Abalone 17vs7_8_9_10	2338	8	39.31	SD
Ecoli0346vs5	205	7	9.25	ND	Abalone 21vs8	581	8	40.50	SD
Ecoli0347vs56	257	7	9.28	ND	Shutter 2vs5	3316	9	66.67	SD
Yeast05679vs4	528	8	9.35	SD	Abalone19	4174	8	129.43	SD

4.2 Imbalanced classification algorithms

Sampling, cost-sensitive methods, and ensembles are among the state-of-the-art solutions for imbalanced classifications (Chao and Peng, 2018; Song et al., 2018).

The sampling methods in the experiments include the synthetic minority oversampling technique (SMOTE), random under sampling (RUS), SMOTE Wilson's edited nearest neighbor (ENN), and SMOTE Tomek Link. Because the datasets used in the experiments do not provide cost ratios, cost-sensitive algorithms choose the best cost ratio from $\{1,2,10,50,100\}$, which is a common practice used in previous studies (Freund and Schapire, 1996). AdaboostM1, which is an updated method based on AdaBoost and bagging, was selected to represent ensemble classifiers.

The performance of traditional classifiers typically decreases on imbalanced datasets (Song et al., 2018). However, in some studies, classic classifiers, such as SVM, outperformed imbalanced classifiers (Veganzones and Séverin, 2018). Thus, we also included some classic classifiers in the experiments, such as SVM and C4.5 (see Table 3).

The radial basis function (RBF) network and multilayer perceptron (MLP), which are classic neural network methods, were selected as tested algorithms and used to detect whether these two algorithms are effective on different datasets.

Table 3 Classification algorithms used in the experiments.

Methods	Remarks	Abbreviation
Sampling		
Synthetic Minority Oversampling Technique (SMOTE) C4.5	Synthetic Minority Oversampling Technique was used as pre-process methods for over sampling. Base learner is C4.5.	SMC
SMOTE SVM	Synthetic Minority Oversampling Technique was used as pre-process methods for over sampling. Base learner is SVM.	SMS
Radom Under Sampling (RUS) C4.5	Radom Under Sampling method was used as pre-process methods for over sampling. Base learner is C4.5.	RUC
RUS SVM	Radom Under Sampling method was used as pre-process methods for over sampling. Base learner is SVM.	RUS
Hybrid methods		
SMOTE Wilson's edited nearest neighbor (ENN) C4.5	Use SMOTE as oversampling and Wilson's edited nearest neighbor (ENN) as under sapling methods in 1972. Base learner is C 4.5.	SENC
SMOTE ENN SVM	Use SMOTE as oversampling and Wilson's edited nearest neighbor (ENN) as under sapling methods in 1972. Base learner is SVM.	SENS
SMOTE Tomek Link C 4.5	Use SMOTE as oversampling and Tomek Links as under sapling methods in 1976. Base learners is C 4.5.	STLC
SMOTE Tomek Link SVM	Use SMOTE as oversampling and Tomek Links as under sapling methods in 1976. Base learners is SVM.	STLS
Cost-sensitive classifiers		
CS-SVM	Assign different costs for instance to obtain minimum total costs.	CSVM
CS-C4.5	The split in nodes is assigned as cost to the minority class	CSC4
Ensemble		
AdaboostM1 C4.5	Use C4.5 as base classifier and Freund and Schapire method in	AdBC

AdaboostM1 SVM	1996. Use SVM as base classifier and Freund and Schapire method in 1996.	AdBS
Bagging C4.5	Base learner was C4.5 and Breiman method was used.	BaC
Bagging SVM	Base learner was SVM and Breiman method was used.	BaS
Classic Classifiers		
C4.5	Quilan's C4.5 algorithm was used in experiments. Set pruning set and 2 instances in minimum leaf.	C4.5
Support vector machine	A radial basis function and polynomial function as the kernel parameters were used. Chose better results between two kernel functions.	SVM
Compared classifiers		
RBF network	Use *k*-means clustering algorithm to provide basis function.	RBF
Multilayer Perceptron	Use backpropagation to classify instances.	MLP

C4.5, SVM, CS-SVM, RBF network, and MLP were implemented using the Weka algorithm settings. A stratified 10-fold cross-validation procedure was applied. First, we randomly divided each dataset into 10 parts, randomly selected nine of them to train a classifier, and used the remaining part as the test set. After ten training and testing cycles, the classification results of the ten test sets were summarized in a confusion matrix, and the classification accuracy indicators were calculated based on the matrix (see Tables 5 and 6).

4.3 Metrics and model settings

The following performance metrics were used in the experiments:

True positive rate (TPR): proportion of real positive data correctly classified as positive;

True negative rate (TNR): proportion of real negative data correctly classified as negative;

AUC: area under the ROC curve;

GM: geometric mean of the true positive rate and true negative rate;

F-measure: harmonic average of the precision and recall;

The indicators of j^{th} DMU are $Y_j = (y_{1j}, y_{2j}, \dots, y_{5j})$, and the outputs are as follows:

y_{1j}: true positive rate;

y_{2j}: true negative rate;

y_{3j}: AUC;

y_{4j}: GM;

y_{5j}: F-measure.

Then, these indices were standardized using the following formulas for models (5) and (6):

$$y_{rj} = y_{rj} \Big/ \underset{j}{Max}\{y_{rj}\}$$

The DEA-WEI models developed in Section 3 were used to empirically investigate the efficiency of different algorithms. Table 4 lists the settings of the DEA-WEI models.

Table 4 Settings of the DEA models

	DMUs	Outputs	Targets
Model (5)	Classic, Cost Sensitive, Ensemble, Pre-process, Hybrid.	TPR,TNR,ACC,AUC,GM, F	DEA efficiency
Model (6)		TPR, TNR	Overall performance scores
Model (11)	RBF, MLP		DEA Benchmark; Efficiency frontier.

The three models have different targets and objectives. The objective function of Model (5) is the efficiency of the DMUs. Model (6) is the dual model of Model (5), and its objective function is overall performance of the evaluation indicators. The objective function of Model (11) is the distance from each DMU to the efficiency curve.

We divided the DMUs into two groups based on the base learner (C4.5 or SVM):

Classic (DMU$_{1c}$): C4.5;

Cost Sensitive (DMU$_{2c}$): CSC4;

Pre-process (DMU$_{3c}$) over sampling: SMC;

(DMU$_{4c}$) under sampling : RUC;

Hybrid (DMU$_{5c}$): SENC;

(DMU$_{6c}$): STLC;

Ensemble (DMU$_{7c}$): AdBC;

(DMU$_{8c}$): BaC.

and

Classic (DMU$_{1s}$): SVM;

Cost Sensitive (DMU$_{2s}$): CSVM;

Pre-process (DMU$_{3s}$) over sampling: SMS;

(DMU$_{4s}$) under sampling, RUS.

Hybrid (DMU$_{5s}$): SENS;

(DMU$_{6s}$): STLS;

Ensemble (DMU$_{7s}$): AdBS;

(DMU$_{8s}$): BaS.

Tables 5 and 6 summarize the average performance of each class of imbalanced learning approaches on three data characteristics (disjunct, overlapping, and noisy).

Table 5 Average performance of imbalanced leaning approaches (base learner C4.5)

		DMU$_{1c}$	DMU$_{2c}$	DMU$_{3c}$	DMU$_{4c}$	DMU$_{5c}$	DMU$_{6c}$	DMU$_{7c}$	DMU$_{8c}$
Disjunct	TPR	90.8625	90.3563	90.7625	83.6625	91.3313	89.7250	90.8188	***91.9375***
	TNR	41.4750	49.5688	58.2375	59.4375	***63.5125***	57.8813	44.8000	39.8875
	AUC	66.3219	69.9625	74.5000	71.5500	***77.4219***	73.8031	67.8094	65.9125
	GM	54.1525	60.4228	70.1363	63.0966	***72.7767***	66.1676	56.5914	51.8429
	F	85.3812	85.5239	86.0977	82.4175	***86.6243***	85.4198	85.6393	85.9921
Overlapping	TPR	97.2667	97.7333	95.4667	97.9667	97.0667	95.9333	98.1000	***98.3667***
	TNR	63.9667	55.9000	76.8000	48.0667	64.3333	***80.1333***	63.5000	58.6333
	AUC	80.6333	76.8167	86.1333	73.0167	80.7000	***88.0333***	80.8000	78.5000
	GM	75.8311	70.2471	84.8721	64.6036	77.9732	***87.1796***	76.2561	71.4014
	F	88.0724	88.3879	87.7500	88.1561	88.1305	88.0164	***88.6463***	88.6131
Noisy	TPR	97.9000	98.3000	96.4556	94.2889	94.9667	96.8667	***98.8889***	98.6889
	TNR	57.9667	43.6778	68.3222	43.0111	70.7667	***69.5000***	41.7444	41.8778
	AUC	77.9389	70.9889	82.3889	68.6500	82.8667	***83.1833***	70.3167	70.2833
	GM	74.0277	55.9744	80.7230	49.5092	81.4624	***81.5781***	51.4863	59.9535
	F	87.8486	87.6246	87.2950	85.5773	86.5388	87.5592	***87.8775***	87.6314

Table 6 Average performance of imbalanced leaning approaches (base learner SVM)

