The Performance Index of Convolutional Neural Network-Based Classifiers in Class Imbalance Problem

Liu, Yanchen; Lai, King Wai Chiu

Published in:
Pattern Recognition

Published: 01/05/2023

Document Version:
Post-print, also known as Accepted Author Manuscript, Peer-reviewed or Author Final version

Published version (DOI):
10.1016/j.patcog.2022.109284

Publication details:
Liu, Y., & Lai, K. W. C. (2023). The Performance Index of Convolutional Neural Network-Based Classifiers in Class Imbalance Problem. *Pattern Recognition*, *137*, Article 109284. https://doi.org/10.1016/j.patcog.2022.109284

The Performance Index of Convolutional Neural Network-Based Classifiers in Class Imbalance Problem

Yanchen Liu and King Wai Chiu Lai*

Department of Biomedical Engineering, Centre for Robotics and Automation
City University of Hong Kong, Kowloon Tong, Hong Kong Special Administrative Region

---

Abstract

Class imbalance is a common problem in many classification domains. This paper provides an evaluation index and one algorithm for this problem based on binary classification. The Model Performance Index (*MPI*) is proposed for assessing classifier performance as a new evaluation metric, considering class imbalance impacts. Based on *MPI*, we investigate algorithms to estimate ideal classifier performance with a fair distribution *(1:1)*, referred to as the ***Ideal Model Performance Algorithm.*** Experimentally, compared with traditional metrics, MPI is more sensitive. Specifically, it can detect all types of changes in classifier performances, while others might remain at the same levels. Moreover, for the estimation of classifier performances, the algorithm reaches small differences between predictions and the values observed**.** Generally, for ideal performances, it achieved error rates of 0.060% - 1.3% for rare class in four experiments, showing a practical value on estimation and representation on the classifier performances.

Keywords: Deep Learning; Convolutional Neural Network; Class Imbalance; Class Balance Index; Model Performance Index

---

1. Introduction

The classification of binary imbalanced data is a challenging problem in real-world applications of classifiers. In the imbalanced data distribution, there is a significantly higher number of samples in one class than in the other classes, which is referred to as class imbalance [1]. For a training dataset, it is considered classed imbalanced when the distributions of the samples across the categories are not equal [2]. It usually occurs in domains such as computer vision [3], fraud detection [4, 5], and other fields such as credit risk evaluation [6] in internet financing and disease diagnoses in the medical domains [7, 8]. Specifically, the class imbalance is very common in medical domains. This is mainly caused by the challenges and practical limitations of data collection [7]. Due to the diseases' low prevalence, it is time-consuming and challenging to acquire sufficient data on patients representing certain diseases, such as cancer [7] and diabetic retinopathy [9]. Therefore, we always suffer from unbalanced learning when building a decision support system for practical problems [10].

Data imbalance has a significant detrimental effect on training traditional classifiers [11] and their performance, producing a biased classification result [12]. This causes classifiers to perform poorly on rare classes [13]. For traditional classifiers such as neural network-based classifiers, it is assumed that those classifiers are trained with a relatively fair dataset distribution [8]. Because of the lack of samples in the rare class, the performance of the classifiers is far from the true

* Corresponding author.

E-mail addresses: kinglai@cityu.edu.hk (King Wai Chiu Lai).

performance. This is due to the limited effect on the overall accuracy caused by the small number of minority classes in the training set [14]. For instance, in disease diagnoses such as mammography, traditional classifiers tend to consider all mammogram images as normal cases and ignore those abnormal cases to achieve high accuracy [15].

Therefore, it is crucial to assess classifier performance with proper metrics in a class imbalance problem, which can be considered an indicator of model performance. Depending on data characteristics, the standard metrics used for evaluating classifier performance may suffer from different forms of distortion [2]. Originally, accuracy and error rate can be considered among the standard evaluated metrics, but the main drawback of this approach is that the overall accuracy is easily biased to the majority class with larger samples and ignores the rare class with lower samples [16], making it difficult for the classifier to perform well on the rare class [17]. Therefore, to overcome this problem, several solutions have been proposed, including class-balanced accuracy (CBA) [18, 19], index of balance accuracy [20, 21], normalized precision rate [22], relevance-based evaluation approaches [23], and multiclass performance scores [24].

The effect of class imbalance on classifier performance also motivates the development of other well-known indices, which can be considered the most related and widespread metrics in class imbalance problems, including recall, specificity, precision, F-measure, geometric mean (G-means) [25], and single-threshold AUC. There are also other metrics that measure the agreement and disagreement among observers, such as Cohen's kappa [26] and Krippendorff's α-reliability [27, 28]. Among them, G-Means and single-threshold AUC are considered the best null-biased

metrics [29]. Although these two metrics are popular candidates for use with imbalanced datasets, they focus only on classification success, without considering classification errors [29]. Next, the second-best metrics are the *Matthews correlation coefficient* (MCC) [30] and Markedness (MK). These metrics can produce more informative and truthful performance by producing high scores only if the classifier obtains good performance in all confusion matrix categories [29, 31]. Precision and recall are considered the third best clusters since they are highly biased, which should be avoided in datasets with large imbalances [29]. Based on metrics, the F-measure arises as one of the most popular evaluation metrics for class imbalance problems [32], as it considers the classifier performance as a weighted average of precision and recall [33]. Although these metrics are popularly used in the class imbalance problem, most of them are constrained to the metrics based on the confusion table from the testing dataset, limiting their use in training datasets.

There are four main criteria to evaluate classifier performance in class imbalanced problems: the minimum cost criterion (MC), the criterion of the maximum geometry mean (MGM) and the maximum sum (MS) of the accuracy on both the minority and majority classes, and the criterion of the receiver operating characteristic (ROC) analysis [17]. The usage of MC is restricted because this index highly relies on the cost of misclassification in real cases, which is generally unknown [34]. The MGM suffers from difficulty in automatic optimization because it maximizes the geometric mean of accuracy with a nonlinear form [35]. MS, instead of the geometric means, maximizes the sum of the accuracy on both the positive and negative classes in a linear manner [36]. Among them, ROC analysis and its association, the area under the curve (AUC), are the most

common metrics used to assess classifier performance [34]. Different from accuracy, the AUC is not biased to the majority class with larger samples, so it assesses the classifier performance in a relatively fair approach. ROC curves illustrate the diagnostic performance of a binary classifier when its discrimination threshold changes.

These evaluation metrics and criteria seek an accurate assessment of classifier performance, but some of them are not suitable for dealing with imbalanced distribution tasks [17], such as accuracy. Even though some of them consider the performance of imbalanced classes in a relatively fair way, they might take into account the current classifier performance without considering the impacts caused by class imbalance distribution, which means that different distributions of datasets may obtain the same performance. For instance, with the same type of dataset, a classifier trained with a fair distribution (1:1) might have the same AUC as a model trained with an imbalanced distribution (1:10). Furthermore, the current evaluation metrics and criteria make it difficult to estimate the classifier performance trained with a fair distribution (1:1) due to the lack of training samples in rare classes.

Therefore, considering these points, there are two purposes in our work. One is to evaluate and estimate actual classifier performance, considering class imbalance impacts. In this work, we propose the Model Performance Index (MPI) by taking the dataset balance into consideration. This allows us to distinguish classifier performances in different class imbalance statuses even when traditional metrics fail to distinguish them. The second is to estimate the classifier performance with sufficient training data using limited training samples. To address this problem, an algorithm

has been investigated to estimate the model performance in an ideal case with a fair distribution (1:1), referred to as the ***Ideal Model Performance Algorithm***.

2. Methodology

2.1 Definition of the Evaluation Metrics

2.1.1 Definition of the Failure Index α in the Confusion Matrix

The evaluation method is a significant factor in the assessment of model performance in the classification task. In the binary classification problem, a confusion matrix, also known as an error matrix, allows visualization of the performance of a classifier. In a confusion matrix, **0** is defined as the Negative Class, which we also call the **Majority Class** or the **Non-Objective Class** in this paper, with a large number of samples in this class. In addition, **1** represents the **Positive Class**, referred to as the **Rare Class** or the **Objective Class,** with a small number of samples in this class. A confusion matrix reports the numbers of **true positives (*TP*)**, **false positives (*FP*)**, **false negatives (*FN*),** and **true negatives (*TN*)**.

Based on the confusion matrix, the worst performance of the classifier can be calculated. The **worst performance** is defined as the performance when the binary classifier fails to detect all positive cases, considering them as negative cases. We define this performance as a new parameter, referred to as the ***General Failure Index (α)***. The General Failure Index α can be considered the minimum value of the F_β-score when the binary classifier considers all positive cases as negative cases, as shown in **Table 1.** This is highly related to the test dataset distribution.

Table 1 The Confusion Matrix of the General Failure Index (α)

		Predictions	
		1	0
True Label	1	(True Positive) TP=0	(False Negative) FN
	0	(False Positive) FP=0	(True Negative) TN

where TP, FN, FP, TN refer to the true-positive, false-negative, false-positive, and true-negative cases, respectively. For the worst performance of the binary classifier, the binary classifier considers all positive cases as negative cases, where *TP=FP=0*.

