INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 19, Issue: 6, Month: December, Year: 2024
Article Number: 6702, https://doi.org/10.15837/ijccc.2024.6.6702

CCC Publications

AGORA
UNIVERSITY PRESS

Split Difference Weighting: An Enhanced Decision Tree Approach for Imbalanced Classification

T. Zhou, X. Gao, X. Sun, L. Han

Tingting Zhou
School of Economics and Management
University of Science and Technology Beijing, China
30 Xueyuan Road, Haidian Distict, Beijing 100083, China
zhoutingting929@163.com

Xuedong Gao*
School of Economics and Management
University of Science and Technology Beijing, China
30 Xueyuan Road, Haidian Distict, Beijing 100083, China
*Corresponding author: gaoxuedong@manage.ustb.edu.cn

Xi Sun
Collaborative Innovation Center of Steel Technology
University of Science and Technology Beijing, China
30 Xueyuan Road, Haidian Distict, Beijing 100083, China
sxfzddd@126.com

Lei Han
School of Economics and Management
University of Science and Technology Beijing, China
30 Xueyuan Road, Haidian Distict, Beijing 100083, China
d202210486@xs.ustb.edu.cn

Abstract

Imbalanced data classification remains a significant challenge in machine learning, particularly in decision tree algorithms where majority class features are often overshadowed. This study introduces a novel split index based on class key decision factor (CKD factor) to address this issue. We propose two new algorithms: Split Difference Decision Tree (SDDT) and Weighted Split Difference Classification and Regression Tree (WSD-CART). These algorithms enhance feature expression for majority classes during node splitting, thereby improving classification performance on imbalanced datasets. Experiments conducted on five UCI datasets with varying imbalance levels demonstrate the effectiveness of our approach. The WSD-CART algorithm consistently outperformed traditional methods, showing significant improvements in F-score, AUC, precision, recall, and accuracy, particularly for majority classes. In a real-world application to space product material classification, our method increased the true positive rate for majority class identification from 66.32% to 76.17%, while maintaining high overall accuracy. This study contributes to the field of imbalanced learning by providing a new perspective on decision tree split criteria. The proposed methods offer both improved classification performance and interpretable decision rules, making them valuable for various domains dealing with imbalanced data.

Keywords: imbalanced classification, class key decision factor, split difference decision tree, weighted split difference classification and regression tree.

1 Introduction

Classification problems are a fundamental aspect of machine learning, aiming to categorize data points into predefined classes based on their attributes. The accuracy and reliability of classification algorithms are crucial as they directly impact decision-making processes [1]. One significant challenge in classification tasks is handling imbalanced data, where some classes (majority classes) have far fewer samples than others (majority classes) [2]. This imbalance can severely affect the performance of standard classification algorithms, leading to poor accuracy in identifying majority class instances, which are often the most critical to detect [3].

In recent years, imbalanced data classification has been widely studied in fields such as medical analysis [4], fault diagnosis[5] [6], fraud detection [7], and network intrusion detection [8]. Among these, ensemble algorithms, as a popular topic in machine learning, are often chosen as effective methods to address imbalanced data classification [9]. The decision tree (DT) has become the preferred base classifiers in ensemble learning due to several advantages [10]. They are inherently interpretable, providing clear and explicit classification rules, which allows for easy understanding and communication of the model decision process, crucial in practical applications [11]. Moreover, decision trees are adept at modeling complex decision boundaries. Their hierarchical structure recursively splits data based on feature values, making them sensitive to data sample perturbations and enabling the capture of intricate patterns within the data. Traditional DT classifiers, such as ID3, CART, and C4.5, use class sample probabilities as the split index. The performance of a tree can be influenced by its split index. While these methods are effective in distinguishing different class samples in balanced classifications, they may lead to reduced accuracy in identifying majority classes in imbalanced classifications, as the features of the majority classes can be overshadowed by those of the majority classes [12].

To address this, various methods have been proposed to tune the split index of DT nodes. Existing methods such as the DKM split index [13], designed to enhance DT induction, primarily favor the majority class and do not effectively mitigate class imbalance. Similarly, the Hellinger distance, proposed as a distribution divergence measure [14], offers some skew-insensitivity but still lacks comprehensive handling of class distribution disparities. To bridge this gap, the article [15] introduces α-divergence, a novel scalar parameterized split index that focuses on generating diversified base classifiers through variable splitting (diversification) within DTs. Despite its demonstrated benefits in handling imbalanced data and enhancing diversity in ensemble learning contexts, α-divergence poses challenges in terms of interpretability. Contrary to DKM, majority entropy [16] is a purity measure that specifically evaluates partitioning of majority class samples. It improves the inductive capabilities of DTs by reducing the number of majority samples outside the range of the majority class. It has been shown to outperform the aforementioned indexes in various classification evaluation, but it may overlook overall class balance and could potentially lead to overfitting in certain scenarios.

These methods mainly focus on node purity without considering class distribution differences between child nodes. This can result in nodes reflecting majority class characteristics, necessitating deeper splits. CCPDT has modified the traditional confidence measure to focus on the sample size within each class, rather than the sample size within each child node [17]. Research proposes a new split index that allows one side of the split to generate highly homogeneous rules[18]. Besides, Lv et al. [19] proposed the concept of a CKD factor that integrates class dispersion and class decision degree, achieving a fair representation of majority class characteristics in the class label determination of leaf nodes. Inspired by the perspectives, this article shifts the focus of building DT node splitting criteria for imbalanced classification to simultaneously evaluate the class disparity between the split child nodes and the homogeneity within the nodes.

Despite these advancements, a significant gap remains in achieving a balance between node purity and class distribution differences between child nodes. Existing methods often focus on one aspect while neglecting the other, leading to DTs that may still be biased towards majority classes. Therefore, a more comprehensive approach is required to enhance the representation of majority classes without compromising overall classification accuracy.

This article aims to improve the node split index of DTs for accurate identification of majority classes in imbalanced classification. The main contributions are as follows:

- A Split Difference Decision Tree (SDDT) is constructed. Based on the concept of the CKD factor, the algorithm proposes a new split difference index, achieving fair representation of features from majority classes in the classification of imbalanced samples. Compared to other methods, this index comprehensively evaluates both intra-node and inter-node dispersion differences in classes, as well as inheritance differences, rather than just class sample distribution and dispersion.

- Weighted Split Difference Classification and Regression Tree (WSD-CART) is constructed. By weighting the split difference with the Gini index, the algorithm combines the purity measure with split difference characteristics to construct a CART classification algorithm weighted by split difference, ensuring fair evaluation of all classes in imbalanced datasets.