		DMU$_{1s}$	DMU$_{2s}$	DMU$_{3s}$	DMU$_{4s}$	DMU$_{5s}$	DMU$_{6s}$	DMU$_{7s}$	DMU$_{8s}$
Disjunct	TPR	92.0375	67.6563	90.4125	78.9063	89.8125	88.7688	90.6125	***92.0563***
	TNR	30.2563	***76.9563***	41.3313	61.9500	48.9375	51.0313	48.5688	38.9938
	AUC	61.1438	***72.3219***	65.8719	70.4281	69.3750	69.9000	69.5906	65.5250
	GM	38.0092	65.6047	54.8291	***68.6206***	60.9904	61.7542	61.3811	45.9303
	F	86.0133	72.3299	85.5310	78.8152	85.4960	84.8388	85.6000	***86.4766***
Overlapping	TPR	97.9667	79.6333	97.7000	96.5333	***99.5000***	98.3000	97.6000	98.5000
	TNR	37.8333	***91.1667***	57.5333	68.9000	47.2000	59.4333	45.2333	42.4333
	AUC	67.8833	***85.4000***	77.6167	82.7167	73.3500	78.8667	71.4167	70.4667
	GM	46.5799	***84.4349***	71.9577	78.0323	63.5298	73.4664	63.0000	61.8959
	F	87.8311	77.3144	88.1811	87.9411	***89.1333***	88.5113	87.5438	87.5761
Noisy	TPR	99.5667	73.9000	98.2000	82.1333	95.0222	94.2000	98.5556	***99.6333***
	TNR	41.8333	***92.0556***	47.3889	75.0889	51.4667	53.6000	43.6222	34.3000
	AUC	70.7056	***82.9889***	72.7944	78.6111	73.2444	73.9000	71.0889	66.9667
	GM	52.3905	***82.0806***	57.7393	77.2611	65.1268	65.6091	59.1360	41.9677
	F	***88.1883***	75.0537	87.7555	78.9454	86.2086	85.8511	87.7224	88.0212

Tables 5 and 6 imply that the hybrid methods performed better with C4.5 as the base classifier, whereas the cost-sensitive methods performed better when SVM was the base learner. The bagging SVM had the best classification accuracy for minority data (TPR).

To test for statistical differences in the mean performance of the algorithms for different data characteristics, we performed a one-way analysis of variance (ANOVA) test at a confidence level of 0.1. Each pair of compared values passed the homogeneity of variance test. For the two base classifiers, we tested the statistical significance of the difference in the average performance of the algorithms for the three data characteristics under different evaluation metrics. The results are shown in Figs. 7 and 8. The mean performances on TPR and F-value were significantly different at the 0.001 confidence level when C4.5 was the base learner. The mean performances for GM and AUC were significantly different at 0.1 and 0.05 confidence levels, respectively.

Table 7 One-way ANOVA test for average performance of imbalanced leaning approaches (base learner C4.5)

	Mean Square	F-value	p-value
TPR	2220.7208	26.3201	0.0000***
TNR	319.8098	2.5704	0.1003
AUC	186.9972	6.8898	0.0050**
GM	412.4132	3.9983	0.0338*
F	16.5858	21.2338	0.0000***

Remark: ***p<= 0.001，** p<= 0.05，*p<= 0.1.

When SVM was the base classifier, the mean TPR and AUC were significant at a 0.1 confidence level. This indicates that the performance of the SVM on the three types of data characteristics may be more average and closer.

Table 8 One-way ANOVA test for average performance of imbalanced leaning approaches (base learner SVM)

	Mean Square	F-value	p-value
TPR	185.2685	2.6733	0.0924*
TNR	93.5332	0.3187	0.7306
AUC	134.8442	5.3266	0.0135*
GM	230.0003	1.6729	0.2118
F	26.2951	1.2261	0.3136

Remark: ***p<= 0.001， ** p<= 0.05， *p<= 0.1.

4.4 Main results and analysis

This subsection answers the following three questions:

Q1: How does the efficiency of imbalanced learning approach change under different data characteristics?

Q2: How does the imbalanced level affect the efficiency of classifiers?

Q3: Given imbalanced data, how can a classification algorithm achieve a satisfactory efficiency?

4.4.1 Q1: How does the efficiency of imbalanced learning approaches change under different data characteristics?

Based on the 68 KEEL datasets, we computed the efficiency of each group of imbalanced learning algorithms for the three types of data characteristics using Model (5). Table 7 summarizes the results and ranks the DMUs based on their efficiencies. For base learner C4.5, DMU_1 (C4.5) and DMU_8 (ensemble bagging) were ranked first in terms of efficiency on the disjunct data, and DMU_4 (under sampling) performed the best on the overlapping dataset. DMU_7 (ensemble AdaboostM1) and DMU_8 (ensemble bagging) achieved the best efficiencies for noisy data.

Table 9 Efficiency of imbalanced learning algorithms on different data characteristics (base learner C4.5)

	Disjunct	Overlapping	Noisy
	Model (5)		
DMU_1(classic)	1	0.886	0.892
DMU_2(cost sensitive)	0.941	0.935	0.969
DMU_3(over sampling)	0.922	0.825	0.826
DMU_4(under sampling)	0.853	1	0.973
DMU_5(hybrid)	0.878	0.902	0.805
DMU_6(hybrid)	0.886	0.809	0.820
DMU_7(ensemble AdBC)	0.971	0.893	1
DMU_8(ensemble BaC)	1	0.906	1

Similar to the results in Table 9, DMU_7 and DMU_8 performed the best on noisy data, and DMU_1 was ranked first on the disjunct data type when the base learner was SVM (Table 10). The differences were that DMU_8 (ensemble bagging) and DMU_1 tied first on the overlapping data, and DMU_3 (oversampling) achieved the best performance on the disjunct data in Table 10.

Table 10 Efficiency of imbalanced learning algorithms on different data characteristics (Base learner SVM)

	Disjunct	Overlapping	Noisy
	Model (5)		
DMU$_1$(classic)	1	1	0.961
DMU$_2$(cost sensitive)	0.643	0.635	0.658
DMU$_3$(over sampling)	1	0.860	0.913
DMU$_4$(under sampling)	0.835	0.776	0.759
DMU$_5$(hybrid)	0.941	0.931	0.933
DMU$_6$(hybrid)	0.912	0.847	0.903
DMU$_7$(ensemble AdBC)	0.953	0.961	1
DMU$_8$(ensemble BaC)	0.915	1	1

We used the Holm post-hoc test to investigate whether the differences in the efficiency of the imbalanced learning algorithms presented in Tables 9 and 10 were significant. The null hypothesis was H$_0$: the means of the efficiency differences of the imbalanced learning algorithms for each data type were the same. Tables 11 and 12 summarize the test results.

Table 11 Holm test for the efficiency difference on different datasets (C4.5 as the base classifier)

Comparison	Statistic	Adjusted p-value	Result
Disjunct vs All datasets	2.42061	0.01549**	H$_0$ is rejected
Overlapping vs All datasets	3.77616	0.00048***	H$_0$ is rejected
Noisy vs All datasets	2.71109	0.01341**	H$_0$ is rejected

Remark: ***p<= 0.001，** p<= 0.05，*p<= 0.1.

Table 12 Holm test for the efficiency difference on different datasets (SVM as the base classifier)

Comparison	Statistic	Adjusted p-value	Result
Disjunct vs All datasets	2.32379	0.02014**	H$_0$ is rejected
Overlapping vs All datasets	3.38886	0.00211**	H$_0$ is rejected
Noisy vs All datasets	2.80791	0.00997**	H$_0$ is rejected

Remark: ***p<= 0.001，** p<= 0.05，*p<= 0.1.

The Holm post-hoc test verified that the difference in efficiency of the imbalanced learning algorithms on different types of datasets was statistically significant.

4.4.2 Q2: How does the imbalanced level affect the efficiency of classifiers?

To answer Q2, we divided the datasets into three classes based on their imbalance ratios: $IR \propto [30,+\infty]$, $IR \propto [10,30)$, and $IR \propto [1,10)$. Table 13 lists the efficiency of each class of algorithms with various imbalance ratios. The indicator values were computed as the average of the two base classifiers (C4.5 and SVM). The algorithms exhibited different levels of robustness for different imbalance ratios. The most affected class of algorithms by IR was DMU$_2$ (cost sensitive), whose efficiency decreased from 0.9428 to 0.7412 when the imbalance ratio increased from less than 10 to over 30. This indicates that cost-sensitive classifiers are unsuitable for highly imbalanced data. The efficiency of DMU$_3$

(oversampling) also decreased as the IR increased. However, the efficiency of DMU_5 (hybrid methods) decreased and then increased, and the efficiency of DMU_4 (undersampling) increased as the IR increased. The efficiency of DMU_6 (ensemble) did not change when IR changed.