From the confusion matrix of the General Failure Index in **Table 1**, $\boldsymbol{\alpha}$ can be defined as:

$$\boldsymbol{\alpha} = F_{\beta(min)} \tag{1}$$

where $F_{\beta(min)}$ is the minimum value of the *F-score* for the majority class, referring to the worst performance of the classifier in the binary task problem, which can be obtained by:

$$Precison = \frac{TN}{TN + FN} \tag{2}$$

$$Recall = 1 \tag{3}$$

$$F_\beta = (1 + \beta^2)\frac{Precision \cdot Recall}{(\beta^2 \cdot Precison) + Recall} \tag{4}$$

where β is used to control the weighting between precision and recall, when F_β considers recall as β times equally important as precision, which can be expressed as:

$$\beta = \frac{R}{P}, \qquad where\ \frac{\partial E}{\partial P} = \frac{\partial E}{\partial R} \tag{5}$$

$$E = 1 - \frac{PR}{\alpha R + (1 - \alpha)P}, \qquad where\ \alpha = \frac{1}{\beta^2\ +\ 1} \tag{6}$$

In some specific situations, for objectively evaluating the model performance, there are usually the same numbers of samples in both classes for testing (1:1 in the testing dataset). In this case, when the classifier considers all positive cases as the negative cases in prediction (*TN=FN*, *TP=FP=0*, $\beta = 1$), it will show the worst performance in the binary task. We defined this specific worst performance as the **specific 'Failure Index'** in these specific situations. Therefore, the specific Failure Index **α** for the Majority Class can be calculated as:

$$\alpha = F_{1(min)} \approx 0.667 \tag{7}$$

where we refer to (7) as the specific 'Failure Index' for majority class (*β=1*).

2.1.2 Definition of the Class Balance Index (CBI)

Based on the Failure Index (α), we propose a parameter called the ***Class Balance Index (CBI)***. ***CBI*** is proposed for the evaluation of the class balance scores in the current status for a specific classifier, considering distribution impacts between two classes, which can be defined as:

$$CBI = \frac{F_\beta - \alpha}{x * \alpha} \tag{8}$$

where F_β defines the classifier performance in the current class imbalance status, α represents the Failure Index, and *x* represents the ratio between the number of samples in the Majority Class and the Rare Class. In this work, we use a specific Failure Index for calculation.

$$x = \frac{|X_i|}{|X_j|} \tag{9}$$

where $|X_i|$ represents the number of samples in the Majority Class and $|X_j|$ represents the number of the samples in the Rare Class.

The *CBI* can be calculated into a certain range of values. For instance, if there are the same number of positive and negative cases, which both refer to ***n*** in the test dataset, the classifier considers all positive cases as negative cases (*TP=0, FP=0, FN=n, TN=n*), where

$$F_\beta = \alpha \tag{10}$$

According to (8), the minimum value of *CBI* can be obtained as 0, which defines the poorest class balance performance, referring to a skew class imbalance status. Specifically, when we have the largest F_β as 1 in ideal cases with a balance distribution (*x=1*), the maximum value of *CBI* is close to 0.5, which defines the highest-class balanced performance, referring to a relatively fair distribution.

2.1.3 Definition of the Model Performance Index (MPI)

Based on the *CBI*, we propose a new index called the ***Model Performance Index (MPI)*** considering the impacts of the imbalance distribution, which refers to the classifier performance with the imbalance distribution impacts. Physically, the *MPI* is designed to balance the impacts of imbalance distribution (*CBI*) and current classifier performance (*F1-score*). Inspired by the F_β-*score*, $\boldsymbol{MPI}_\mu$ is defined as a weighted average of the *Class Balance Index* (*CBI*) and current model performance F_β:

$$MPI_\mu = f(F_\beta,\ CBI)$$

The MPI_{μ} can be expressed as follows:

$$MPI_{\mu} = (1+\mu^2)\frac{F_{\beta}\cdot CBI}{(\mu^2\cdot F_{\beta})+CBI} \tag{11}$$

where *CBI* is considered μ times as important as F_{β}. Physically, *MPIμ* considers the impacts of imbalance distribution μ times as important as the current classifier performance.

Figure 1 shows the importance of class imbalance in *MPI*, which depends on μ. When μ is set to *0 (μ=0)*, *MPI* considers the current model performance only. When it refers to a value between *0* and *1* (*0<μ<1*), *MPI* considers more current model performance than class imbalance impacts. When $\boldsymbol{\mu}$ is set to *1 (μ=1)*, *MPI* considers current model performance and class imbalance impact equally. Finally, when $\boldsymbol{\mu}$ refers to a value larger than *1* (***μ>1***), *MPI* considers more class imbalance impacts than the current model performance.

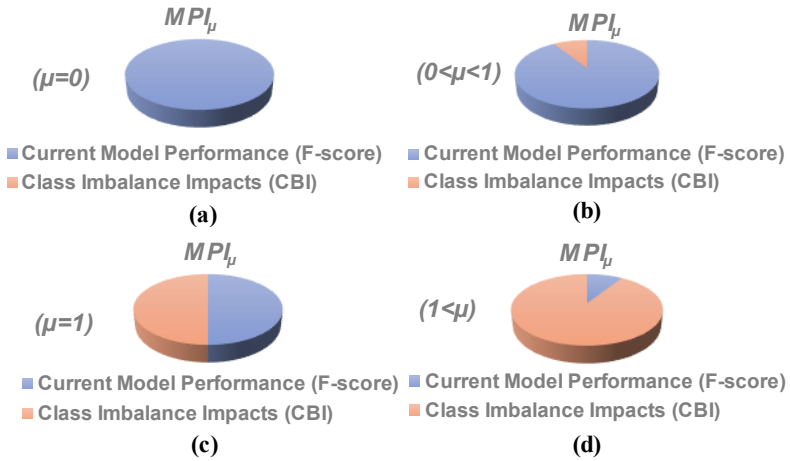


Figure 1 The importance of class imbalance in *MPI*, which depends on μ. (a) $\mu=0$. (b) $0<\mu<1$. (c) $\mu=1$. (d) $\mu>1$.

For further analysis, the first derivative of MPI_{μ} can be calculated as:

$$MPI' = \frac{dMPI}{d\mu} = \frac{2\mu}{\left(F_{\beta}*\mu^2+CBI\right)^2}*\left(CBI-F_{\beta}\right)*CBI*F_{\beta} \tag{12}$$

Because *CBI* is smaller than the current F_β*-score*, which indicates $MPI' < 0$, MPI_μ belongs to a monotonically decreasing function.

For instance, **Figure 2** shows the *Maximum MPI Performance* (*MMP*) function with respect to increasing μ in ideal cases, where the classifier obtains the best performance by $F_\beta = 1$ with the same numbers of samples in two classes (*x=1*) and the highest-class balance scores (*CBI=0.5*,$\alpha = 0.667$). This shows that MPI_μ decreases as μ increases. Physically, μ refers to the importance of class imbalance (CBI) in MPI performance. With the increment of μ, the impacts of class imbalance will be improved, considering impacts on the *CBI* more than the current classifier performance (*F-score*). This leads to the deterioration of the model performance.

In this work, we consider current classifier performance more than class imbalance status ($0<\mu<1$). This is mainly because some classifiers can still obtain great performance when suffering from a very skewed class imbalance status, and class imbalance has limited impacts on the classifier performance in these cases. In the definition of the *MPI*, we consider the current performance with a partial class imbalance impact ($\mu=0.1$), where *MMP* refers to a value close to 1 (MMP≈0.99).

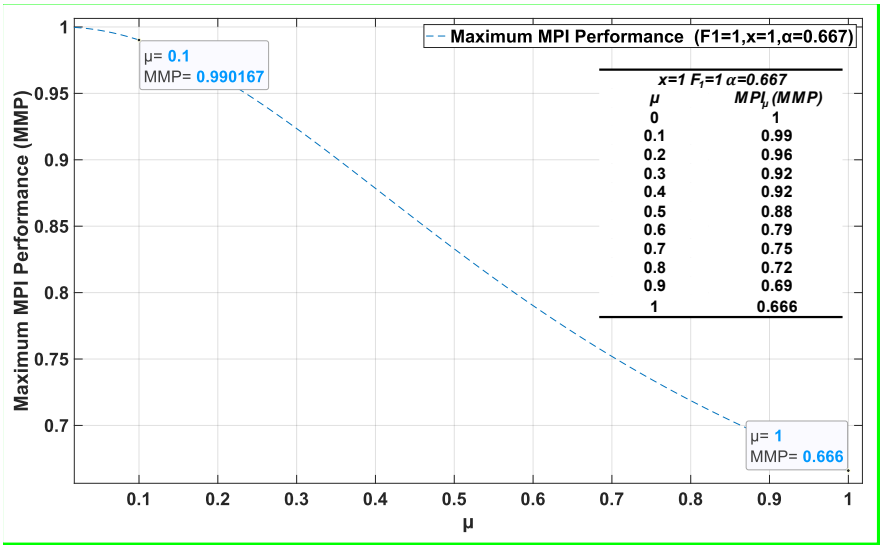


Figure 2 The maximum *MPI* performance (*MMP*) with respect to μ. μ is set to *0.1*.