The rest of the article is organized as follows. Section 2 presents the preliminaries related to this research. Section 3 presents the theory and method of the split index, including two algorithms. Section 4 conducts an experiment on the public UCI datasets and an application for space product material classification. The article is concluded in Section 5.

2 Research foundation

Section 2.1 takes the Gini index, the classic node split index of the CART for binary classification as an example, pointing out its advantages and shortcomings in the DT classification process, especially the defects in the imbalanced classification (the same applies to other classic DT classification algorithms). Section 2.2 describes related concepts of CKD factor proposed by [19] and clarifies its significance in determining leaf node class labels in imbalanced classification, laying the theoretical groundwork for the next section to construct new node split index based on the concept of CKD factor.

2.1 Analysis of traditional decision tree split indexes

Considering sample distribution of the DT leaf nodes presented in Table 1 as an illustrative case, assume that a given DT non-leaf node $R_{j=0}$, under the condition that attribute $A_k = a$, the sample set $D_{j=0}$ on node $R_{j=0}$ is divided into two parts, $D_{j=1}$ (left node $R_{j=1}$) and $D_{j=2}$ (right node $R_{j=2}$), where $D_{j=1} = (p_1, n_1)$, $D_{j=2} = (p_2, n_2)$, $D_{j=0} = (p, n)|p = p_1 + p_2$, $n = n_1 + n_2$, with p representing the number of positive samples P, and n representing the number of negative samples N.

Table 1: Example of sample distribution of leaf nodes

Parent node	Leaf node	C_0	C_1	Total
$R_{j=0}(p,n)$	$R_{j=1}(p_1,n_1)$	$p_1 = 800$	$n_1 = 20$	$p_1 + n_1 = 820$
	$R_{j=2}(p_2,n_2)$	$p_2 = 200$	$n_2 = 180$	$p_2 + n_2 = 380$
Total		$p_1 + p_2 = 1000$	$n_1 + n_2 = 200$	

For the binary classification problem above, the Gini index of the probability distribution at the non-leaf node $R_{j=0}$ could be transformed into:

$$Gini(D_{j=0}) = 2pn/(p+n)^2 \tag{1}$$

Under the condition that attribute $A_k = a$, the conditional Gini index could be transformed into:

$$Gini(D_{j=0}, A_k = a) = \frac{p_1 n_1}{p_1 + n_1} \cdot \frac{2}{p_1 + n_1} + \frac{p_2 n_2}{p_2 + n_2} \cdot \frac{2}{p_2 + n_2} \tag{2}$$

$$Gini(D_{j=0}, A_k = a) = \frac{2}{p+n} \left(\frac{p_1 n_1}{p_1 + n_1} + \frac{p_2 n_2}{p_2 + n_2} \right) \tag{3}$$

$Gini(D_{j=0})$ represents the uncertainty of the set $D_{j=0}$, and $Gini(D_{j=0}, A_k = a)$ represents the uncertainty of the set$D_{j=0}$ after being split by $A_k = a$. The smaller the $Gini(D_{j=0}, A_k = a)$, the higher the purity of the samples in the child nodes after the split.

Based on formulas (2) and (3), analyze the composition and advantages of the Gini index:

(1) It can evaluate the dispersion of the parent node sample in child nodes. In formula (2), the proportion of the sample of the split left and right nodes to the parent node sample size are $\frac{p_1+n_1}{p+n}$, $\frac{p_2+n_2}{p+n}$, that are evaluations regarding the probability distribution of samples. If $p + n$ is constant, to make the Gini index as small as possible, the difference between $(p_1 + n_1)$ and $(p_2 + n_2)$ should be as large as possible;

(2) It can evaluate the comprehensive ability of different classes to make majority decisions in each child node. In formula (2), the products of the probabilities of the majority class in the samples of the left and right split nodes are given by $\frac{p_1 n_1}{(p_1+n_1)^2}$ and $\frac{p_2 n_2}{(p_2+n_2)^2}$, respectively. If $(p_1 + n_1)$ and $(p_2 + n_2)$ are constant, to make the Gini index smaller, the larger the difference between p_1 and n_1 and the larger the difference between p_2 and n_2 are needed.

The aforementioned analysis holds true for balanced datasets but falters in imbalanced classification scenarios, where the inherent disparity in the number of positive and negative samples precludes accurate recognition of the majority class with such classification preference.

Similar to prevalent DT algorithms, including ID3, gain ratio, and most improvements [20] [21] [22] to split index, the Gini index lacks the capability to assess class differences between left and right sub-nodes post-splitting. In formula (3), $p + n$ is constant, a smaller Gini index requires a larger difference between p_1 and n_1, as well as between p_2 and n_2. However, even if $p_1 > n_1$ and $p_2 > n_2$, both maintaining substantial differences, the resulting left and right child nodes may exhibit no discernible class differences. Especially in imbalanced classification, this defect will be more obvious.

As shown in Table 1, based on the Gini index, the parent node $R_{j=0}(1000 : 200)$, is partitioned into two leaf nodes: $R_{j=1}(800 : 20)$ and $R_{j=2}(200 : 180)$. Here C_1 and C_0 signify the two classes in the binary classification problem, with C_1 being the minority class. During the tree's construction, the Gini index tends to select the majority class (C_0) as the splitting point in order to maximize classification performance. However, this predisposition can result in overfitting the generated DT towards class C_0, ultimately diminishing its capability to identify samples belonging to class C_1. Consequently, as evident from Table 1, both child nodes ($R_{j=1}$ and $R_{j=2}$) might inadvertently become dominated by class C_0, due to conditions $800 > 20$, $200 > 180$.

In summary, the logic of node splitting by the Gini index (including the majority of DT split indexes) can be generalized as dividing by the mainstream, not by the class feature [23]. Such idea easily leads to the split child nodes being dominated by the majority class, causing the classification features of the majority class to be unrecognized. At the same time, it is easy for the class features of two child nodes to be indistinguishable after splitting, forcing the classification model to explore deeper layers, thus reducing classification efficiency.

Therefore, this article attempts to propose a new DT node split index to address the issue that the Gini index does not consider the inability of small sample classes to vote due to low sample probability in imbalanced classifications.