Table 13 Efficiency tendencies with different imbalance levels

	IR≥30		10≤IR<30		IR<10	
	Model (5)	Ranks	Model (5)	Ranks	Model (5)	Ranks
DMU_1	0.9915	A	1	A	0.9974	A
DMU_2	0.7412	C	0.7752	C	0.9428	B
DMU_3	0.9549	B	0.9779	B	0.9988	A
DMU_4	1	A	1	A	0.9945	A
DMU_5	0.9971	A	0.9823	B	0.9927	A
DMU_6	1	A	1	A	1	A

4.4.3 Q3: Given an imbalanced data, how to judge whether a classifier has achieved a satisfactory efficiency?

This question answers whether a new algorithm has an acceptable efficiency compared to existing methods. This is the main question that should be answered in the development of new algorithms for imbalanced classification. Imbalanced classification algorithms should improve the accuracy of the minority class and ensure that the accuracy of the majority class is within an acceptable range. Benchmark algorithms are needed as the basis for comparison.

In this study, the following approach is proposed to determine the benchmark algorithms. First, we establish a DEA-WEI benchmark model based on the given indices, such as TPR and TNR. The target value of the evaluation index is calculated using model (11), which is designed for this purpose. If the target value is the same as the actual value, the algorithm is based on the efficiency boundary curve. Thus, we use the algorithm for the boundary curve as the benchmark algorithm.

There are two cases in which a DMU is inefficient. A DMU is located inside the efficiency boundary, which means that its efficiency is lower than that of the benchmark algorithm. However, a DMU outside the efficiency boundary outperforms the benchmark algorithm on bases of input consumption. Thus, it has better performance, as measured by the selected indices, but its efficiency is lower than that of the benchmark algorithms. Algorithms outside the efficiency boundary are feasible in practical applications. In summary, effective algorithms are those whose efficiency is at or outside the equilibrium state of different outputs.

Table 14 DEA benchmark for Wine1

Wine 1	Benchmark	Output1	Output2	At or outside of boundary	Is it effective
		TPR	TNR		
RBF	Actual	73.9	49.2	×	×
	Targets	97.7	50.7		
MLP	Actual	96.6	99.0	√	×
	Targets	94.9	97.3		
OSS+SVM	Actual	86.8	78.0	×	×
	Targets	96.0	78.6		

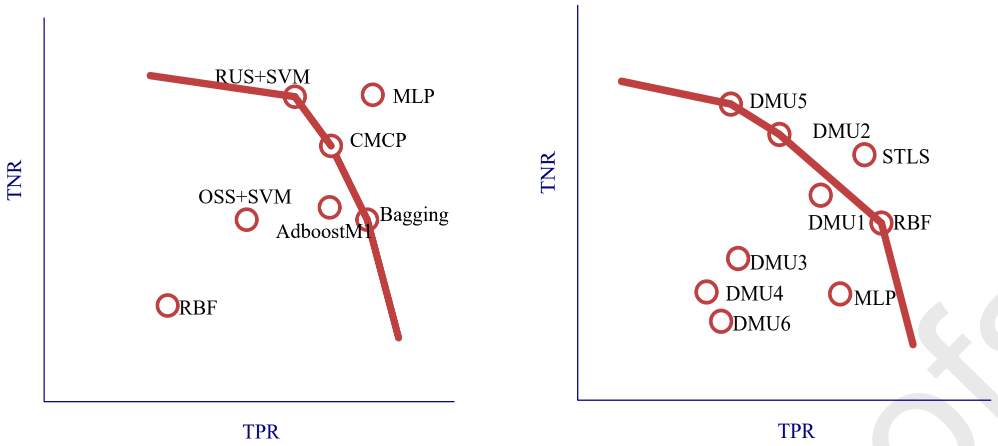
RUS+SVM	Actual	93.2	98.3	√	√
	Targets	93.2	98.3		
AdboostM1	Actual	94.1	86.4	×	×
	Targets	95.5	86.5		
CS-MCQP	Actual	94.9	96.6	√	√
	Targets	94.9	96.6		
Bagging	Actual	96.0	78.6	√ be	√
	Targets	96.0	78.6		

Table 14 uses dataset Wine 1 as an example to show the DEA benchmark Model (11). The TPR and TNR were used as the outputs of this model because they are the most important performance metrics in imbalance classification. The efficiency boundary (Fig. 1 (a)) consisted of RUS+SVM, CMCP, and bagging. OSS+SVM, RBF, and AdbostM1 were inefficient because they were located inside the efficiency boundary. RUS+SVM, CMCP, and bagging were efficient, as they were at the boundary. Given a new algorithm and a given dataset, we can compare an algorithm with a benchmark to determine its efficiency.

Table 15 DEA benchmark for overlapping data

Overlapping	Benchmark	Output1	Output2	At or outside of boundary	Is it effective
		TPR	TNR		
RBF	Actual	96.9	66.6	√	√
	Targets	96.9	66.6		
MLP	Actual	90.8	37.5	×	×
	Targets	94.3	69.7		
Smote+Tomek link	Actual	97.1	69.8	√	×
	Targets	95.4	68.4		

Table 15 shows the evaluation of the MLP, RBF, and STLS for overlapping data using TPR and TNR as outputs. From Fig. 1(b), we can observe that RBF is effective, whereas MLP is not, compared with the three benchmarks (DMU$_5$, DMU$_2$, and RBF). STLS is outside the boundary, which means that it is inefficient because it should focus more resources to obtain higher outputs.



(a)Wine 1 (b) Overlapping

Figure 1 DEA efficiency boundary

In summary, the effectiveness of a new algorithm according to its position on the efficiency boundary, which was constructed using a DEA-WEI model, was evaluated.

4.5 Efficiency performance on artificial datasets

Many factors can affect the accuracy of classification, such as dimensionality of the dataset, noise intensity, and extreme outliers. However, these data are not readily available as real data. Therefore, in this section, we use several artificial datasets to test the possible changes in the efficiency of classification algorithms in different environments.

4.5.1 Dimensionality of data

To investigate the influence of data dimensionality, we used the Chinese text corpus from the Chinese Academy of Sciences (http://mtgroup.ict.ac.cn/new/resource/index.php). The corpus was collected from more than 20,000 news-webpages and contained approximately 150,000 Chinese words. The words in each webpage were extracted to form a file. They were manually divided into ten categories: environment, computer, transportation, education, economy, military, sports, medicine, art, and politics. The training and test sets included 1,884 and 934 files, respectively.

To investigate the influence of the dimensionality of data, we chose the most commonly used text classifier, k-nearest neighbor (KNN), as the base classifier, and applied it to seven dimensions: 100,000, 30,000, 10,000, 3,000, 1,000, 500, and 200. Dimension refers to the number of features, and the features are Chinese words in the corpus.

The feature selection methods used in the experiment included category information (CI), information gain (IG), expected cross entropy (CE), mutual information (MI), weight of evidence (WE), and χ^2 statistics.

The accuracy of each method for different numbers of features is presented in Table 16.

Table 16 Accuracies of KNN under different dimensionalities on test sets

Number of features	CI	IG	CE	WE	χ^2	MI

100000	71.49	71.37	71.51	71.45	69.91	71.52
30000	72.95	71.15	71.63	73.02	69.16	**74.44**
10000	78.75	79.17	79.16	76.12	74.45	71.57
3000	88.63	87.76	87.25	85.01	84.58	59.31
1000	**89.40**	**88.89**	**88.94**	**89.07**	**87.04**	41.09
500	87.14	86.19	86.55	84.04	83.12	41.58
200	83.31	82.98	82.36	81.37	81.15	33.67

Table 17 DEA efficiency of KNN under different dimensionalities on test sets

Number of features	DEA efficiency
100000	0.938
30000	1.000
10000	1.000
3000	1.000
1000	1.000
500	0.877
200	0.671

When the data dimension was 1,000, KNN achieved the highest accuracy. When the dimensionality of the dataset was increased to 3,000 and above, the accuracy decreased. However, the efficiency of KNN decreased dramatically as the dimensionality decreased from 500 to 200 (Table 17).

4.5.2 Influence of noise intensity

Existing studies (López et al., 2013; 2014) have proven that the influence of noise intensity on classifiers can be overcame by data preprocessing methods, such as SMOTE+ENN and SPIDER2. We analyzed the efficiency of several data preprocessing methods on the Subclus dataset from KEEL, in which classes were generated by introducing 20% Gaussian noise. The base classifier was C4.5, and the evaluation indices were TPR, TNR, and AUC.

Table 18 Performance of data preprocessing methods under noise data (base classifier is C4.5)

	None				20% Gaussian noise			
	TPR	TNR	AUC	GM	TPR	TNR	AUC	GM
None	100	90.29	95.14	95.02	0	100	50.00	0
RUS	100	78.00	89.0	88.32	97.00	74.00	85.50	84.72
SMOTE	96.14	95.29	95.71	95.71	89.14	88.00	88.57	88.57
SMOTE+ENN	96.76	96.23	96.49	96.49	96.25	95.73	95.99	95.99
SPIDER2	100	100	100	100	94.80	90.33	92.56	92.54

Table 19 DEA efficiency of classifiers under noisy data (base classifier is C4.5)

The Methods	DEA efficiency	
	None	20% Gaussian noise
None	0.929	0.433
RUS	1.000	1.000
SMOTE	0.887	0.879
SMOTE+ENN	0.886	0.876
SPIDER2	0.883	0.895

The results showed that the efficiency of C4.5 was considerably affected by noise. Without data preprocessing (Table 18), the classifier could not identify positive data when there was 20% noise. Similarly, the efficiency of the classifiers (Table 19) was significantly affected by noise. However, data preprocessing methods can significantly reduce the influence of noise on the performance and efficiency of classifiers.