2.2 Definition of the Ideal Performance Function Algorithm

2.2.1 Basic Distribution Functions on the Imbalanced Model Performance Index (MPI)

In this section, we define a ***Distribution Function*** to express the changes in classifier performances with respect to imbalanced distribution statuses, referred to as ***Basic Distribution Functions***. The Distribution Functions explain the performance changes with respect to the distribution ratios.

From the previous sections, the $\boldsymbol{MPI_{\mu}}$ has been developed to explain the classifier performance, balancing both distribution impacts as well as the current model performance. From (11), it can be reorganized into:

$$MPI_{\mu} = \frac{1}{\frac{\alpha\mu^2}{(1+\mu^2)(F_{\beta}-\alpha)}x + \frac{1}{(1+\mu^2)F_{\beta}}} \tag{13}$$

Where α represents the *Failure Index, x* represents the ratio between samples in two classes, F_{β} represents the model performance in the current class imbalance status, *CBI* represents the Class Imbalance Index, and μ represents the weights of the *CBI* in the MPI_{μ}.

Furthermore, (13) can be considered into the functions with two parameters, referred to as ***Basic Distribution Functions***:

$$MPI_{\mu} = \frac{1}{a * x + b} \tag{14}$$

where

$$a = \frac{\alpha\mu^2}{(1+\mu^2)(F_{\beta}-\alpha)}, b = \frac{1}{(1+\mu^2) * F_{\beta}} \tag{15}$$

Physically, according to (14), the classifier performance MPI_{μ} deteriorates as the distribution bias *x* increases. The sensitivity and effectiveness of the *MPI* are further discussed in **Appendix C**.

Moreover, we can obtain the range of b by $(1+\mu^2) * F_{\beta}$. For instance, when $\mu = 0.1$ and $0.667 \leqslant F_{\beta} \leqslant 1$, then the range of b can be calculated as:

$$0.99 \leqslant b \leqslant 1.484$$

Specifically, in this paper, when the training data of the Rare Class and the Majority Class are sufficient and in a fair distribution (1:1), the *MPI* of the trained classifier is defined as the $\boldsymbol{MPI^{ideal}}$ (*x=1*), which can be expressed by:

$$MPI^{ideal} = \frac{1}{a+b} \tag{16}$$

2.2.2 Quadratic Model Performance Function on Imbalanced MPI

With the basic distribution functions, the classifier performance in a class imbalance status can be presented. Specifically, the class performance in fair distribution *(1:1*), also called ***Ideal Model Performance*** (***MPI^{ideal}***), can be calculated and estimated roughly.

However, due to the variety of training processes, the classifier performance will vary greatly even when training in the same dataset. Considering this point, we introduce a new function called the ***Quadratic Model Performance Distribution Function*** based on basic distribution function, and it minimizes the impacts of the performance variety physically by the defined **Adjustment Factor $\boldsymbol{\varepsilon}$**.

This new function can be defined as:

$$\boldsymbol{MPI_{\mu} = \frac{1}{\varepsilon * x^2 + a * x + b}} \tag{17}$$

where $\boldsymbol{\varepsilon * x^2}$ ($\boldsymbol{\varepsilon > 0}$)is used to improve the precision of the $\boldsymbol{MPI_{\mu}}$ performance. $\boldsymbol{\varepsilon}$ is named the ***Adjustment Factor***. *a* refers to the coefficient of ratio *x*, a constant larger than 0.00198($a \in [0.0198, +\infty)$) for the Majority Class and larger than -0.0099 ($a \in [-0.0099, +\infty)$) for the Rare Class. *b* refers to a constant with an initial range between 0.99 and 1.48 for the Majority Class ($b \in [0.99,1.48]$) and larger than 0.99 for the Rare Class ($b \in [0.99, +\infty)$).

2.2.3 Important Distribution Points on Imbalanced Model Performance

Based on the basic distribution function, we present a new concept of ***Important Distribution Points***, which is a set of high-quality important distribution data points. Those points can perform an accurate representation of the classifier performance changing with respect to distribution ratio *x*. They are crucial for representing the classifier performance changing with respect to distribution ratios. Physically, a set of high-quality important distribution points represents accurately, while a set of poor ones leads to high errors in those representations.

The important points are taken from different levels of classifier performances, as shown in **Figure 3**, using the defined ***MPI Gap Function.*** The ***MPI Gap Function*** calculates a set of high-quality important points in classifier performances that change with respect to distribution ratio *x*. This function considers the difference in the performances between two different class imbalance statuses, which can be expressed by:

$$x_i = t_i x_0 \; ; \; f(t_i) = \frac{1}{a * x_0 + b} - \frac{1}{a * x_i + b} - G_i \tag{18}$$

where $\boldsymbol{x_0}$ represents the distribution ratio between the Majority Class and the Rare Class in the first class imbalance status, $\boldsymbol{x_i}$ represents the distribution ratio in other class imbalance statuses, G_i represents the difference in *MPI* performances between models in two class imbalance statuses, t_i refers to the ratio between x_i and x_0, and a and b refer to coefficients of x_0 and x_i. The calculation of the important data distribution points has been provided in ***Algorithm I*** in Appendix A.

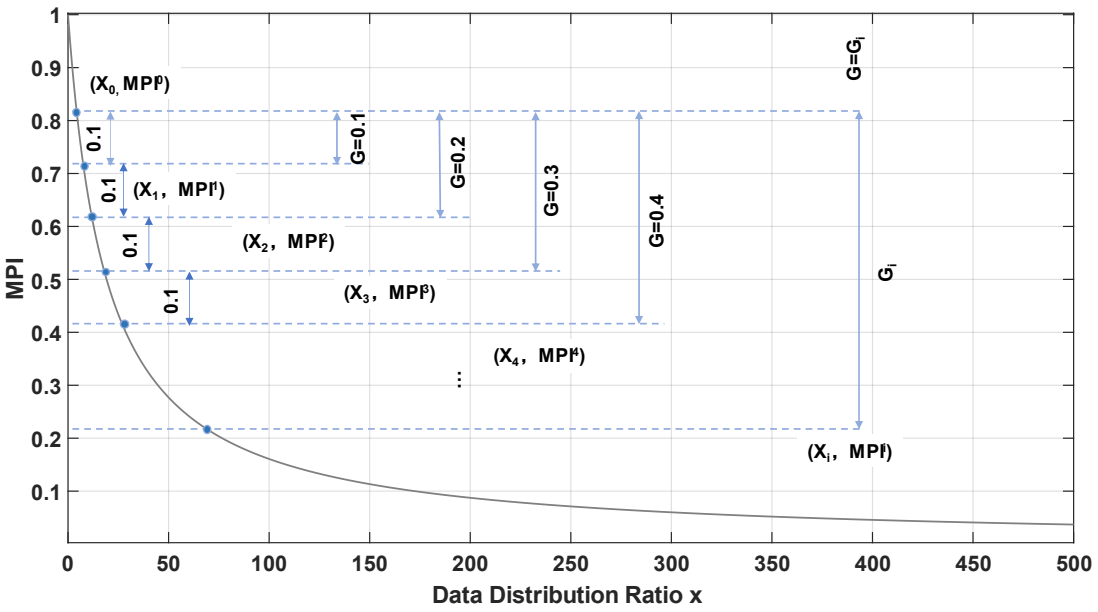


Figure 3 The MPI Performance of two models in different distributions differs by G_i. The important points are taken from different levels of classifier performances.

2.2.4 Ideal Performance Function Algorithm

With those important distribution points and Quadratic Model Performance, we present an ***Ideal Performance Function Algorithm*** to predict the imbalanced and ideal classifier performances (*1:1*), evaluating the prediction results by newly defined evaluated functions. The ***Ideal Performance Function Algorithm*** is shown in ***Algorithm II***.

In the ***Ideal Performance Function Algorithm***, the first important distribution points ($x_{0,}MPI^0$) ($x_0=10$), referred to as the ***Initial Important Distribution Point***, are considered as the input, and the Ideal Model Performance (MPI^{ideal}) and its errors are the output.

First, the important distribution points are obtained through ***Algorithm I***. Based on MPI^0, we obtain performances in other important imbalance distribution statuses by G_i. Specifically, if the classifier shows a great performance on the first imbalance distribution status ($MPI^0 > 0.6$), a small number of samples in the Rare Class would be required for obtaining the important distribution points ($G_i > 0,\ t > 1,\ MPI \geqslant 0.6$). In contrast, if the classifier shows poor performance on MPI^0 ($MPI^0 < 0.6$), more samples in the Rare Class should be required to obtain good performance ($0 < t < 1,\ G_i \neq 0,\ MPI < 0.6$). Physically, a high value of *MPI* presents a great model capability on the classification task, while a poor performance is no value at all.

Next, we obtain all model performances in those important points outputted by ***Algorithm I***, followed by a clearance on those obtained classifier performances. This is mainly because if the *MPI* performance of a model is too low (lower than 0.1), the model also fails to classify the Rare Class and the Majority Class practically. Next, based on the cleared model performances, the boundary conditions and the requirements on the evaluation functions, we use the quadratic distribution function for curve fitting and estimation on the classifier MPI^i (MPI^{ideal}) performance. Finally, the predicted classifier performance is evaluated by errors calculated by the evaluated functions.

Algorithm II: Ideal Model Performance Algorithm

1. ***Input***: The Initial Class Imbalance Status (x_0, MPI^0) in first class imbalance status, the Classifier Performance Difference G_i (G_i=0.1,0.2,0.3,0.4).