2.2 Concepts related to leaf node class key decision factor

In reviewing the outcomes presented in Table 1, it is evident that the class label of the leaf nodes is assigned according to the "majority rule", whereby the class with the greater number of samples within a node is selected—hence, node $R_{j=2}$ is categorized as class C_0. However, a closer inspection reveals that an overwhelming 90% of class C_1 samples are actually allocated to this same leaf node $R_{j=2}$. Given that C_1 represents the minority class, its numerical inferiority impedes the fair representation of its class characteristics, thereby contradicting the fundamental goal of classification.

The purpose of classification is to derive scientific classification rules through equitable treatment and exploration of the distinct features from different classes, and to reflect objectively existing patterns, rather than basing decisions simply on sample probabilities. Therefore, to address the issue of determining leaf node class labels in the classification of imbalanced samples, the CKD factor has been proposed [19]. Below are the relevant definitions pertaining to this index.

Assuming there is a leaf node R_j, with a set of sample classes $C = \{C_i \,|\, i = 0, 1, 2, 3, \dots, I\}$, where $|C_i|$ denotes the number of samples belonging to class C_i, and C_i^j represents the set of samples belonging to class C_i in leaf node R_j, $|C_i^j|$ denotes the number of samples in C_i^j.

Definition 1. *Leaf Node Class Dispersion* α_{ij}

Represents the degree of dispersion of class C_i in leaf node i.e., the proportion of the number of C_i class samples in node R_j to the total number of C_i class samples in the entire set.

$$\alpha_{ij} = \frac{|C_i^j|}{|C_i|} \tag{4}$$

Definition 2. *Leaf Node Class Decision Degree* β_{ij}

Represents the authoritative strength of class C_i samples in leaf node R_j, that is, the proportion of the number of C_i class samples in node R_j to the total number of samples in node R_j.

$$\beta_{ij} = \frac{|C_i^j|}{\sum_{i=1}^{n} |C_i|} \tag{5}$$

Based on formulas (4) and (5) , it can be understood that both α_{ij} and β_{ij} have a value range of [0,1]. The larger the value of α_{ij}, the more concentrated the characteristics of that class are reflected in the node. Similarly, the larger the value of β_{ij}, the stronger the majority voting power of that class within the node. Therefore, combining the two leads to the following Definition 3.

Definition 3. *Leaf Node CDK Factor* d_{ij}

The product of the leaf node class dispersion and the class decision degree.

$$d_{ij} = \alpha_{ij}\beta_{ij} = \frac{|C_i^j|}{|C_i|} \cdot \frac{|C_i^j|}{\sum_{i=1}^{n} |C_i^j|} \tag{6}$$

The leaf node CDK factor is used to comprehensively measure the performance strength of each class in the leaf node, $d_{ij} \in [0, 1]$.

The Gini index adheres to the principle of majority voting, and both it and d_{ij} consider β_{ij}, while the introduction of α_{ij} in d_{ij} incorporates the inheritance ratio of the class samples in the leaf nodes from the total class samples into the determination of the leaf node class labels. This reflects the membership degree of the class samples to all attribute features of the leaf node after being progressively divided layer by layer. Although the minority class is at a disadvantage in terms of sample quantity (i.e., β_{ij} is smaller), if it achieves a higher α_{ij} value compared to the majority class, it can still attain the same comprehensive expression strength of node features as the majority class. Therefore, the CKD factor is more suitable for imbalanced classification than the Gini index.

The larger the d_{ij}, the greater the likelihood that class C_i will be the classification label in node R_j. When categorizing a leaf node based on its CDK factor, the class with the highest d_{ij} is chosen as the classification label, not the one with the largest sample size. Hence, a leaf node class feature recognition mode L_j based on CDK factor is defined, as illustrated in formula (7).

$$L_j = C_i \quad (\text{where } d_{ij} = \max(d_{Ij}),\, i \in \{1, 2, \dots, I\}) \tag{7}$$

Calculate α_{ij}, β_{ij}, d_{ij}, and L_j for each class of the leaf nodes in Table 1, as shown in Table 2.

- Comparative analysis of leaf node $R_{j=1}$ class label determination: If based on the traditional DT algorithm, the leaf node $R_{j=1}$ would be labeled as class C_0 due to the larger proportion of C_0 class samples. However, if based on the leaf node CDK factor, according to formula (7), the CDK factor value of C_0 is greater than that of C_1, leading to the same determination that the class label of $R_{j=1}$ is C_0.

- Comparative analysis of leaf node $R_{j=2}$ class label determination: If based on the traditional DT algorithm, the leaf node $R_{j=2}$ would be labeled as class C_0 due to the larger proportion of C_0 class samples. However, if based on the leaf node CDK factor, the class label of $R_{j=2}$ class label should be C_1. The two rules yield inconsistent results because the CKD factor, after integrating the α_{ij}, allows the majority class C_1, which inherits the sample probability from the same class in the root node, to fully exert its influence in the identification of the leaf node class label.

Table 2: Class feature recognition results of leaf node for Table 1

Node	Class	α_{ij}	β_{ij}	d_{ij}	L_j
$R_{j=1}$	$C_{(i=0)}$ $C_{(i=1)}$	0.800 0.100	0.976 0.024	0.781 0.002	$L_{(j=1)} = C_0$
$R_{j=2}$	$C_{(i=0)}$ $C_{(i=1)}$	0.200 0.900	0.526 0.474	0.105 0.462	$L_{(j=2)} = C_1$

In summary, the class decision degree has a similar capability to evaluate the sample class probability as authority, expressing the authoritativeness of the sample class. The class dispersion degree α_{ij} calculates the inheritance probability of the leaf node sample class from the root node sample class, eliminating the impact of uneven sample distribution on the leaf node class determination. Compared with the Gini index, the application of the CKD factor index, which integrates α_{ij} and β_{ij}, in the identification of leaf node classes, can better compensate for the deficiencies in the classification voting of majority class samples in imbalanced datasets.

3 Theory and Method of Split Difference

To address the issue of improving accuracy in imbalanced classification problems, this section proposes a splitting criterion of DT nonleaf nodes named the Split Difference Index γ, based on the research foundation. The construction mechanism of γ is clarified in Subsection 3.1. A Split Difference Decision Tree algorithm (SDDT) is constructed in Subsection 3.2 and a Weighted Split Difference Classification and Regression Tree algorithm (WSD_CART) is constructed in Subsection 3.3. These methods are designed to enhance the overall classification performance of the sample by improving the feature representation ability of majority classes. Additionally, the computational complexity of the algorithms is analyzed separately following the algorithm steps.