4.5.3 Influence of extreme outliers

We used an artificial dataset from López et al. (2013) (Fig. 13 in López et al., 2013), which has 1,000 examples with an IR of 90% (i.e., one positive instance per 10 instances), to study the influence of extreme outliers on the performance and efficiency of classifiers. C4.5 was the base classifier. Areas covered by outliers showed the percentage of data scattered with outliers. In this case, 0% indicates a complete separation of the minority and majority data, whereas 100% indicates that the two classes are completely interleaved.

The performance of the classification is listed in Table 20.

Table 20 Performance of C4.5 on dataset with different distributions of outliers

Area covered by outlier	TPR	TNR	AUC	GM
0	100	100	100	100
20%	79.00	100	89.50	88.88
40%	49.00	100	74.50	70.00
50%	47.00	100	73.50	68.56
60%	42.00	100	71.00	64.81
80%	21.00	99.89	60.44	45.80
100%	0	100	50.00	0.00

As the proportion of outliers (positive instances) gradually increased, the performance and efficiency of the classifiers decreased. (Table 21).

Table 21 DEA efficiency of classifiers under different distribution area of outliers

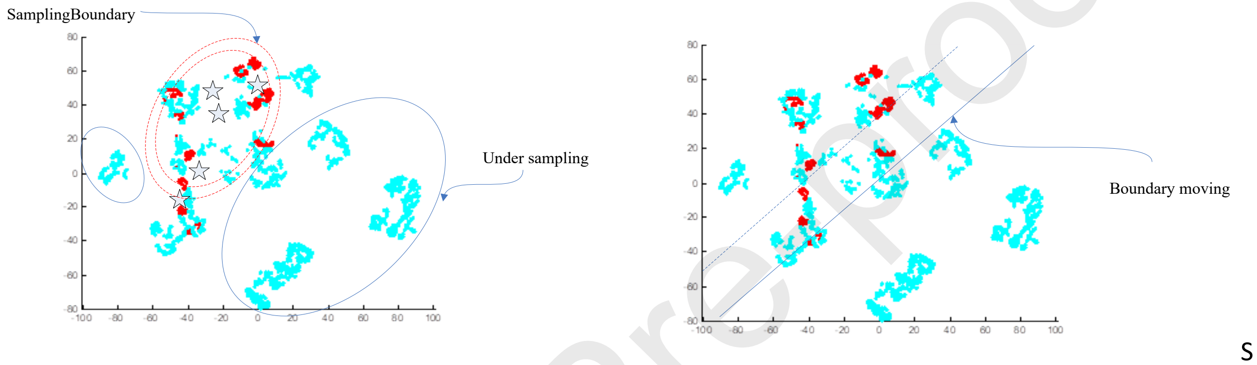
Area covered by outlier	DEA efficiency
0	1.000
20%	0.895
40%	0.745
50%	0.735
60%	0.710
80%	0.605
100%	0.500

5. Discussions

In imbalanced classification, the improvement in the accuracy of the minority class often occurs at the expense of the accuracy of the majority class. The efficiency boundary of DEA attempts to determine a trade-off between mutually restrictive metrics. This section analyzes the performance of each group of imbalanced learning methods under the three types of data characteristics, and proposes a three-step process to select appropriate classifiers for a dataset with certain characteristics and IRs. Data were visualized using t-SNE (Maaten, and Hinton, 2008).

5.1 Efficiency of imbalanced approaches on the three types of data characteristics

Disjunct and Scarcity. In the experiment, the performance of all approaches on disjunct datasets was better than that of the other two types of data characteristics, except for the ensemble methods (Table 8). Because the oversampling approaches (stars in Fig. 2(a)) cannot change the boundary range (separate hyperplane) of the minority class (red circles in Fig. 2(a)), they cannot significantly improve the classification results and consequently improve the efficiency. The undersampling approaches can improve the accuracy of the minority class, but they simultaneously decrease the accuracy of the majority class (Fig 2(a)). Thus, they are not efficient for this type of data. The cost-sensitive group, whose principle is to push the boundary to the minority class, can improve the accuracy of the minority class, but it also leads to a rapid decrease in the accuracy of the majority class (Fig. 2(b)).



(a) Sampling (b) Cost sensitivity

Figure 2 Segment0

Overlapping. Compared to disjunct data, overlapping is easier to be classified because the two classes are located in a concentrated area (Fig. 3). The classic classifiers such as SVM have a higher efficiency for this type of data (Table 8). The ensemble methods have high efficiency on the three ranges of imbalance ratios. The cost sensitive classifiers are not effective for overlapping data.

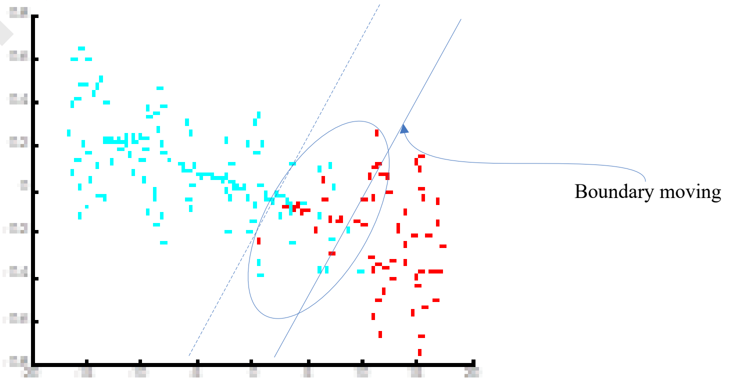


Figure 3 Wine1

Noisy. Classifying noisy data is a demanding task owing to the difficulty to determine the boundary between the two classes (Fig. 4). As oversampling approaches add minority data to a training set (stars in Fig. 4), the boundary moves closer to the majority class, resulting in the increase in the accuracy of the minority class and decrease in the accuracy of the majority class. Thus, they are not effective. Traditional classifiers cannot largely improve the accuracy when the noisy data exist in the majority class, thus their efficiency is not satisfactory either.

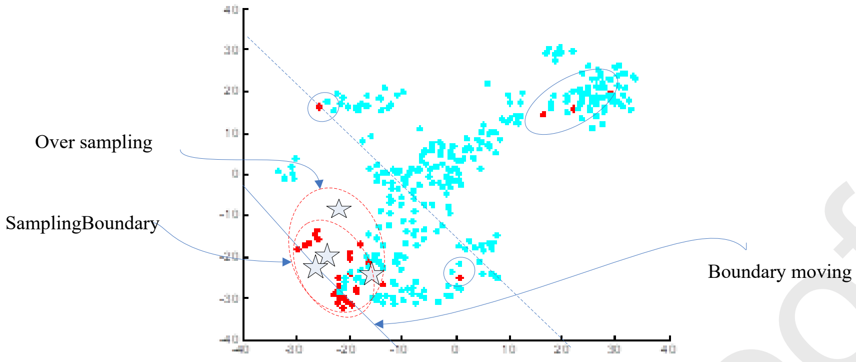
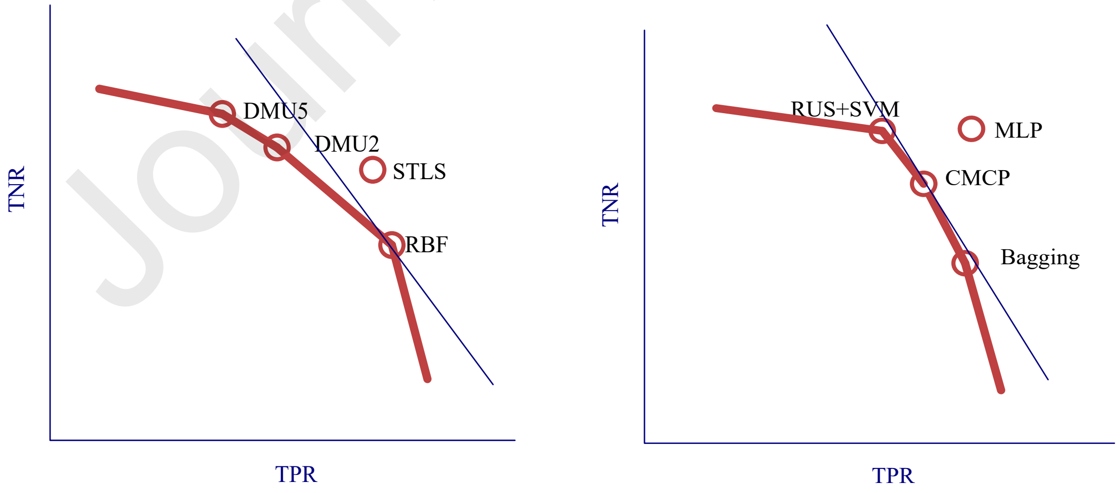


Figure 4 Ecoli0147vs2356

The process of testing the effectiveness of a new algorithm starts with selecting some traditional classic algorithms and building an efficiency curve. The effectiveness of the new algorithm is determined based on the distance between the curve and new algorithm. A tested algorithm is used to show the process. Figure 5 shows the efficiency curve and classifiers located at and outside the tangent of the efficiency curve. STLS approach has higher accuracy but its efficiency is not the best, because it is located outside the efficiency boundary curve, that is, “attainable set” proposed in Liu et al. (2011) in the DEA. Basically, it is unnecessary to use more complex technologies when simple methods can solve the same problem. However, it is a feasible solution for classifiers to be located outside the tangent of the efficiency curve (Fig. 5(a)) when the accuracy is used as the only metric. For example, the MLP and STLS in Fig. 5(b).