   If $0.6 \leqslant MPI^0 \leqslant 1$, # *If the initial classifier performance is good*

   $MPI^i = MPI^0 - G_i$, $(0 < i \leqslant 4)$ # *less data required for training*

   Else # *If the initial classifier performance is poor*

   $MPI^i = MPI^0 \pm G_i$, $(0 < i \leqslant 4)$ # *more data required for training*

2. ***Output***: Ideal Model Performance ($x=1$), errors

3. Obtain Important Distribution Points using **Algorithm1**: $\{x_i, MPI^i\}$

4. Clearing Data: # *Clearing useless models*
5. **For** i ← 1, **do**:
6. **if** $MPI^i > 0.1$, # *if the classifier performance is more than 0.1, it will be stored.*
7. **then**: $MPI^i = MPI^i$
8. ***Else***, # *if the classifier performance is less than 0.1, it will be removed.*
9. *Continue.*
10. End for
11. (x_i, MPI^i) $(3 \leqslant i \leqslant 4)$ # *Three to four points for Next Step.*

12. Getting the model performance of the important distribution points by training

13. Curve Fitting: (x_i, MPI^i) $(3 \leqslant i \leqslant 4)$, Quadratic Distribution Function
14. ε,a,b ←(x_i, MPI^i); $0 < \varepsilon$; $MPI^i = \frac{1}{\varepsilon * x_i^2 + a * x_i + b}$
15. **If** Rare Class
16. $a \in [-0.0099, \infty)$, $b \in [0.99, \infty)$
17. **Else if** Majority Class
18. $a \in [0.0198, \infty)$, $b \in [0.99, 1.48)$
19. ***If*** R-square>0.98
20. Output:ε,a,b
21. Else
22. Retrain in **Step 10**

23. Getting the Performance of the Ideal Point (1:1):
24. $MPI^{ideal} = 1/(\varepsilon + a + b)$
25. **Output**: MPI^{ideal} #*output classifier performance in a fair distribution (1:1)*
26. **Evaluation**: $errors = |MPI_{true} - MPI_{prediction}|$

3. Experiment

Two objectives are considered in this paper. The first objective is to assess the classifier performance considering distribution impacts, and the second objective is to predict the **Ideal Model Performance** (MPI^{ideal}) when suffering from a limitation of samples in the Rare Class. We have proposed indices, distribution functions, and an algorithm in the previous sections. To test those proposed measures, we performed experiments on four benchmark datasets and evaluated our defined evaluation functions.

3.1 Training Dataset and Neural Network

In the experiment, we used four benchmark datasets, MNIST [37], CIFAR-10 [38], Cats vs. Dogs [39], and one medical dataset, Retinal optical coherence tomography [40], for training and testing networks. Since MNIST and CIFAR10 are 10-class samples and we focused on the two-class classification task in this work, two specific classes were chosen from those benchmark datasets, including '3' and '8' from MNIST and Plane and Automobile from CIFAR-10. The distribution ratios were adjusted in the experiments. For MNIST and CIFAR-10, the number of samples in the Majority Class was fixed at 5000. For more complex datasets, Cat vs. Dog and Retinal OCT in medical imaging, the number of samples in the Majority Class was fixed at 10000.

For each dataset, different CNNs with a set of hyperparameters were utilized for training. We used four well-known neural network architectures: Traditional Convolution Neural Network (VGG16), Residual Neural Networks (ResNets50), and Densely Connectedly Neural Network (DenseNets121 & DenseNets201). These architectures are independent of the classifier, and they

can be combined with fully connected layers or standard classifiers to perform classification tasks in class imbalance. All of the neural networks used in the study were feedforward neural networks with several hidden classification layers and a single node for outputs. All experiments were performed in the Jupyter Notebook in Docker using an NVIDIA DGX Station Version 4.2.0 (GNU/Linux 4.15.0-66-generic x86_64) workstation. TensorFlow code to reproduce the results is available at *https://github.com/cityuhknrl2022/MPI-performance*.

In the training process, some values remained fixed (i.e., all samples resized to 224×224 before feeding into networks; all networks were trained from random initialization of weights without pretraining; all classifiers were trained by the Adam optimization method; binary-cross entropy was used as a loss function; all classifiers were trained for a maximum of 200 epochs or until the training loss was less than 0.01; the threshold between two classes was fixed at 0.5). Furthermore, to reduce the errors caused by the training process, several requirements for the training result should be followed: 1) The training process should be converged. 2) The local minimum should be avoided during the training process. 3) The training loss should be less than 0.01. 4) Each model should be trained several times for precise evaluation.

3.2 Evaluation Function

The performances of the proposed measures in the experiments were validated by the test datasets and evaluated by the **Evaluation Functions.** In this section, we introduce two evaluation functions, **Ideal Model Performance Evaluation Functions** and **Distribution Evaluation Functions.**

3.2.1 Metrics Evaluation of Ideal Model Performance

The metrics evaluation of ideal model performance is defined as the errors between the true *MPI* (MPI^i_{true}) and its corresponding predicted *MPI* ($MPI^i_{prediction}$) of the models when trained with a specific class imbalance status, referred to as **Ideal Model Performance Evaluation Functions**. From the test dataset, the performance of the models (True MPI) can be obtained by the calculation using the **Model Performance Index**. Its corresponding Predicted MPI can be obtained by the proposed ***Ideal Performance Function Algorithm***. Based on the **True MPI** and its corresponding **Predicted MPI,** we can obtain the absolute value of the loss between the prediction results and grounding truth to represent this value, which can be defined as:

$$errors = |MPI^i_{true} - MPI^i_{prediction}|$$

where MPI^i_{true} refers to the true value obtained by the MPI performance of the model by practical training, which can be calculated by the Class Balance Index (CBI_i) and *F-score* performance, and $MPI^i_{prediction}$ refers to the prediction results using the ***Ideal Performance Function Algorithm***. The errors range from 0 to 1, where 0 indicates the perfect prediction of the model performance, and 1 indicates the totally wrong prediction of the model performance.

3.2.2 Evaluation of Distribution Functions

The Ideal Model Performance is highly correlated with the distribution functions. In addition to the evaluation of the Ideal Model Performance, we also use the evaluation functions to test the distribution functions. The Residual Sum of Squares (RSS), Coefficient of Determination (R^2),

Adjusted Coefficient of Determination ($\bar{R}^2$) and Root-Mean-Square Error (RMSE) are used for a further validation of the distribution functions. The RSS and RMSE are utilized to measure the differences between the sample value or population values predicted by the distribution function and the values observed. R^2 and $\bar{R}^2$ represent how well observed outcomes are replicated by the distribution functions.

4. Results

4.1 Experimental Results on Metrics

4.1.1 Class Balance Index and Model Performance Index (CBI and MPI)

Figure 4 reports the experimental results of the *CBI* and *MPI* in the four datasets. Both the *CBI* and *MPI* decrease with respect to the increment of distribution ratio *x*. Physically, with increasing distribution ratios, both the distribution imbalance (*CBI*) and classifier performance (*MPI*) worsen. Specifically, the *CBI* scores show a sharp decrease to a small value at first, and it gradually slows until it reaches the worst balance scores (*CBI=0*), which refers to the worst distribution imbalance.

However, MPI might still yield a certain value when CBI remains a poor value. This is mainly because the MPI index considers a weighted combination of both imbalance impacts and current performance. For instance, as results in MNIST (**Figure 4.a**), although the Class Balance Index (*CBI*) shows a poor value on the class balance status as close to zero (*CBI=0.00414*), *MPI* still has a relatively certain score on the classification task (*MPI=0.290*). This is mainly because of the

great classification performance of the classifier itself in the current class imbalance status (F_1=*0.936*), and its poor class balance status is balanced by considering its great performance in the *MPI* scores.