3.1 CKD factor oriented to split processes of non leaf node

Based on analyzing the role of CKD factor in leaf node class label determination and the shortcomings of traditional split indexes, the feasibility of proposing a non leaf node split index for DTs based on CKD factor is studied.

Unlike calculating CKD factor under a determined sample distribution of leaf node, the splitting of non leaf nodes is a dynamic exploration of optimal splitting points. Therefore, the following series of concepts are redefined.

Definition 4. *Class Dispersion Degree of Subnode α^s_{ij} during Splitting*

Represents the dispersion degree of class C_i in subnode R_j, i.e., the proportion of the number of C_i class samples in node R_j to the number of C_i class samples in parent node R_{j-1}.

$$\alpha^s_{ij} = \frac{|C^j_i|}{|C^{j-1}_i|} \tag{8}$$

Definition 4 differs from Definition 1. For imbalanced classification, Definition 1 takes a global perspective and analyzes the proportion of samples of each class in the leaf nodes that are inherited from the samples of each class in the total sample set. In contrast, Definition 4 views the parent node as the root node of a single splitting process and analyzes the proportion of samples of each class in child nodes that are inherited from those of the parent node.

Definition 5. *Class Decision Degree of Subnode β^s_{ij} during Splitting*

Represents the authority of class C_i samples in subnode R_j, i.e., the proportion of C_i class samples in node R_j relative to the total number of samples in node R_j. This definition is consistent with the content of Definition 2.

$$\beta^s_{ij} = \frac{|C^j_i|}{\sum_{i=1}^{n}|C^j_i|} \tag{9}$$

According to formulas (8) and (9), $\alpha^s_{ij} \in [0, 1]$, $\beta^s_{ij} \in [0, 1]$. A larger value of α^s_{ij} indicates that the characteristics of the class are more concentrated in that child node, while a larger β^s_{ij} indicates a stronger majority voting power of the class in the child node. Similarly, α^s_{ij} and β^s_{ij} are used to construct the CKD factor of the child node during the splitting process.

Definition 6. *CKD Factor of Subnode d^s_{ij} during Splitting*

The product of α^s_{ij} and β^s_{ij}, to measure the strength of feature representation of each class in the child nodes generated by the node split.

$$d^s_{ij} = \alpha^s_{ij}\beta^s_{ij} = \frac{|C^j_i|}{|C^{j-1}_i|} \cdot \frac{|C^j_i|}{\sum_{i=1}^{n}|C^j_i|} \tag{10}$$

According to formula (10), $d^s_{ij} \in [0, 1]$. The larger the d^s_{ij}, the greater the likelihood that class C_i will be the classification label in child node R_j. When $d^s_{ij} = 1$, it means that all samples of class C_i from the parent node are distributed in child node R_j, and all samples in child node R_j belong to class C_i.

Let the CKD factor of the positive samples p_1 in child node R_1 be:

$$d^s_{p_1R_1} = \alpha^s_{p_1R_1}\beta^s_{p_1R_1} = \frac{{p_1}^2}{(p_1 + n_1)(p_1 + p_2)} \tag{11}$$

Similarly, the CKD factor of the negative samples (n_1) in child node R_1, and the CKD factor of the positive (p_2) and negative samples (n_2) in child node R_2 are as follows:

$$d^s_{n_1R_1} = \alpha^s_{n_1R_1}\beta^s_{n_1R_1} = \frac{{n_1}^2}{(p_1 + n_1)(n_1 + n_2)} \tag{12}$$

$$d^s_{p_2R_2} = \alpha^s_{p_2R_2}\beta^s_{p_2R_2} = \frac{{p_2}^2}{(p_2 + n_2)(p_1 + p_2)} \tag{13}$$

$$d^s_{n_2R_2} = \alpha^s_{n_2R_2}\beta^s_{n_2R_2} = \frac{{n_2}^2}{(p_2 + n_2)(n_1 + n_2)} \tag{14}$$

3.2 CKD factor oriented to split processes of non leaf node

Figure 1 illustrates the conceptual framework for constructing a novel split index, based on the CKD factor. This process necessitates the fusion of the CKD factor indexes of child nodes during the split to achieve fair expression of node class features. Two requirements must be fulfilled: First, in deriving the difference in the CKD factor within a node, it is necessary to maximize CKD factor difference between classes, aiming to emphasizing the main class feature within the node; Second, in deriving the difference of inter-node CKD factor, it is necessary to make the split left and right nodes tend to express different class features, achieving better classifying samples through once node division. Based on these two processes, the relevant Definition 7 (to meet the first requirement) and Definition 8 (to meet the second requirement) are proposed.

Definition 7. *Intra-node CKD Factor Difference $d^s_{R_j}$*

Used to quantitatively measure the difference in CKD factor within a child node, to identify the main class features of the child node, as shown in formula (15). $d^s_{R_j} \in [-1, 1]$, and the larger the absolute value of $d^s_{R_j}$, the greater the difference in CKD factor between the two class samples within the node.

$$d^s_{R_j} = d^s_{p_i R_j} - d^s_{n_i R_j} \tag{15}$$

$$L^s_{R_j} = \begin{cases} P, & \text{if } d^s_{R_j} \geq 0 \\ N, & \text{if } d^s_{R_j} < 0 \end{cases} \tag{16}$$

The non-leaf node class feature identification pattern $L^s_{R_j}$ is a qualitative expression of the intra-node CKD factor difference, as shown in formula (16). If $d^s_{R_j} \geq 0$, the main class feature of the node is the class P; if $d^s_{R_j} < 0$, the main class feature of the node is the class N. Therefore, when determining the class of non-leaf nodes based on the intra-node CKD factor difference during the splitting process, the class with the greater $d^s_{R_j}$ is selected as the class feature $L^s_{R_j}$ of the non-leaf node.