(a) RBF and STLS as tested algorithms (b) MLP as tested algorithm.

Figure 5 Algorithms outside the efficiency curves

5.2 Imbalanced classifier selection

Based on the above analysis, we propose a three-step process for selecting the appropriate imbalanced classifiers:

Step 1: For a given imbalanced dataset that should be classified, the data characteristics are determined using data visualization technology (such as t-SNE).

Step 2: For a dataset with an IR⩾30, if the data characteristics are disjunct, it is recommended to use standard classifiers or ensemble methods. For overlapping data, it is recommended to use undersampling methods such as RUS based on C4.5. The use of ensemble methods is recommended for noisy data.

For a dataset with an IR<10, if the dataset is disjunct, it is recommended to use oversampling, such as SMOTE based on SVM. If a dataset overlaps, standard classifiers or ensemble methods are suitable. If a dataset is noisy, ensemble methods are a reasonable choice.

For a dataset with 10⩽IR<30, classic algorithms can be used for imbalanced data with disjunct characteristics, and ensemble methods are suitable for imbalanced data with noisy or overlapping characteristics.

Step 3: If a new algorithm should be developed for imbalanced datasets, its effectiveness can be verified using benchmark classification methods. First, traditional imbalanced learning methods that should be compared are selected, the efficiency curve is constructed, and the new algorithm is compared with the curve to determine its efficiency.

Table 22 summarizes the recommendations.

Table 22 Recommendations for imbalanced classification methods

Data	Imbalance ratio	Data intrinsic characteristic	Recommended methods
1	IR>30	Disjunct	Standard classifiers or ensemble methods
2		Overlapping	Under sampling methods
3		Noisy	Ensemble methods
4	10<IR<30	Disjunct	Standard classifiers
5		Overlapping	Ensemble methods
6		Noisy	Ensemble methods
7	IR<10	Disjunct	Over sampling
8		Overlapping	Standard classifiers or ensemble methods
9		Noisy	Ensemble methods

Remark: The criterion for the choice of the classifier may be related to the research object. For example, data preprocessing and cost-sensitive methods are potential options for imbalanced classification tasks. Other criteria may include economic cost, interpretability, and production requirements. For example, in NASA software defects, different cost-sensitive algorithms are feasible alternative algorithms owing to their economic costs (Siers and Islam, 2018).

5.3 Comparisons

Garcia et al. (2009) proposed the IBA method, which also considers the accuracy of the majority and minority data. The IBA is calculated as follows:

$$IBA = (1 + TP_{rate} - TN_{rate}) \cdot Gmean^2$$

The results of the two methods, presented in Tables 23 and 24, are significantly different. The bestperforming algorithm obtained using IBA was not the most effective algorithm based on our method. Similarly, the most efficient algorithm did not have the highest IBA score. For example, in a noisy dataset, the ensemble method was the most efficient based on our approach, whereas the hybrid sampling achieved the highest IBA score. Although these two methods are based on the idea of reconciling the classification accuracy of the two types of data, IBA modifies the average accuracy, whereas our approach establishes an efficiency curve and evaluates the advantages and disadvantages of an algorithm. We pursued relative efficiency rather than maximum average accuracy.

Table 23 Comparison of the proposed method and existing IBA method (C4.5)

	Disjunct		Overlapping		Noisy	
	IBA	Our Method	IBA	Our Method	IBA	Our Method
DMU_1(classic)	0.438	1	0.767	0.886	0.767	0.892
DMU_2(cost sensitive)	0.514	0.941	0.700	0.935	0.484	0.969
DMU_3(over sampling)	0.652	0.922	0.855	0.825	0.835	0.826
DMU_4(under sampling)	0.495	0.853	0.626	1	0.371	0.973
DMU_5(hybrid)	0.677	0.878	0.807	0.902	0.824	0.805
DMU_6(hybrid)	0.577	0.886	0.880	0.809	0.848	0.820
DMU_7(ensemble)	0.468	0.971	0.783	0.893	0.417	1
DMU_8(ensemble)	0.409	1	0.712	0.906	0.564	1

Table 24 Comparison of the proposed method and existing IBA method (SVM)

	Disjunct		Overlapping		Noisy	
	IBA	Our Method	IBA	Our Method	IBA	Our Method
DMU_1(classic)	0.234	1	0.347	1	0.433	0.961
DMU_2(cost sensitive)	0.390	0.643	0.631	0.635	0.551	0.658
DMU_3(over sampling)	0.448	1	0.726	0.860	0.503	0.913
DMU_4(under sampling)	0.551	0.835	0.777	0.776	0.639	0.759
DMU_5(hybrid)	0.524	0.941	0.615	0.931	0.609	0.933
DMU_6(hybrid)	0.525	0.912	0.750	0.847	0.605	0.903
DMU_7(ensemble)	0.535	0.953	0.605	0.961	0.542	1
DMU_8(ensemble)	0.323	0.915	0.598	1	0.291	1

The proposed model provides a comprehensive evaluation of various performance metrics (such as accuracy, AUC, and F-measure). Compared to traditional metrics, DEA does not pursue the highest accuracy rate of either class, such as the highest TP or TN rates, or aggregated performance metrics (e.g., AUC); rather, it aims to obtain a trade-off between evaluation indicators by constructing an efficiency curve, which is composed of points with relative ratios of benefit and cost.

Remark: The evaluation ideas for the classifier can be categorized into selecting the average performance of multiple targets, focusing on a single indicator such as the accuracy of the minority class, and balancing the two types of data classification accuracy. The results obtained using different methods may be different. Generally, the evaluation target of the selection algorithm should be determined based on the characteristics of the theoretical research object. An efficiency curve, which is a relatively effective combination of the TP and TN rates, is an intuitive interpretation of classifier efficiency. As shown in Figure 1, the efficiency of any classifier can be observed intuitively by comparing its position to the efficiency boundary.

6. Conclusion

Imbalanced classification is an important research area because imbalanced data are ubiquitous. Determining a balance between the accuracy of minority and majority data is a challenging task in imbalanced classification. Furthermore, the characteristics effects of imbalanced data on learning algorithms is an important but understudied problem. Inspired by the concept of efficiency in decision making, the use of an *efficiency* curve, which is established using DEA-WEI, is proposed in this study to evaluate imbalanced algorithms by identifying a trade-off between the benefits of improved accuracy of the minority class and the costs of reduced accuracy of the majority class.

The experiments were based on the average performance of 68 KEEL datasets and three artificial datasets. The results showed that: 1) the efficiency of the algorithms was affected by the characteristics of the imbalanced data. DMU_7 (ensemble AdaboostM1) and DMU_8 (ensemble bagging) achieved the best efficiency on noisy data, regardless of the base learners (C4.5 and SVM). DMU_1 (C4.5 or SVM) was the most efficient for disjunct data. Further, base learners had some effect on the efficiency of the imbalanced algorithms; 2) the algorithms exhibited different robustness for different imbalance ratios. The cost-sensitive algorithms were the most affected, and their efficiency decreased from 0.9428 to 0.7412 when the imbalance ratio increased from less than 10 to over 30. Furthermore, we described a method to evaluate the efficiency of imbalanced algorithms for a given imbalanced data by comparing its position to the efficiency boundary, which was constructed using a DEA-WEI model.

This study had two limitations. First, the results were based on open and artificial datasets. Validation using large-scale real-life imbalanced data is a future research direction. Second, determining the degree to which the accuracy of a classifier on a given dataset can be improved is important. We will further investigate this problem. Furthermore, a stochastic DEA incorporating a statistical model and sampling process is a promising direction for measuring the efficiency change of classifiers.

Acknowledgments:

This research was supported in part by grants from the National Natural Science Foundation of China (#71874023, #71725001, #71771037, #71971042).

References:

[1] Barella, V. H., Garcia, L. P., de Souto, M. C., Lorena, A. C., & de Carvalho, A. C. (2021). Assessing the data complexity of imbalanced datasets. Information Sciences, 553, 83-109.

[2] Brzezinsky D, (2018) Visual-based analysis of classification measures and their properties for class imbalanced problems. Information Sciences 462, 242-261.

[3] Chao, X., & Peng, Y. (2018). A cost-sensitive multi-criteria quadratic programming model for imbalanced data. Journal of the Operational Research Society, 69(4), 500-516.

[4] Chao X., Kou G., Peng Y., Herrera-viedma E. & Herrera F., (2021) An efficient consensus reaching framework for large-scale social network group decision making and its application in urban resettlement, Information Sciences 575, 499-527.

[5] Charnes A, Cooper WW, Rhodes E. (1978) Measuring the efficiency of decision making units. European Journal of Operational Research 2:429–44.

[6] Chen, X., Gong, C., & Yang, J. (2021). Cost-sensitive positive and unlabeled learning. Information Sciences, 558, 229-245.