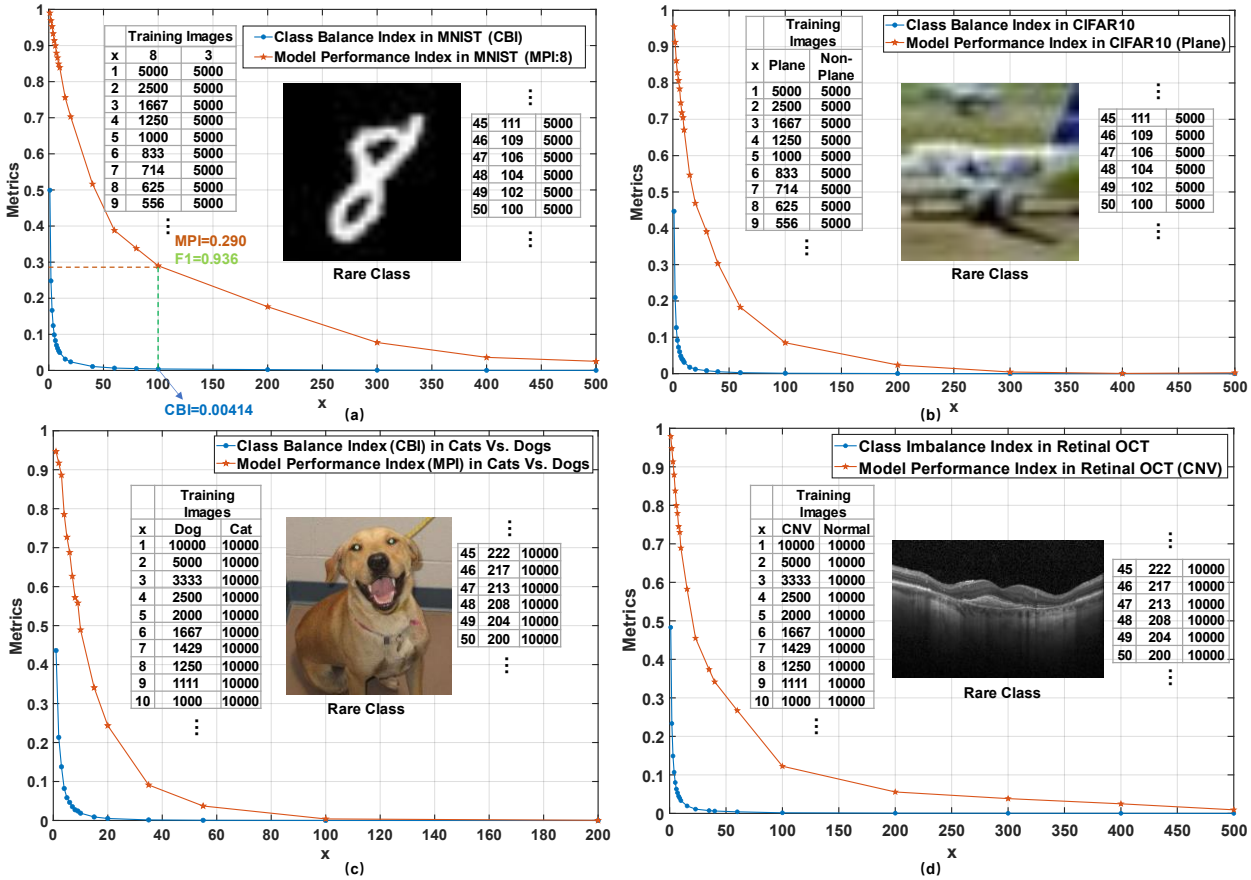


Figure 4 Class Balance Index (CBI) and Model Performance Index (MPI) in the four datasets. (a) MNIST. (b) CIFAR-10. (c) Cats & Dogs (from *Kaggle Challenge in 2013*). (d) Retinal OCT (from *Cell, 2018*).

4.1.2 Comparison of Model Performance

This section assesses the sensitivity properties of metrics with respect to classifier performance changes to the distribution ratios. Generally, a robust class imbalance index should detect every performance difference changing with respect to distribution ratios. Five popular metrics are used to demonstrate that MPI is more sensitive than others.

Figure 5 reports classifier performance changes with respect to distribution ratios in four experiments. The F1-score, AUC, G-Means, MCC, MK and MPI score are utilized for the evaluation of their performance. Conventionally, the classifier performance deteriorates as the distribution bias increases, which can be represented by traditional evaluation metrics. From the results, two conclusions can be observed.

First, when the capability of those classifiers can be distinguished by traditional metrics, compared with G-Means, MCC, MK, F1-score and AUC, the deterioration of MPI is significant and sensitive. For instance, as shown in **Figure 5.c**, in the detection of 'dogs', MCC has the most significant deterioration (by *69%*), followed by MPI (by *48%*). However, MCC fails to classify classifier performances in some cases. For instance, MCC fails to distinguish classifier performance by changing -0.001 between specific class imbalance statuses (*1:8* and *1:9*). This is not plausible since other metrics show a deterioration in performance in this case, such as AUC (by changing 0.007) and MK (by changing 0.005). However, in this case, the MPI score still decreases by 0.014 with a significant deterioration, which is more plausible and sensitive than MCC.

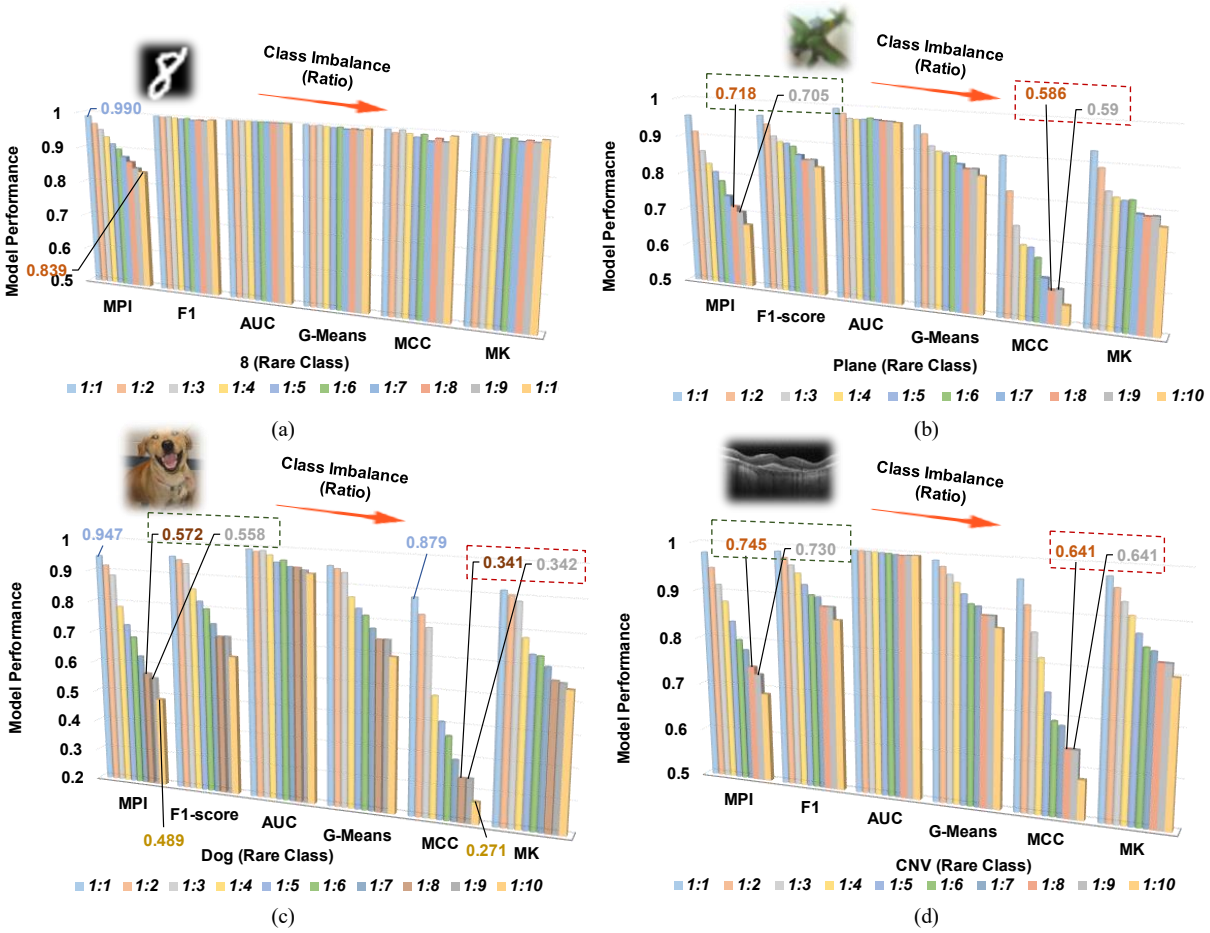


Figure 5 Classifier performance changes with respect to distribution ratios using different datasets. (a) MNIST. (b) CIFAR-10. (c) Cats & Dogs (from *Kaggle Challenge in 2013*). (d) Retinal OCT (from *Cell, 2018*). The MPI is more sensitive than other metrics.

Second, MPI can detect all types of changes in classifier performance, while the other metrics fail to distinguish changes in classifier performance, remaining at the same level. In some special cases, as the distribution bias increases, the change in traditional metrics is very small and even remains constant. In this situation, the *MPI* score can detect tiny changes in classifier performance

in different distribution bias statuses, while the traditional metrics remain at the same level. As shown in **Figure 5.a**, for the detection on the '8' sample, as the distribution bias increases, the G-Means, MCC, MK, F1-score and AUC performances of those classifiers remain similar (approximately 0.98 to 1), while the MPI performance of the classifier decreases by 0.151 (15.3%) from 0.990 to 0.839. In general, it can represent classifier performances under various distribution statuses, while the traditional metrics remain at the same level. In general, the MPI is the most sensitive index, and it can detect every performance difference with respect to distribution ratios.

4.2 Experimental Results on Algorithms

In this section, we examine the properties of the proposed ***Ideal Model Performance Algorithm***. First, we obtain the calculated ***Important Distribution Points*** using ***Algorithm I.*** Then, based on Important Distribution Points, we predict classifier performances using the ***Quadratic Distribution Function***. Finally, we evaluate the predicted ***Ideal Model Performance*** based on the ***Quadratic Distribution Function.***

4.2.1 Important Distribution Points

To obtain the Important Distribution Points, we first obtained the model performance of the initial points (x_0, MPI^0) for the Rare Classes and the Majority Classes. Then, as shown in **Figure 6** and ***Algorithm I***, based on the MPI^0 of the initial point, we used the ***MPI Gap Function*** to calculate other important distribution ratio points in different class imbalance statuses for both classes.