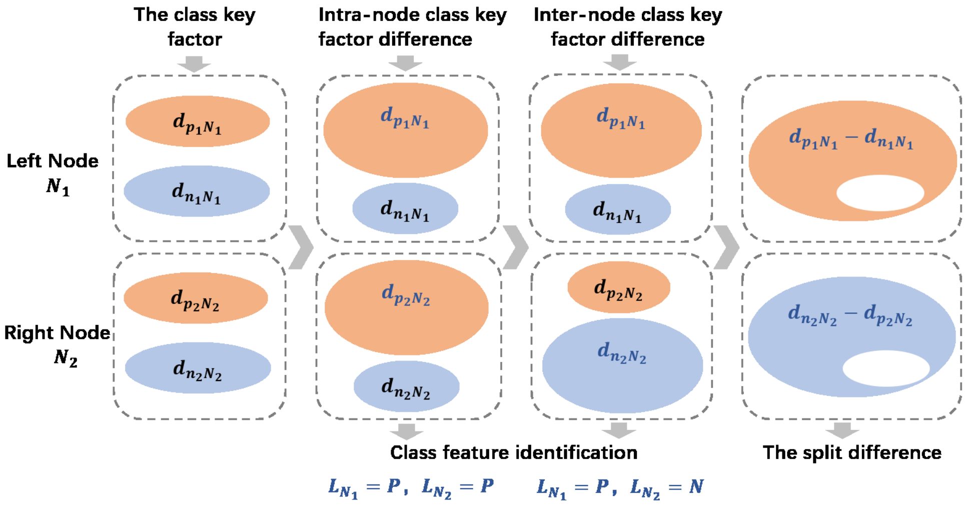

The class key factor
Intra-node class key factor difference
Inter-node class key factor difference
Left Node N_1
Right Node N_2
$d_{p_1N_1}$
$d_{n_1N_1}$
$d_{p_2N_2}$
$d_{n_2N_2}$
$d_{p_1N_1} - d_{n_1N_1}$
$d_{n_2N_2} - d_{p_2N_2}$
Class feature identification
The split difference
$L_{N_1} = P$, $L_{N_2} = P$
$L_{N_1} = P$, $L_{N_2} = N$

Figure 1: The construction framework for the split difference index

Definition 8. *Inter-node CKD Factor Difference* $d^s_{(R_j, R_{(j+1)})}$

Used to determine the class feature difference between child nodes R_j and R_{j+1}, as shown in formula (17).

$$d^s_{(R_j, R_{j+1})} = \begin{cases} 0, & \text{if } L^s_{R_j} = L^s_{R_{j+1}} \\ 1, & \text{if } L^s_{R_j} \neq L^s_{R_{j+1}} \end{cases} \tag{17}$$

The inter-node CKD factor difference $d^s_{(R_j, R_{j+1})}$ is a Boolean judgment formula, $d^s_{(R_j, R_{(j+1)})} \in \{0, 1\}$. If $d^s_{(R_j, R_{j+1})} = 0$, it indicates that the class features of the child nodes are the same; if $d^s_{(R_j, R_{j+1})} = 1$, it indicates that the class features of the two child nodes are different. To ensure that the class features of the child nodes generated during the node splitting process are as distinct as possible, the inter-node CKD factor difference tends to be 0.

Referring to Figure 1, through the two-step derivation of the intra-node and inter-node CKD differences in the non-leaf node class feature expression strategy, the qualitative expression of the class features of the child nodes is achieved. To quantitatively measure the expression of class features of the child nodes, the $d^s_{R_j}$ and $d^s_{(R_j, R_{j+1})}$ are combined to construct a new split index of DT non-leaf nodes, i.e., the split difference index, to achieve fair classification of minority class in the process of classifying imbalanced samples.

Definition 9. *Split Difference* $\gamma^s_{(R_j, R_{j+1})}$

The index is used to comprehensively evaluate the CKD factor difference within and between child nodes during the node splitting process. The calculation formulas are shown in (18) and (19). $\gamma^s_{(R_j,R_{j+1})} \in [0,1]$. The larger the value of $\gamma^s_{(R_j,R_{j+1})}$, the more distinct the representation of different classes in the left and right nodes.

$$\gamma^s_{(R_j,R_{j+1})} = \frac{1}{2}|d^s_{R_j} - d^s_{R_{j+1}}| \tag{18}$$

$$\gamma^s_{(R_j,R_{j+1})} = \frac{1}{2}|d^s_{p_iR_j} + d^s_{n_{i+1}R_{j+1}} - d^s_{n_iR_j} - d^s_{p_{i+1}R_{j+1}}| \tag{19}$$

Extreme value analysis of split diversity $\gamma^s_{(R_j,R_{j+1})}$:

1. If $\gamma^s_{(R_j,R_{j+1})} = 1$, it indicates that the main class features of the child nodes are pure class P and pure class N, which is an optimal node splitting situation;
2. If $\gamma^s_{(R_j,R_{j+1})} = 0$, it indicates that the main class features of the child nodes are both class P or class N, and $d^s_{R_j} = d^s_{R_{j+1}}$, which is a poor situation.

In the formula (19), $d^s_{p_iR_j}$, $d^s_{n_{i+1}R_{j+1}}$, $d^s_{n_iR_j}$, and $d^s_{p_{i+1}R_{j+1}}$ respectively represent the "distribution probability" after eliminating the sample difference between minority class and majority class within the child nodes. This criterion integrates two measurements:

1. By calculating the proportion of category samples in the child nodes inherited from the parent node category samples, it can eliminate the influence of the overall sample probability of that category on the probability of categories inherited by the child nodes;
2. By calculating the proportion of category samples in the child nodes from the sample proportion in this node, it conducts a majority vote.

$d^s_{p_iR_j} - d^s_{n_iR_j}$ and $d^s_{n_{i+1}R_{j+1}} - d^s_{p_{i+1}R_{j+1}}$ reflect the class differences of the same node, while $d^s_{p_iR_j} - d^s_{n_iR_j} - (d^s_{n_iR_j} - d^s_{p_{i+1}R_{j+1}})$ reflects the class differences between nodes. The absolute value is taken to ignore the order of difference making, so that $\gamma^s_{(R_j,R_{j+1})}$ can treat two child nodes and two classes more fairly.

Hence, a DT node split index that eliminates the interference of large and small class sample distribution differences while fully distinguishing the differences in child node classes has been constructed.

By combining the leaf node class feature pattern recognition method based on CKD factor for imbalanced classification in Section 3.1 with the split difference index proposed in this section, a DT node split algorithm based on split difference is constructed. This algorithm divides the construction of imbalanced classification DT into two parts: the determination of class labels for leaf nodes and the determination of splitting points for non-leaf nodes. The specific algorithm steps are as follows.

According to formula (3), $\text{Gini}(D_0, A_i = a) = \frac{2}{p+n}\left(\frac{p_1n_1}{p_1+n_1} + \frac{p_2n_2}{p_2+n_2}\right)$. After calculating p_1, n_1, p_2, and n_2 at step 7 of Algorithm 1, the computational complexity of $p + n = p_1 + n_1 + p_2 + n_2$ is $O(1)$, the computational complexity of $\frac{p_1n_1}{p_1+n_1}$ and $\frac{p_2n_2}{p_2+n_2}$ are $O(1)$, therefore, the computational complexity of the Gini index is still $O(1)$.