[7] Chouhan, S. S., & Rathore, S. S. (2021). Generative Adversarial Networks-Based Imbalance Learning in Software Aging-Related Bug Prediction. IEEE Transactions on Reliability. DOI: 10.1109/TR.2021.3052510.

[8] Cook, W. D., Ramón, N., Ruiz, J. L., Sirvent, I., & Zhu, J. (2019). DEA-based benchmarking for performance evaluation in pay-for-performance incentive plans. Omega, 84, 45-54.

[9] Du, G., Zhang, J., Jiang, M., Long, J., Lin, Y., Li, S., & Tan, K. C. (2021). Graph-Based Class-Imbalance Learning With Label Enhancement. IEEE Transactions on Neural Networks and Learning Systems. DOI: 10.1109/TNNLS.2021.3133262.

[10] Elyan, E., Moreno-Garcia, C. F., & Jayne, C. (2021). CDSMOTE: class decomposition and synthetic minority class oversampling technique for imbalanced-data classification. Neural computing and applications, 33(7), 2839-2851.

[11] Fernández,A., García,S., Chawla, N. V. , & Herrera, F. (2018). Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. Journal of Artificial Intelligence Research, 61, 863-905.

[12] Ferri C., Hernández-Orallo J. and Modroiu R. (2009). An experimental comparison of performance measures for classification. Pattern Recognition Letters 30(1):27–38.

[13] Fu, S., Yu, X., & Tian, Y. (2022). Cost sensitive ν-support vector machine with LINEX loss. Information Processing & Management, 59(2), 102809.

[14] Galar, M., Fernández, A., Barrenechea, E., & Herrera, F. (2013). Eusboost: enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. Pattern Recognition, 46(12), 3460-3471.

[15] García, V., Mollineda, R. A., & Sánchez, J. S. (2009, June). Index of balanced accuracy: A performance measure for skewed class distributions. In Iberian conference on pattern recognition and image analysis (pp. 441-448). Springer, Berlin, Heidelberg.

[16] He H. and Garcia E.A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21(9):1263–1284

[17]Kang, Q., Shi, L., Zhou, M., Wang, X., Wu, Q., & Wei, Z. (2017). A Distance-Based Weighted Undersampling Scheme for Support Vector Machines and its Application to Imbalanced Classification. IEEE Transactions on Neural Networks and Learning Systems, (99), 1-14.

[18]Khorshidi, H. A., & Aickelin, U. (2021). Constructing classifiers for imbalanced data using diversity optimisation. Information Sciences, 565, 1-16.

[19]Kou, G., Peng, Y., & Wang, G. (2014). Evaluation of clustering algorithms for financial risk analysis using mcdm methods. Information Sciences, 275(11), 1-12.

[20]Li, K., Wang, B., Tian, Y., & Qi, Z. (2021). Fast and Accurate Road Crack Detection Based on Adaptive Cost-Sensitive Loss Function. IEEE Transactions on Cybernetics. DOI: 10.1109/TCYB.2021.3103885.

[21]Liu WB, Zhang DQ, Meng W, Li XX, Xu F. (2011) A study of DEA models without explicit inputs. Omega, 39(5):472–80.

[22]Lomax S. and Vadera S(2013) A survey of cost-sensitive decision tree induction algorithms. ACM Computing Surveys 45:1-35.

[23]López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. Information Sciences, 250(11), 113-141.

[24]López, V., Fernández, A., & Herrera, F. (2014). On the importance of the validation technique for classification with imbalanced datasets: addressing covariate shift when data is skewed. Information Sciences, 257(2), 1-13.

[25]Luque, A., Carrasco, A., Martín, A., & de las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognition, 91, 216-231.

[26]Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(11), 2579–2605.

[27]Maurya, C. K., & Toshniwal, D. (2018). Large-Scale Distributed Sparse Class-Imbalance Learning. Information Sciences, 456, 1-12.

[28]Michael B. Cohen, Yin Tat Lee, and Zhao Song. (2021). Solving Linear Programs in the Current Matrix Multiplication Time. Journal of ACM, 68, 1, Article 3 (February 2021), 39 pages. DOI: 10.1145/3424305

[29]Mullick, S. S , Datta, S. , Dhekane, S. G. , & Das, S. . (2020). Appropriateness of performance indices for imbalanced data classification: an analysis. Pattern Recognition, 102, 107197.

[30]Napierala, K., & Stefanowski, J. . (2016). Types of minority class examples and their influence on learning classifiers from imbalanced data. Journal of Intelligent Information Systems, 46(3), 563-597.

[31]Ng, W. W., Zhang, J., Lai, C. S., Pedrycz, W., Lai, L. L., & Wang, X. (2018). Cost-Sensitive Weighting and Imbalance-Reversed Bagging for Streaming Imbalanced and Concept Drifting in

Electricity Pricing Classification. IEEE Transactions on Industrial Informatics. DOI: 10.1109/TII.2018.2850930

[32] Peng, Y., Kou, G., Wang, G., & Shi, Y. (2011). Famcdm: a fusion approach of mcdm methods to rank multiclass classification algorithms. Omega, 39(6), 677-689.

[33] Sun, L., Zhang, J., Ding, W., & Xu, J. (2022). Feature reduction for imbalanced data classification using similarity-based feature clustering with adaptive weighted K-nearest neighbors. Information Sciences, 593, 591-613.

[34] Thai-Nghe, N., Gantner, Z., & Schmidt-Thieme, L. (2011). A new evaluation measure for learning from imbalanced data. 537-542. Proceedings of International Joint Conference on Neural Networks, pp. 537–542. doi:10.1109/IJCNN.2011.6033267.

[35] Tsai, C. F., Lin, W. C., Hu, Y. H., & Yao, G. T. (2019). Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. Information Sciences, 477, 47-54.

[36] Richhariya, B., & Tanveer, M. (2020). A reduced universum twin support vector machine for class imbalance learning. Pattern Recognition, 102, 107150.

[37] Roy, A., Qureshi, S., Pande, K., Nair, D., Gairola, K., Jain, P., ... & Kakarlapudi, A. V. (2019). Performance comparison of machine learning platforms. INFORMS Journal on Computing, 31(2), 207-225.

[38] Ruiz, J. L., Segura, J. V., & Sirvent, I. (2015). Benchmarking and target setting with expert preferences: An application to the evaluation of educational performance of Spanish universities. European Journal of Operational Research, 242(2), 594-605.

[39] Sáez, J. A., Luengo, J., Stefanowski, J., & Herrera, F. (2015). SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. Information Sciences, 291, 184-203.

[40] Siers, M. J., & Islam, M. Z. (2018). Novel algorithms for cost-sensitive classification and knowledge discovery in class imbalanced datasets with an application to NASA software defects. Information Sciences.459,53-70.

[41] Song, Q., Guo, Y., & Shepperd, M. (2018). A Comprehensive Investigation of the Role of Imbalanced Learning for Software Defect Prediction. IEEE Transactions on Software Engineering. DOI: 10.1109/TSE.2018.2836442

[42] Sowah, R. A., Kuditchar, B., Mills, G. A., Acakpovi, A., Twum, R. A., Buah, G., & Agboyi, R. (2021). HCBST: An Efficient Hybrid Sampling Technique for Class Imbalance Problems. ACM Transactions on Knowledge Discovery from Data (TKDD), 16(3), 1-37.

[43] Thabtah, F., Hammoud, S., Kamalov, F., & Gonsalves, A. (2020). Data imbalance in classification: Experimental evaluation. Information Sciences, 513, 429-441.

[44] Veganzones, D., & Séverin, E. (2018). An investigation of bankruptcy prediction in imbalanced datasets. Decision Support Systems, 112, 111-124.

[45] Vuttipittayamongkol, P., Elyan, E., & Petrovski, A. (2021). On the class overlap problem in imbalanced data classification. Knowledge-based systems, 212, 106631.

[46] Wang, S., & Yao, X. (2012). Relationships between diversity of classification ensembles and single-class performance measures. IEEE Transactions on Knowledge & Data Engineering, 25(1), 206-219.

[47] Wang, N., Liang, R., Zhao, X., & Gao, Y. (2021). Cost-Sensitive Hypergraph Learning With F-Measure Optimization. IEEE Transactions on Cybernetics. DOI: 10.1109/TCYB.2021.3126756.

[48] Xie, Y. , Qiu, M. , Zhang, H. , Peng, L. , & Chen, Z. . (2020). Gaussian distribution based oversampling for imbalanced data classification. IEEE Transactions on Knowledge and Data Engineering, DOI：10.1109/TKDE.2020.2985965

[49] Zheng, Z., & Padmanabhan, B. (2007). Constructing ensembles from data envelopment analysis. Informs Journal on Computing, 19(4), 486-496.

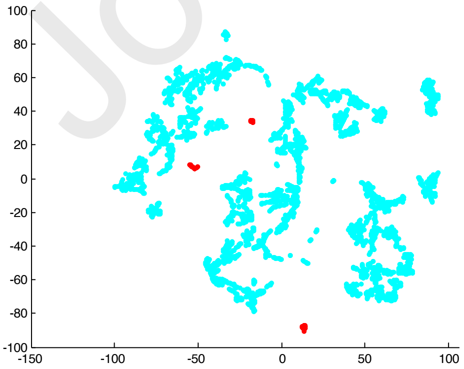
Appendix A: visual datasets bipartition

In this appendix, 12 KEEL and UCI datasets are used as examples to illustrate the three types of imbalanced data distributions (Table A-1). Visualization was implemented by t-SNE dimensionality reduction (Maaten and Hinton, 2008).