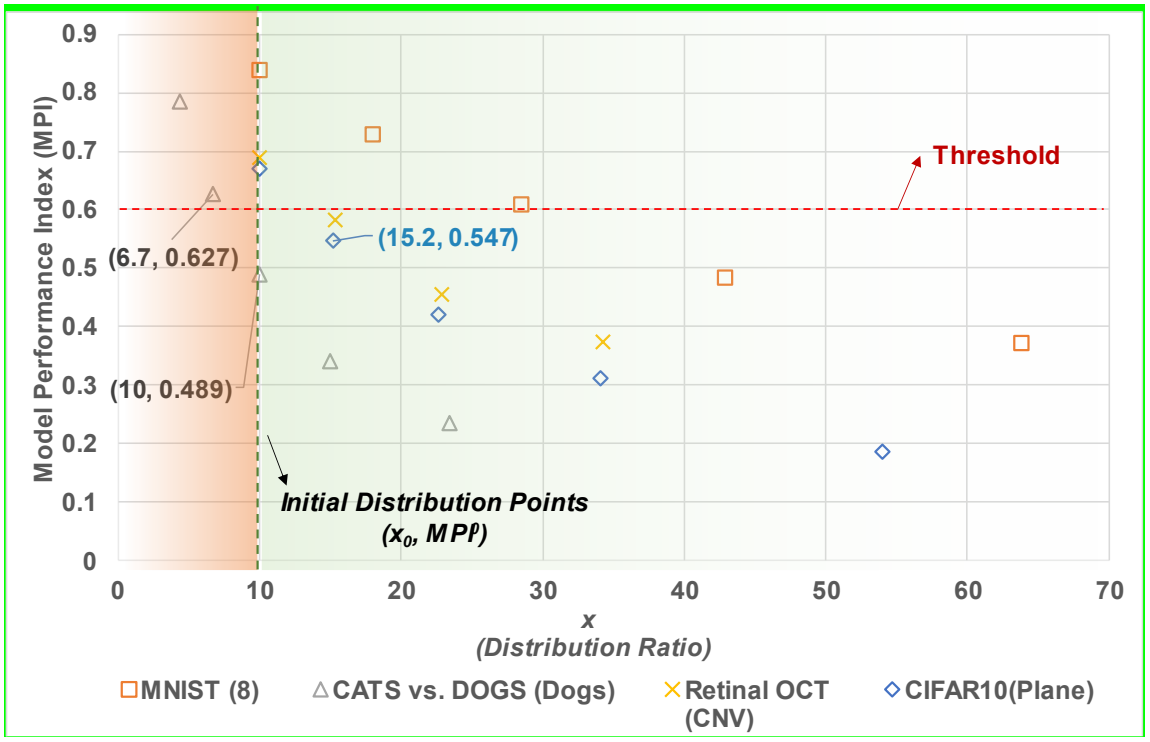


Figure 6 Important distribution points and their MPI performance for the rare class. The points can be set on both sides of the initial points.

Figure 6 reports the important distribution points of the Rare Classes in experiments. These important distribution points can be calculated on both sides of the initial distribution points (x_0, MPI^0). They are highly related to the initial model performance (MPI^0). Physically, a poor MPI^0 has no value at all since the classifier will become even worse as the distribution imbalance increases. Therefore, more samples are required to obtain a value classifier performance for training.

For instance, as shown in **Figure 6**, in Cats vs. Dogs, the second important data distribution point x_1 was calculated as *6.7* when the initial point was set to *10* (x_0=*10*). This is because the

MPI^0 *(MPI^0=0.489)* is poor (lower than 0.6), and more data will be required to train an expectable classifier. In contrast, they can be calculated as calculated as 17.996, 15.211, and 15.341 in MNIST, CIFAR10 and Retinal OCT, respectively. This is because for these experiments, only a small number of samples is required to obtain an expected performance for those classifiers.

4.2.2 Predictions on the Classifier Performance

Next, the classifier performances can be predicted by the ***Quadratic Distribution Function***, which can be used for further estimation in the MPI^{ideal}. Based on those cleared important distribution points, we predicted classifier performances using the ***Quadratic Distribution Function*** by restrictions on RSS, R^2, Adjust-R^2, and RMSE. **Figure 7** reports the predicted classifier performance and assessment results of the Quadratic Distribution Function. The boundary conditions of the Quadratic Distribution Function are shown in **Appendix B**.

In general, the R-square of the fitting curve is more than 99%, and the SSE is less than 0.001. The adjusted R-square is more than 0.97, and the RMSE is less than 0.1. These parameters show the credibility of the obtained Quadratic Distribution Function as well as its outcomes.

4.2.3 Evaluations on the Ideal Model Performance

With the obtained Quadratic Distribution Function, the MPI^{ideal} can be calculated. It achieves a precise prediction on an ideal performance simple dataset, while a few errors occur when testing with a more complex dataset. **Table 2** shows the estimations of the ideal model performance;

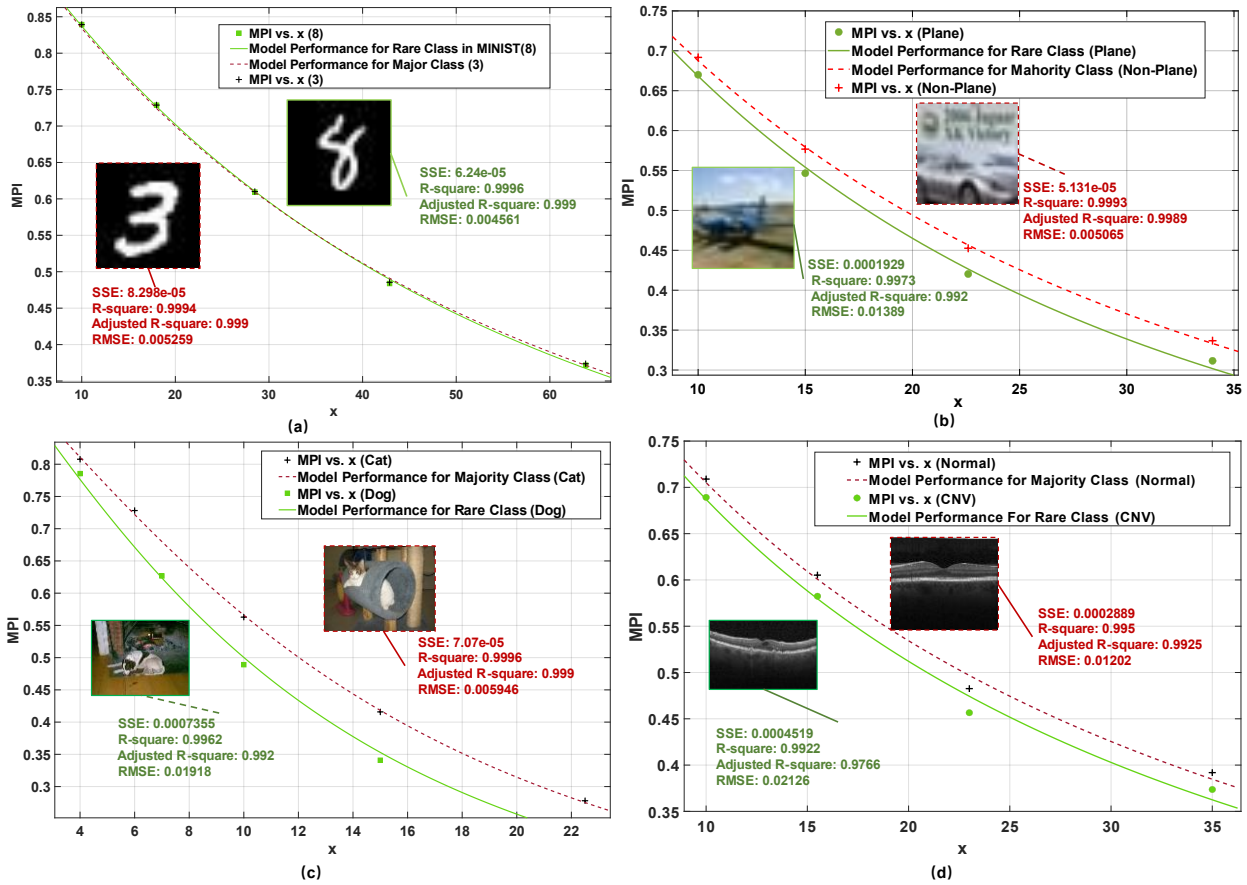


Figure 7 Predicted model performance for both classes in all datasets. (a) MNIST. (b) CIFAR-10. (c) Cats & Dogs (from *Kaggle Challenge* in 2013). (d) Retinal OCT (from *Cell, 2018*).

the Ideal Model Performance Algorithm obtains error rates of 0.060% and *0%* for the Rare Class and the Majority Class in MNIST, respectively. It is shown that this algorithm can obtain the ideal performance precisely in a simple dataset. For the more complex datasets, such as the CIFAR-10, Cats vs. Dogs, and Retinal OCT, there are a few errors in the calculation of the performances. This is mainly caused by the complex features in the data, but the results still show the practical value of the ideal model performance predictions.