The logic of Algorithm 1 is similar to that of traditional DT. However, instead of using traditional indices like the Gini index for node splitting, it replaces them with γ. According to formula (19), $\gamma^s_{(R_j,R_{j+1})} = \frac{1}{2}\left|\frac{p_1^2}{(p_1+n_1)(p_1+p_2)} + \frac{n_2^2}{(p_2+n_2)(n_1+n_2)} - \frac{n_1^2}{(p_1+n_1)(n_1+n_2)} - \frac{p_2^2}{(p_2+n_2)(p_1+p_2)}\right|$, the basic operations are still performed by p_1, n_1, p_2, and n_2, and the computational complexity of the algorithm is still $O(1)$, which is the same as that of the CART algorithm.

The splitting criterion for Algorithm 1 is the splitting difference, $\gamma^s_{(R_j,R_{j+1})}$. Using $d^s_{p_1R_1}$ as an example, $\gamma^s_{(R_j,R_{j+1})}$ eliminates the disadvantage of a small class proportion p_1 in $p + n$ by measuring the inheritance ratio $p_1/(p_1 + p_2)$ from the parent node category sample. This ensures that $\gamma^s_{(R_j,R_{j+1})}$ fairly evaluates both minority and majority classes when assessing child node splits. Unlike the Gini index, which emphasizes purity measurement, Algorithm 1 focuses more on expressing differences between classes among split child nodes and within child nodes themselves. It tends to make child nodes exhibit two different classes, thereby enabling faster and more accurate identification of minority class samples.

Algorithm 1 SDDT: Split Difference Decision Tree

```
Require: $D_{\text{train}}$, $D_{\text{test}}$
Ensure: $DT$, $tree$
 1: Initialize the root node $R_{(j=0)}$, Sample space $D_j = D_{\text{train}}$, $p_{(j=0)}$, $n_{(j=0)}$;
 2: Initialize the best split parameters of $R_j$: $v_j = (\gamma_{\text{best}}, A, a, D^l_{\text{best}}, D^r_{\text{best}}) = (0,0,0,0,0)$;
 3: if $R_j$ satisfies the $DT$ stop condition then
 4:     Identify the class feature $L_j$ of $R_j$ using Formula (4);
 5: else
 6:     for $i \in I$, $k \in K$ do
 7:         Obtain $D^l_{ik}$, $D^r_{ik}$, $p^l_{ik}$, $n^l_{ik}$, $p^r_{ik}$, and $n^r_{ik}$ by splitting $D_j$ when $A_i = a^k_i$;
 8:         Calculate $d^{p^l}_{ik}$, $d^{n^l}_{ik}$, $d^{p^r}_{ik}$, and $d^{n^r}_{ik}$ using Formulas (11), (12), (13), (14);
 9:         Calculate $\gamma(D^l_{ik}, D^r_{ik})$ using Formula (18);
10:         if $\gamma(D^l_{ik}, D^r_{ik}) > \gamma_{\text{best}}$ then
11:             $v_j(\gamma_{\text{best}}, A, a, D^l_{\text{best}}, D^r_{\text{best}}) = (\gamma(D^l_{ik}, D^r_{ik}), A_i, a^k_i, D^l_{ik}, D^r_{ik})$;
12:         end if
13:     end for
14:     $D_j = D^l_{\text{best}}$; repeat steps 2-14, traverse down the $DT$ child nodes;
15:     $D_j = D^r_{\text{best}}$; repeat steps 2-15, traverse down the $DT$ child nodes;
16: end if
17: Generate the $DT$, $tree$.
```

3.3 Weighted Split Difference Classification and Regression Tree

The classic DT split indexes evaluate the node splitting process based on the distribution probability of class samples of child nodes. Such evaluation philosophy ensures the concentrated expression of the probability of the majority class in the child nodes, such that the smaller the Gini index, the higher the purity of the child node samples. However, for imbalanced data, minority class samples cannot be equally recognized due to their naturally low sample size.

The proposal of the split difference index γ can well compensate for the shortcomings of classical split indexes in expressing features of minority class. Therefore, integrating Gini index and γ index during the node splitting process to generate a weighted split index can better ensure the ideal degree of node splitting, i.e., both ensuring the purity of the child node samples and solving the feature expression of minority class samples.

Generating a weighted split index through the setting of a global weight parameter is an exploratory process. To obtain the best classification results, different weights need to be adapted for variable sample sizes, feature sizes, and levels of sample imbalance. We know that the smaller the Gini index, the higher the sample purity and the greater the split difference, the greater the class difference of the child nodes. Therefore, the weights ω and $1 - \omega$ are set to adjust the decision-making degree of the Gini index and the γ index during the node splitting, $\omega \in [0, 1]$.

Assuming $\omega = 0$, SG solely relies on the Gini index, indicating that node splitting only concerns purity measurement. As ω gradually increases from 0, node splitting begins to consider the impact of split diversity gradually. And when $\omega < 0.5$, the Gini index exerts a stronger influence on SG compared to γ. Conversely, when $\omega > 0.5$, the influence of γ on SG becomes greater than that of the Gini index, directing node splitting to focus more on split difference, while the emphasis on purity measurement weakens. When $\omega = 1$, SG relies entirely on γ, and node splitting exclusively considers split difference. However, the optimal split's concern for purity and split difference varies with variable sample distributions. Therefore, ω needs to be a global variable. By traversing all possible ranges of ω to obtain the best SG value, the best node splitting is achieved through a comprehensive consideration of both purity and split difference. Therefore, the calculation process of the weighted split difference and Gini index (SG) is shown in Formula (20).

$$SG(N_j, D_j, A_i = a^k_i, \omega) = \omega \cdot \gamma + (1 - \omega) \cdot (1 - \text{Gini}) \tag{20}$$

In formula (20), $Gini \in [0, 1]$, $\gamma \in [0, 1]$, therefore $SG \in [0, 1]$. Theoretically, the larger SG is when

Gini is smaller and γ is larger. Meanwhile, the higher the internal sample purity of the two nodes after the split, the greater the intra-node class difference, the greater the inter-node class difference, and the better the node split results. Therefore, it is considered that when the weighted sum of γ and Gini index takes the maximum value $\max(SG)$, the corresponding weight is the optimal weight ω^{best}. Select the best split parameter setting $SG(N_j, A_i = a_i^k, \omega^{\text{best}})$ to split child nodes, and traverse downwards to generate the entire DT. The CART classification algorithm weighted by split difference is presented as follows.