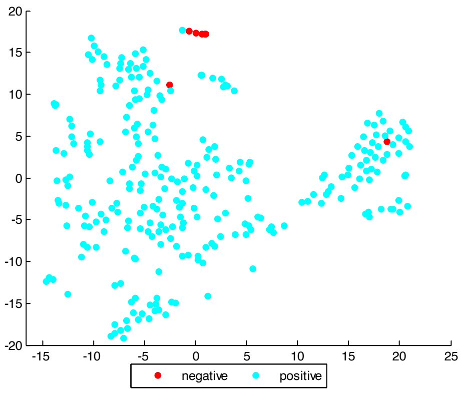
Table A-1 UCI datasets

	Data Distributions		
	Scarcity and disjunct	Overlapping	Noisy
Datasets	Shuttle 2vs5	Yeast 5	Ecoli 0_1_4_6vs5
	Ecoli 0_1_3_7 vs 2_6;	Yeast 0359vs78	Echo
	Pageblocks13vs4	Yeast 4	newthyroid1
	Pageblocks 2vs 3	Hepatitis	Glass

Scarcity and disjunct. Scarcity refers to a situation in which a class of data is significantly rare in a dataset that most classifiers cannot recognize this minority class (Fig. A-1). A disjunct is a special case of scarcity, which is characterized by dispersed small groups of minority data surrounded by data from a majority class (Fig. A-2).

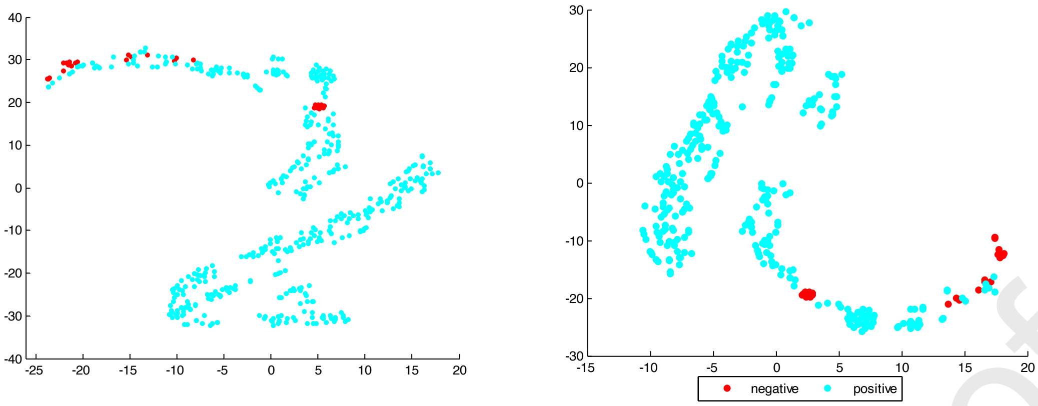


(a) shuttle 2vs5



(b) Ecoli 0_1_3_7 vs 2_6

Figure A-1 Scarcity

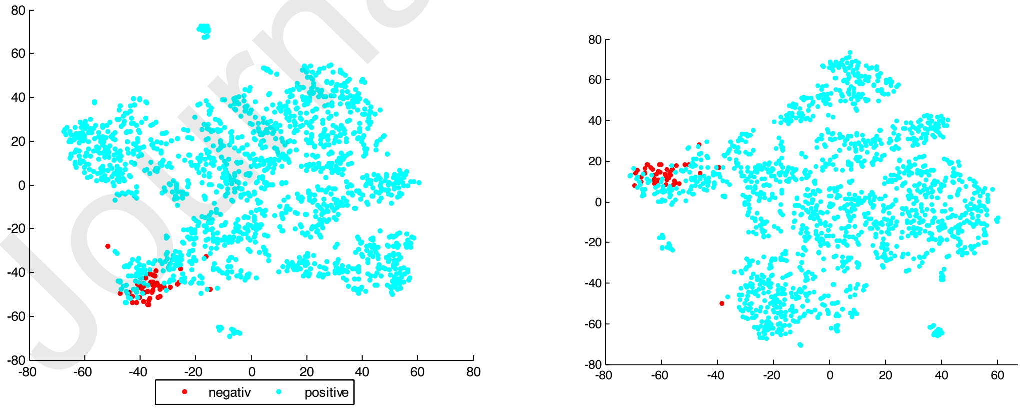


(a) Pageblocks13vs4 (b) Pageblocks 2vs 3

Figure A-2 Disjunct

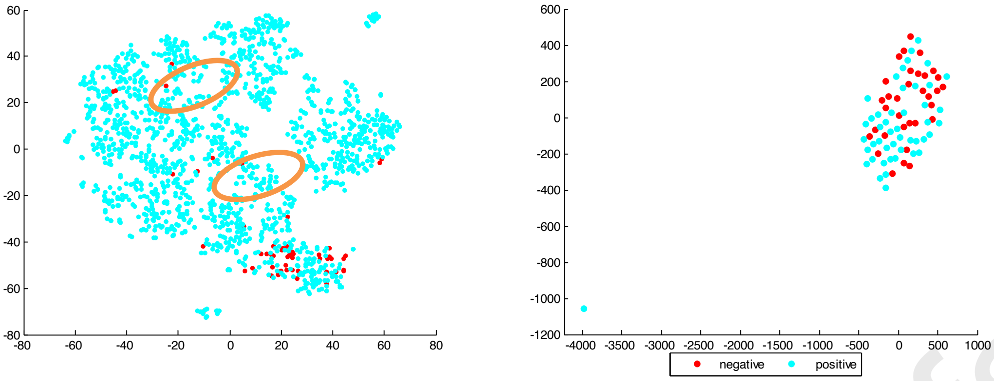
To improve the classification accuracy of the minority data, approaches such as adding more minority data by resampling or integrating several classifiers results by ensemble have been developed to address the scarcity and disjunct characteristics (Galar et al, 2013). Some classifiers can obtain higher classification accuracy of the minority class while sacrificing the accuracy of the majority class. Under these circumstances, traditional metrics that focus only on the accuracy of one class or average accuracy of two classes are not sufficient.

Overlapping: Overlapping is the area in which the minority and majority classes are evenly distributed. Overlapping data can be divided into three categories: First, the data of the minority class are located close to their own class (Fig. A-3). Second, the minority class is distributed in the middle of the other class (the circle in Fig. A-4 (a)). The third category is borderline data, which are located inside or around the border of the overlapping area of the two classes (Fig. A-4 (b)).



(a) Yeast 5 (b) Yeast 0359vs78

Figure A-3 Overlapping I

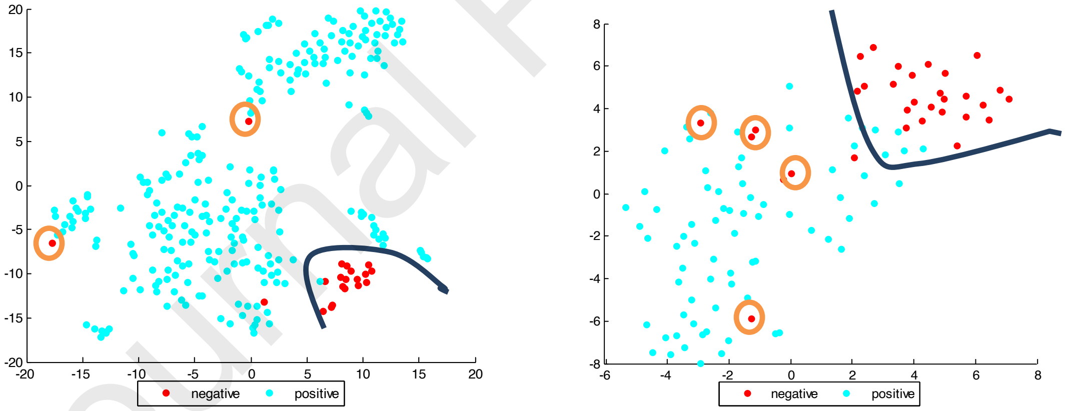


(a) Yeast 4 (b) Hepatitis

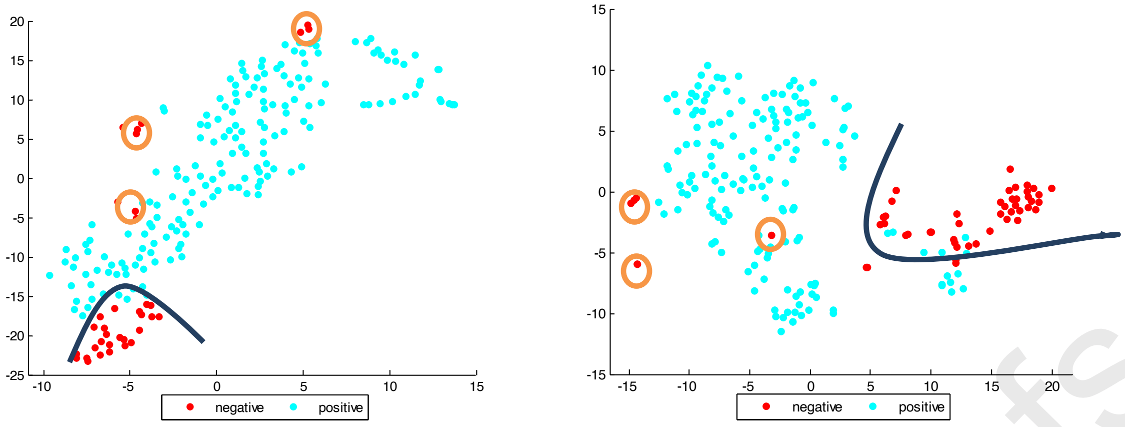
Figure A-4 Overlapping II

The performance of a classifier is often affected by the degree of overlap between classes. The higher the degree of overlap, the worse the classification result. Many algorithms have been developed to address overlapping datasets (López et al., 2014). In these studies, geometric mean or F-measure are used to evaluate classifiers, which are based on averaging accuracies and are insensitive to the improvement of the accuracy of the minority class.