Table 2 Errors of the Ideal Model Performance for Rare Classes and Majority Classes for All Datasets

Errors of the Ideal Model Performance for Rare and Majority Classes for All Datasets (R: Rare Class M: Majority Class)

	MNIST		CIFAR-10		CATS vs. DOGS		RETINAL OCT	
	R	M	R	M	R	M	R	M
MPI_{true}	0.990	0.990	0.955	0.954	0.947	0.947	0.979	0.979
MPI_{pre}	0.991	0.990	0.967	0.969	0.951	0.948	0.966	0.969
Errors	(-)0.001	(-)0.000	(-)0.012	(-)0.015	(-)0.004	(-)0.002	0.013	0.01
Error/ MPI^{true} %	0.06%	0	1.26%	1.57%	0.463 %	0.163%	1.3%	1%

5. Conclusion

This paper provides one evaluation measure and one algorithm for the class imbalance problem, which are referred to as the ***Model Performance Index (MPI)*** and the ***Ideal Model Performance Algorithm***. The Model Performance Index is proposed for the evaluation of classifier performance, which considers trained classifiers from imbalanced datasets. It allows us to perform evaluations and comparisons between classifier performances in various situations, especially in different class imbalance statuses. Compared with some traditional metrics, this index is more sensitive than others, while the traditional metrics might remain at the same level with the increment of the distribution imbalance. In addition, the ***Ideal Model Performance Algorithm*** allows us to perform estimations of the classifier performances with a fair distribution using a limited number of samples. In general, these experimental results show the practical value of the measurement of the model performance considering class imbalance.

Although the proposed metrics assess the classifier performance considering class imbalance impacts, some improvements can be done to overcome some limitations in the future. For example, current work focused on Convolutional Neural Network-based classifiers only, and other classifiers are not discussed. However, we believe that these metrics have the potential for other classifiers, such as traditional neural networks and other machine learning classifiers. In addition, the proposed metrics are based on two-class problems, prohibiting their use in a multiclass environment. Therefore, in our future work, we will expand the theoretical analysis and experiments to other classifiers using multiclass datasets.

Acknowledgment

The work is partially supported by grants from the TBRS grant from the Research Grant Council of the Hong Kong Special Administrative Region Government (T42-717/20-R).

References

[1] M. Buda, A. Maki, M.A. Mazurowski, A systematic study of the class imbalance problem in convolutional neural networks, Neural Netw. 106 (2018) 249-259.

[2] S.S. Mullick, S. Datta, S.G. Dhekane, S. Das, 2020. Appropriateness of performance indices for imbalanced data classification: An analysis. Pattern Recognition. 102, 107197.

[3] Z. Wang, Q.D. Dong, W. Guo, D.D. Li, J. Zhang, W.L. Du, 2022. Geometric imbalanced deep learning with feature scaling and boundary sample mining. Pattern Recognition. 126, 108564.

[4] Z. Yang, W.H. Tang, A. Shintemirov, Q.H. Wu, Association Rule Mining-Based Dissolved Gas Analysis for Fault Diagnosis of Power Transformers, Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews. 39 (2009) 597-610.

[5] Z.B. Zhu, Z.H. Song, Fault diagnosis based on imbalance modified kernel Fisher discriminant analysis, Chemical Engineering Research & Design. 88 (2010) 936-951.

[6] L.X. Cui, L. Bai, Y.C. Wang, X. Jin, E.R, 2021. Hancock, Internet financing credit risk evaluation using multiple structural interacting elastic net feature selection. Pattern Recognition. 114, 107835.

[7] M.A. Mazurowski, P.A. Habas, J.M. Zurada, J.Y. Lo, J.A. Baker, G.D. Tourassi, Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance, Neural Netw. 21 (2008) 427-436.

[8] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 42 (2011) 463-484.

[9] S. Piri, D. Delen, T.M. Liu, H.M. Zolbanin, A data analytics approach to building a clinical decision support system for diabetic retinopathy: Developing and deploying a model ensemble, Decision Support Systems. 101 (2017) 12-27.

[10] L.C. Zhao, Z.W. Shang, J. Tan, M.L. Zhou, M. Zhang, D.G. Gu, T.P. Zhang, Y.Y. Tang, 2022. Siamese networks with an online reweighted example for imbalanced data learning. Pattern Recognition. 132, 108947.

[11] N. Japkowicz, S. Stephen, The class imbalance problem: A systematic study[1], Intelligent Data Analysis. 6 (2002) 429-449.

[12] S. Suh, P. Lukowicz, Y.O. Lee, 2022. Discriminative feature generation for classification of imbalanced data. Pattern Recognition. 122, 108302.

[13] N.V. Chawla, N. Japkowicz, A. Kotcz, Editorial: special issue on learning from imbalanced data sets, ACM SIGKDD explorations newsletter. 6 (2004) 1-6.

[14] Y. Lu, Y.M. Cheung, Y.Y. Tang, Bayes Imbalance Impact Index: A Measure of Class Imbalanced Data Set for Classification Problem, IEEE Trans Neural Netw Learn Syst. 31 (2020) 3525-3539.

[15] Y.X. Liu, Y. Liu, B.X.B. Yu, S.H. Zhong, Z.J. Hu, 2023. Noise-robust oversampling for imbalanced data classification. Pattern Recognition. 133, 109008.

[16] S.M. Abd Elrahman, A. Abraham, A review of class imbalance problem, Journal of Network and Innovative Computing. 1 (2013) 332-340.

[17] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, Handling imbalanced datasets: A review, GESTS international transactions on computer science and engineering. 30 (2006) 25-36.

[18] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, A. Hussain, Comparing Oversampling Techniques to Handle the Class Imbalance Problem: A Customer Churn Prediction Case Study, Ieee Access. 4 (2016) 7940-7957.

[19] K.H. Brodersen, C.S. Ong, K.E. Stephan, J.M. Buhmann, The Balanced Accuracy and Its Posterior Distribution, in: 2010 20th International Conference on Pattern Recognition, IEEE, Istanbul, Turkey, 2010, pp. 3121-3124.

[20] V. García, R.A. Mollineda, J.S. Sánchez, Index of Balanced Accuracy: A Performance Measure for Skewed Class Distributions, in: Pattern Recognition and Image Analysis, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 441-448.

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

[22] S. Daskalaki, I. Kopanas, N. Avouris, Evaluation of classifiers for an uneven class distribution problem, Applied Artificial Intelligence. 20 (2006) 381-417.

[23] P. Branco, L. Torgo, R.P. Ribeiro, Relevance-Based Evaluation Metrics for Multi-class Imbalanced Domains, in: Advances in Knowledge Discovery and Data Mining, Springer International Publishing, Cham, 2017, pp. 698-710.

[24] T. Kautz, B.M. Eskofier, C.F. Pasluosta, Generic performance measure for multiclass-classifiers, Pattern Recognition. 68 (2017) 111-125.

[25] X. Guo, Y. Yin, C. Dong, G. Yang, G. Zhou, On the class imbalance problem, in: 2008 Fourth international conference on natural computation, IEEE, 2008, pp. 192-201.

[26] J. Cohen, A Coefficient of Agreement for Nominal Scales, Educational and psychological measurement. 20 (1960) 37-46.

[27] K. Krippendorff, Estimating the Reliability, Systematic Error and Random Error of Interval Data, Educational and psychological measurement. 30 (1970) 61-70.

[28] K. Krippendorff, Content analysis : an introduction to its methodology, Fourth edition. ed., SAGE, Los Angeles, 2019.

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

[30] S. Boughorbel, F. Jarray, M. El-Anbari, 2017. Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. Plos One. 12, e0177678.

[31] D. Chicco, G. Jurman, 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics. 21, 6.

[32] A. Estabrooks, N. Japkowicz, A Mixture-of-Experts Framework for Learning from Imbalanced Data Sets, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[33] L.A. Jeni, J.F. Cohn, F. De La Torre, Facing imbalanced data--recommendations for the use of performance metrics, in: 2013 Humaine association conference on affective computing and intelligent interaction, IEEE, 2013, pp. 245-251.

[34] A.P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, Pattern Recognition. 30 (1997) 1145-1159.

[35] M. Kubat, S. Matwin, Addressing the curse of imbalanced training sets: one-sided selection, in: Proceedings of the 14th International Conference on Machine Learning, Morgan Kaufmann, Los Altos, CA, 1997, pp. 179–186.

[36] J.W. Grzymala-Busse, L.K. Goodwin, X. Zhang, Increasing sensitivity of preterm birth by changing rule strengths, Pattern Recognition Letters. 24 (2003) 903-910.

[37] [dataset] Y. LeCun , L. Bottou , Y. Bengio , P. Haffner , et al. , Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324 .

[38] [dataset] A. Krizhevsky , G. Hinton , et al. , Learning multiple layers of features from tiny images, Technical Report, Citeseer, 2009 .

[39] [dataset] Kaggle, Dog Vs. Cat Competition. http://www.kaggle.com/c/dogs-vs-cats, 2014.

[40] [dataset] Kermany, Daniel; Zhang, Kang; Goldbaum, Michael (2018), “Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification”, Mendeley Data, V2, doi: 10.17632/rscbjbr9sj.2.