Algorithm 2 WSD_CART: Weighted Split Difference Classification and Regression Tree

```
Require: $D_{\text{train}}$, $D_{\text{test}}$, weight parameter $\omega \in [0,1]$
Ensure: $Eva_{\text{test}}^{\omega_{\text{best}}} = (F_{\text{test}}^{\omega_{\text{best}}}, AUC_{\text{test}}^{\omega_{\text{best}}}, PRE_{\text{test}}^{\omega_{\text{best}}}, REC_{\text{test}}^{\omega_{\text{best}}}, ACC_{\text{test}}^{\omega_{\text{best}}})$, $tree_{\text{best}}$, $\omega_{\text{best}}$
 1: Initialize the root node $R_{j=0}$, Sample space $D_j = D_{\text{train}}$, $p_{j=0}$, $n_{j=0}$, $\omega_{\text{best}} = 0$, $tree_{\text{best}} = \text{null}$;
 2: for $\omega \in [0,1]$ do
 3:     Initialize the best split parameters of $R_j$, $v_j$: $(\gamma_{\text{best}}, A, a, D_{\text{best}}^l, D_{\text{best}}^r) = (0,0,0,0,0)$;
 4:     Initialize $Eva_{\text{test}}^{\omega_{\text{best}}} = (0,0,0,0,0)$, $SG_{\text{best}} = 0$;
 5:     if $R_j$ satisfies the $DT$ stop condition then
 6:         Identify the class feature $L_j$ of $R_j$ using Formula (4);
 7:     else
 8:         for $i \in I$, $k \in K$ do
 9:             Obtain $D_{ik}^l, D_{ik}^r, p_{ik}^l, n_{ik}^l, p_{ik}^r, n_{ik}^r$ by splitting $D_j$ when $A_i = a_i^k$;
10:             Calculate $\gamma(D_{ik}^l, D_{ik}^r)$ using Formula (18);
11:             Calculate $SG(N_j, D_j, A_i = a_i^k, \omega)$ using Formula (20);
12:             if $SG(N_j, D_j, A_i = a_i^k, \omega) > SG_{\text{best}}$ then;
13:                 $v_j(\text{SG}, A, a, D^l, D^r) = (SG(N_j, D_j, A_i = a_i^k, \omega), A_i, a_i^k, D_{ik}^l, D_{ik}^r)$;
14:                 $\omega_{\text{best}} = \omega$;
15:             end if
16:         end for
17:         $D_j = D_{ik}^{l_{\text{best}}}$; repeat steps 3-16, traverse down the $DT$ child nodes;
18:         $D_j = D_{ik}^{r_{\text{best}}}$; repeat steps 3-17, traverse down the $DT$ child nodes;
19:     end if
20:     Generate the $DT$ under $\omega_{\text{best}}$, $tree_{\text{best}}$;
21:     Calculate $Eva_{\text{test}}^{\omega_{\text{best}}} = (F_{\text{test}}^{\omega}, AUC_{\text{test}}^{\omega}, PRE_{\text{test}}^{\omega}, REC_{\text{test}}^{\omega}, ACC_{\text{test}}^{\omega})$;
22: end for
23: Output $Eva_{\text{test}}^{\omega_{\text{best}}}$ under $\omega_{\text{best}}$, $tree_{\text{best}}$;
```

The inner computational logic of Algorithm 2 is similar to that of traditional DT. Instead of using traditional criteria like the Gini index for node splitting, it replaces them with SG, so the inner computational complexity of Algorithm 2 is the same as that of the CART algorithm. Additionally, an outer loop is added to traverse weight parameters ω. By traversing the parameter space, multiple corresponding DTs can be generated. The classification results of these DTs are compared, and the tree with the optimal evaluation is selected as the output of the algorithm. The corresponding value of ω represents the optimal weight ratio of the Gini index and γ. The SG index, which combines the advantages of both evaluation indexes, achieves better classification performance than its components. However, the computational complexity of this algorithm is linearly related to the size of the ω parameter space, meaning that the number of candidate DTs generated corresponds to the number of parameter space traversals. Ultimately, one optimal result is selected. Therefore, Algorithm 2 has a higher computational complexity compared to traditional DTs.

4 Experimental result

4.1 Dataset and experimental design

The binary classification experiment was conducted using the UCI dataset and the space product material dataset (SPM). These UCI datasets with varying levels of imbalance are usually difficult to accurately classify by traditional DT algorithms. Therefore, these datasets with different sample size and attribute size were selected to validate the effectiveness of proposed algorithms in imbalanced classification. Additionally, to execute the CART algorithm, attributes of the selected dataset are all numerical. Table 3 describes the basic characteristics of the experimental datasets at different imbalanced levels, including sample size, positive/negative sample ratio, and number of attributes.

The SPM dataset comprises space product materials obtained from the logistics center of the China Academy of Launch Vehicle Technology. It contains a total of 921 samples, with a positive and negative sample ratio (Sample distribution) of 1:3.77 for generic and non-generic material categories. The SPM dataset is utilized for the subsequent application of classification of space product materials. Additionally, the YS dataset selected "NUC" as positive samples and "ME2" class as negative samples. Furthermore, after removing missing values, 829 data points remained in the MM dataset.

Table 3: Information for the dataset

Dataset	Source	Abbreviation	Data size	Sample distribution	Attributes
Yeast	UCI	YS	480	1:8.4	9
Customer Churn	UCI	CC	3150	1:5.36	13
SPM	Survey	SPM	921	1:3.77	10
Breastw	UCI	BW	683	1:1.86	9
Mammographic Mass	UCI	MM	829	1:1.07	6
Raisin	UCI	RS	900	1:1	7

Four DT split indexes were selected for experimental results comparison. The details of these split indexes are presented in Table 4. The term "Split tendency" represents the tendency of the optimal value of different split indexes. The split index $Gini + d_{ij}$ is a compared index proposed by Lv et al. [19], while γ and SG are new split indexes proposed in Section 3.2 and Section 3.3, respectively. For the weight parameters setting of SG, $\omega \in [0, 1]$ and the step length is 0.1. A value of 0.1 was selected to observe the trend of the impact of the two indexes on classification performance in the WSD-CART algorithm. (If better classification results are desired, the parameter traversal step length can be set to be smaller, such as 0.01.) By traversing the weight parameters, the optimal evaluation result of SG will be determined, which is then used for comparison with those of other indexes.