Noise: In this study, noisy data refers to data that is away from a concentrated area of the minority or majority class when the two classes have clear boundaries. For examples, in Fig. A-5(a), a clear boundary is observed between the positive and negative classes, except for a few noisy negative data points.



(a) Ecoli 0_1_4_6vs5 (b) Echo

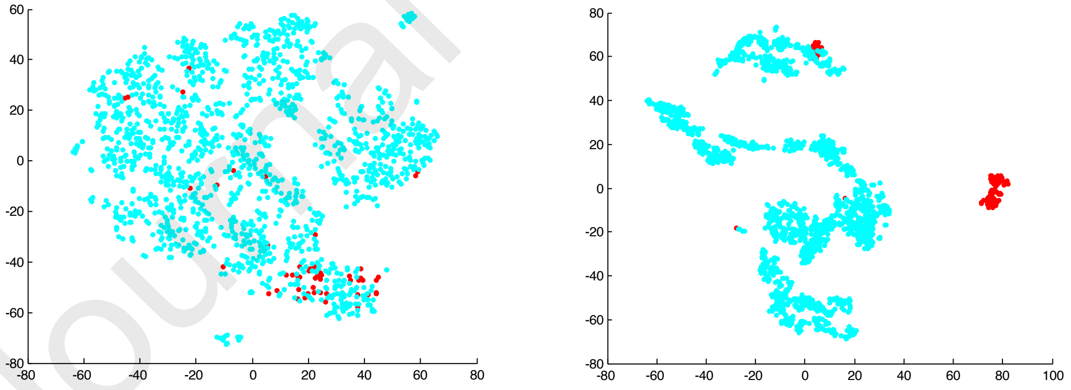


(c) newthyroid1 (d) Glass

Figure A-5 Noisy data

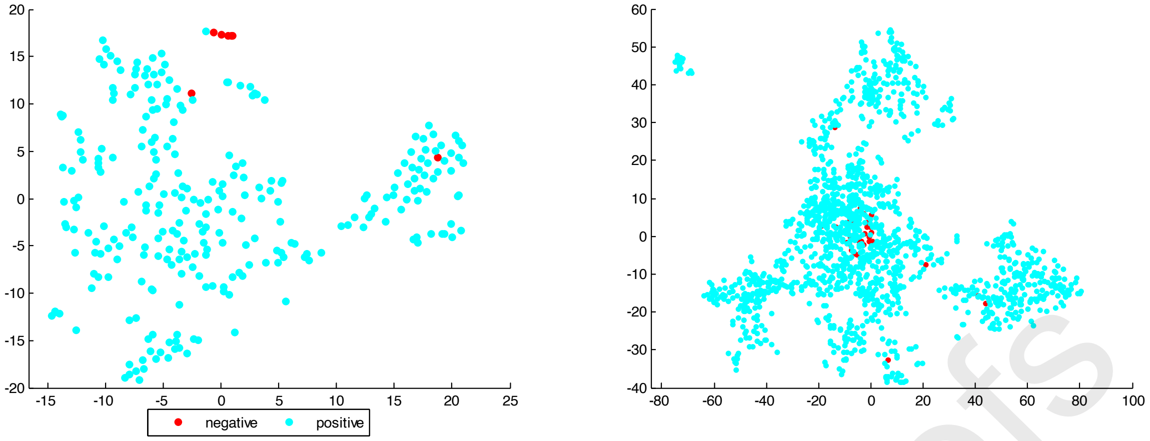
Based on the average accuracy rate of the two classes, López et al. (2013) concluded that noisy data is the most influential factor of classification results. The effect of noisy data on the efficiency of classifiers should be further explored to provide a basis for a reasonable selection of classifiers for this type of data.

However, some datasets cannot be strictly classified into certain categories. For example, Yeast4 and Shuttle0vs4 can be classified as either noisy or overlapping data (Fig. A-6) because some of the data are concentrated in an isolated area, and several data points are far away from the minority class and mixed with the majority class. Fig. A-7(b) shows a mixture of disjunct and noisy data. Fig. A7(a) can be divided into two types of distribution: disjunct and noisy. The amount of minority data is scarce, and outliers are mixed in the area of the majority data.



(a) Yeast4 (b) Shuttle0vs4

Fig. A-6 Mixture of overlapping and noisy data



(a) E. coli 0137 vs. 26 (b)Yeast6

Figure A-7 Mixture of disjunct and noisy data

Appendix B: Balancing and domain costs.

Balancing costs are the costs incurred from the misclassification of the majority and minority data. The costs of TP and TN are set to 0. The cost ratio between FN and FP is often set as N/P, where N and P are the numbers of records in the majority and minority classes, respectively (Siers and Islam, 2018). However, it is difficult to determine the optimal cost ratio that maximizes the classification accuracy. Table 3 summarizes the balancing cost matrices. The total balancing cost is $TBC = C_{FN} \times FN + C_{FP} \times FP$, and C_{FN} and C_{FP} are the balancing costs of misclassification (Veganzones and Séverin, 2018).

Table B-1 Matrix of the balancing cost

		Predicted class	
		Positive class	Negative class
True class	Positive class	0	C_{FN}
	Negative class	C_{FP}	0

Domain costs are the expenditures of misclassification records in real-life production. Siers and Islam (2018) used a case study to show that software engineers should pay actual production costs to repair the FN and FP data in NASA software defects. Additionally, the software development business should assign resources to fix the defective modules; thus, TPs also require human resource costs to correct the errors. Generally, the domain costs should be determined by the real economic cost. Theoretically, it can be a preset fixed value for simplicity.

Appendix C: (*Lemma*) Maximizing *AUC* is equivalent to minimizing the total balancing costs if the cost ratio is set as a linear function of the N/P. Further, $AUC + k \times TBC = 2$, where k is the ratio of the misclassification cost to the total number of samples.

Proof: assuming $C_{FN}/C_{FP}=N/P$ and $C_{FN}=kN$, the total balancing cost can be transformed as follows:

$$
\begin{aligned}
TBC &= C_{FN}\times\frac{e_1}{n_1}\times p_1 + C_{FP}\times\frac{e_2}{n_2}\times p_2\\
&= C_{FN}\times\frac{FN}{P}\times\frac{P}{P+N} + C_{FP}\times\frac{FP}{N}\times\frac{N}{P+N}\\
&= \frac{1}{P+N}\left(C_{FN}\times FN + C_{FP}\times FP\right)\\
&= \frac{k^{-1}}{P+N}\left(N\times FN + P\times FP\right)\\
&= k^{-1}\left(\frac{FN}{P}+\frac{FP}{N}\right)\\
&= k^{-1}(2-AUC)
\end{aligned}
$$

Thus, we have $AUC+k\times TBC=2$. ■

Based on the above relationship between the total balance cost and accuracy rate, the total balance cost index was not set as the input index of the evaluation model in our experimental analysis.

Appendix D: (*pseudocode*)

DEA-WEI evaluation algorithm

```
Input: Imbalanced datasets (ID); Classification algorithms a_i; Tested algorithms; classification results measured by selected evaluation indexes e_j.
Output: Efficiency curve (EC) composed of benchmark algorithms; distance from each algorithm to the efficiency curve (D_r).
1.  Perf_ij=compute_performance(a_i, e_j, ID)   // Calculate the performance of different
                                                    classification algorithms using evaluation indexes
2.  For i=1:1:$n$; j=1:1:$m$                     // $n$ is the number of algorithms; $m$ is the
                                                    number of evaluation indexes
      theta_i=compute_efficiency(Perf_ij) ;     // Compute the efficiency based on model (5)
  End
3. For a given algorithm $r$,
      D_r=compute-distance(h_rj+l_ri, ID);      // Compute the distance based on model (11)
      If the D_r=0,
       the algorithm is regarded as a benchmark;     //Compose the efficiency curve
       else
         continue;
  End
4.  Output EC and D_r.
```

Xiangrui Chao: Conceptualization, Methodology, formal analysis, data curation, Writing- Original draft preparation, Writing - Review & Editing

Gang Kou: Conceptualization, Supervision

Yi Peng: Conceptualization, Writing- Original draft preparation. Writing - Review & Editing

Alberto Fernández: Writing - Review & Editing