Supplementary

Appendix A

In **Appendix A**, we provide the algorithms for the calculation of the important data distribution in **Algorithm I.**

Algorithm I: Important Distribution Points

Input: Distribution Ratio between the Majority and Rare classes x_0 *in first class imbalance status*.

The Initial Class Imbalance Status (x_0, MPI^0), the Classifier Performance Difference G_i (G_i=0.1,0.2,0.3,0.4).

If $0.6 \leqslant MPI^0 \leqslant 1$, *# If the initial classifier performance is good*

$MPI^i = MPI^0 - G_i$, $(0 < i \leqslant 4)$ *# less data required for training*

Else *# If the initial classifier performance is poor*

$MPI^i = MPI^0 \pm G_i$, $(0 < i \leqslant 4)$ *# more data required for training*

Output: Important Distribution Points $\{x_i\}$.

 # Distribution Ratio between the Majority and Rare classes x_i in important class imbalance status

1. ***For*** $i \leftarrow 1$, ***do*:**
2. Calculate the t_i of the Important Data Distribution Ratio:

   # Ratio between two class imbalance statuses x_i and x_0, when the classifier performance differs by G_i)

   $t_i = f(b) \leftarrow a = g(b) \leftarrow (x_0, MPI^0)$; $x_i = t_i x_0$;

   $t_i = m(a, b)$

   If $0.6 \leqslant MPI^0 \leqslant 1$,

   #less samples required for training for next class imbalance status

   #When the classifier performance differs by G_i

   $$f(t_i) = \frac{1}{a * x_0 + b} - \frac{1}{a * x_i + b} - G_i$$

   Else

   # more samples required for training for next class imbalance status

   #When the classifier performance differs by G_i

   $$f(t_i) = \frac{1}{a * x_0 + b} - \frac{1}{a * x_i + b} \pm G_i$$

3. Calculate the t_{imin} of the Important Data Distribution Ratio:

---

4. # *The minimum ratio between two class imbalance statuses x_i and x_0*
5. $t_i = \frac{b-c_1^i}{b-c_2^i}$ ($c_j^i \in$ R, $0< t_i$) $\leftarrow t_i = f(b)$
6. ***t_{imin}=min(t_i)***
7.
8. ***If*** Rare Class, ***then***:
9. $t_i = \frac{b-c_1^i}{b-c_2^i}$ ($0<t_i$, initial range: b $\in [0.99, +\infty)$, $c_j^i \in$ R)
10. If $b< c_1^i \cap b< c_2^i$, then:
11. *S=min* (c_1^i, c_2^i)*;0.99<b<S*
12. $t_{imin} \leftarrow t_i = \frac{b-c_1^i}{b-c_2^i}$
13. Else
14. *Continue*
15. ***Else if*** *Majority Class*, ***then****:*
16. $t_i = \frac{b-c_1^i}{b-c_2^i}$ ($0<t_i$, initial range: b $\in [0.99,1.48)$, $c_j^i \in$ R)
17. ***If*** $b< c_1^i \cap b< c_2^i$, ***then***:
18. *S=min* (c_1^i, c_2^i)*; 0.99<b<S*
19. $t_{imin} \leftarrow t_i = \frac{b-c_1^i}{b-c_2^i}$
20. ***Else***
21. ***Continue***
22. End ***for***
23. *Output x_i. # Based on x_i, calculate x_i by t_{imin} ($x_i = t_{imin} x_0$)*
24. #If *$1<t_{imin}$*, fewer samples are required for next class imbalance status
25. #If *$0<t_{imin}<1$*, more samples are required for next class imbalance status
26. *#Output* the nearest class imbalance status when differs by G_i from the initial performance *MPI^0*

where a) *(x_0, MPI^0)* refers to the initial class imbalance points (*x_0 =10* in this work). Physically, *x_0* represents the distribution ratio between the majority class and the rare class in the first class imbalance status, and $\boldsymbol{x_i}$ represents the distribution ratio in other class imbalance statuses. b) G_i refers to the change in *MPI (⊿MPI^i)*. Physically, it represents the difference in MPI performances between models in two class imbalance statuses. c) t_i refers to a set of positive ratios between two **Important Distribution Points** x_0 and x_i. Physically, to obtain the next important class imbalance point, *t>1* shows that fewer samples are required for training, and 0<t<1 shows that more samples are required for training. t_{imin} is the minimum value of t_i. d) *a* and *b* refer to coefficients of x_1 and x_i. e) *m(x)* refers to the relationships between *t* and two parameters *a, b*. f) *g(x)* refers to the relationships between *'a'* and *'b'*. g) *f(x)* refers to the

relationships between *'t'* and *'b'*. h) c^i_j refers to a real number from the reorganization of function *t=f(b)*, obtained from calculation processes. i) S refers to the minimum value in a set of c^i_j

In this work, our implementation currently generates three to four nearest important distribution points. Physically, their model performance is lower or higher than the initial performance $\boldsymbol{MPI^0}$ by a constant difference defined by G_i. To obtain these points, when the classifier performance differs G_i from the initial point, we calculate the ratio (t_i) between i^{th} class imbalance statuses (x_i) and the initial class imbalance status (x_0), where x_i refers to the distribution ratio between the rare classes and the majority classes. Physically, we find the nearest class imbalance status (x_i, MPI^{ith}) that differs by G_i from the initial imbalance status (x_0, MPI^0).

Appendix B

In **Appendix B,** we list the boundary conditions of Quadratic Distribution Function for the Rare Classes in **Table B. 1.**

Table B. 1 The Boundary Conditions of the Quadratic Distribution Function for the Rare Classes in Experiments.

Dataset	(X_0, MPI^0)	Conditions		
		$\boldsymbol{MPI^0}$	***Threshold***	**a, b**
MNIST (8)	**x**	**10.000**	⩾0.6	-0.0099⩽a, 0.99⩽b<1.1916
	MPI	**0.839**		
CIFAR-10 (Plane)	**x**	**10.000**	⩾0.6	-0.0099⩽a, 0.99⩽b<1.48
	MPI	**0.670**		
CATS vs. DOGS (Dogs)	**x**	**10.000**	<0.6	-0.0099⩽a, 0.99⩽b<2.04
	MPI	**0.489**		
Retinal OCT (CNV)	**x**	**10.000**	>0.6	-0.0099⩽a, 0.99⩽b<1.45
	MPI	**0.689**		

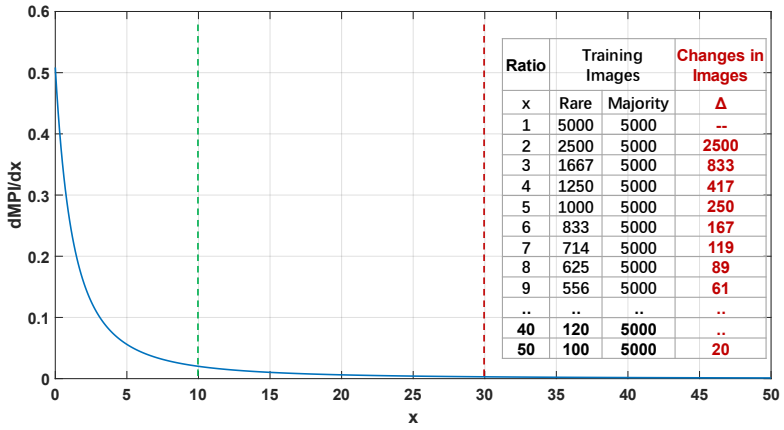
Here, (x_0, MPI^0) refers to the First Important Distribution Point; a and b refer to the parameters in Basic Distribution Functions.

Appendix C Derivative of the Model Performance Index (MPI)

In this section, the sensitivity and effectiveness of the MPI are discussed. Mathematically, the derivative of a function of a real variable is considered to measure the sensitivity when changing the outputs with respect to a change in its inputs. Therefore, to assess the sensitivity and effectiveness of the proposed metrics, the first directive of MPI has been calculated as:

$$\left|\frac{dMPI}{dx}\right| = \left|-\frac{a}{(a*x+b)^2}\right| \tag{19}$$

(19) measures how much the *MPI* changes with respect to a change in its distribution ratios *x*. For instance, assuming that the classifier obtains an *MPI* score of 0.9 with a fair distribution (***1:1***), as shown in **Figure C.1**, the sensitivity and effectivity of the *MPI* change with increasing distribution ratio.



Ratio	Training Images		Changes in Images
x	Rare	Majority	Δ
1	5000	5000	--
2	2500	5000	2500
3	1667	5000	833
4	1250	5000	417
5	1000	5000	250
6	833	5000	167
7	714	5000	119
8	625	5000	89
9	556	5000	61
..	..	..	..
40	**120**	**5000**	..
50	**100**	**5000**	**20**

Figure C.1 Changes in MPI performance with respect to changes in the distribution ratio, the first derivative of MPI performance.

With increasing distribution ratio x, its sensitivity and effectiveness decrease. It decreases quickly at the beginning but subsequently decreases slowly, from approximately 0.5 to 0. It is more sensitive and efficient before 1:10, but its sensitivity decreases to close to 0 after 1:30. This is plausible since at the very beginning, the number of training samples is highly influenced by distribution ratios, leading to a significant impact on the classifier performance. However, when the training set has a large distribution ratio, the changes in training samples obtain a small value, leading to a small impact on classifier performance.