Table 4: Split index for each DT algorithm

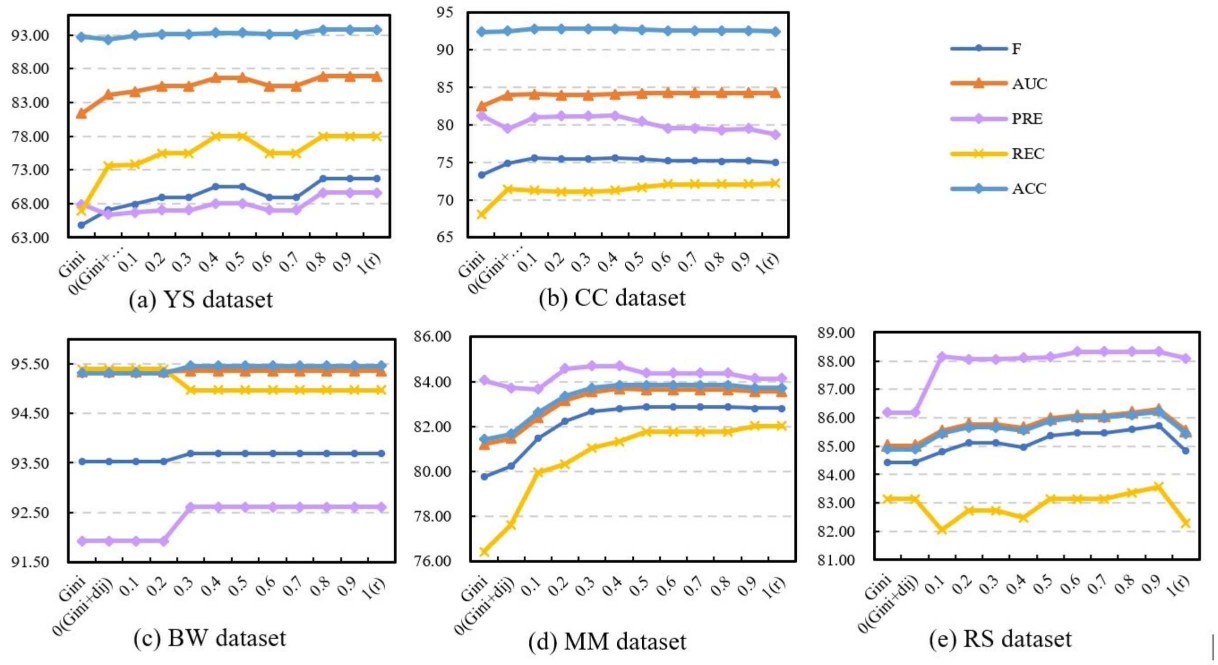
Split index	Gini	Gini+d_{ij}	γ	SG
Algorithm	CART	KF_CART	SDDT	WSD_CART
Actor	Compared	Compared	Proposed	Proposed
Split tendency	MIN	MIN	MAX	MAX

The experiment employs the average of ten-fold cross validation results as the algorithm outcome, and assesses the classification proficiency of γ and SG listed in Table 4 using five extensively utilized classic classification evaluation metrics: F-score, AUC, REC (recall), PRE (precision), and ACC (accuracy) [24]. This study conducts two sets of experiments: (1) to validate the performance of the proposed DT algorithms, namely SDDT, using UCI datasets, and further demonstrate the superiority of the WSD-CART algorithm; (2) to address the accurate identification challenge of generic materials in the space product material classification domain, the WSD-CART algorithm is applied to the SPM dataset to obtain pertinent decision support.

4.2 Results Analysis

Figure 2 illustrates the classification performance evaluation of various split indexes across different datasets with varying levels of imbalance. The horizontal axis scale represents the Gini index, while the remaining 11 scales depict the traversal of different ω values for the SG index. Specifically, when $\omega = 0$ and $\omega = 1$, the SG index precisely corresponds to Gini+d_{ij} and γ, respectively.

It is evident that, with the exception of the REC line in Figure 2(e), the performance evaluation of each dataset exhibits an overall upward trend as the ω of γ increases. This indicates that, relative to the compared algorithms, the SDDT algorithm and WSD-CART algorithm possess strong



(a) YS dataset

(b) CC dataset

(c) BW dataset

(d) MM dataset

(e) RS dataset

Figure 2: Comparison of classification performance of algorithms on UCI dataset

classification capabilities on both balanced and imbalanced data. Furthermore, the WSD-CART algorithm demonstrates robust stability and robustness across different datasets, consistently maintaining its classification performance over multiple runs and exhibiting high classification capabilities when handling various levels of dataset imbalance.

To obtain the optimal weight parameters of the SG index for the classification of each dataset in Figure 2, we identify a set of optimal classification evaluations as the final outcomes of the SG index by jointly screening for the maximum values of F and AUC. The weight corresponding to this set of evaluations is deemed the optimal weight ω_{best}, reflecting the most favorable impact ratio of the Gini index and γ on the optimal classification of the dataset. For instance, the optimal evaluation result for YS is determined by the SG index within a weight interval [0.8,1], where the values of F, AUC, PRE, REC, and ACC all attain their peak. Since the DT structures for the SG index division on the weight interval [0.8,1] are consistent, all resulting evaluations are identical and can be considered optimal. Analogously, distinct optimal evaluation weights are identified for various datasets: 0.4 for CC, a range of [0.3,1] for BW, [0.5,0.8] for MM, and 0.9 for RS. Each of these weights underscores the dataset-specific optimization achieved through the SG index framework. Overall, compared to a single value, the weight interval reflects that the SG index is more applicable and stable on these datasets.

These SG evaluation results are contrasted and summarized with the classification evaluation results of the other three indexes in Table 5. For each dataset, the best evaluation results are denoted in bold, and the second-best evaluation results are underlined.

From Table 5, it is discernible that the WSD_CART algorithm, which employs the SG index, exhibits a significantly superior classification effect compared to other algorithms. The γ index of the SDDT algorithm shows the best evaluation on the YS and BW datasets and a second-best evaluation on the MM and RS datasets. The PRE values of Gini+d_{ij} for the datasets are not greater than those of the Gini index, while the REC values of Gini+d_{ij} are not less than those of the Gini index. This indicates that the CKD factor index (d_{ij}) has enhanced the correct prediction ability of the "class 1" samples as the minority class in the leaf nodes when dealing with imbalanced classification, but it cannot avert the possibility of negative samples being misjudged as positive. In comparison to Gini+d_{ij}, the overall classification evaluation of γ is favorable, but the PRE and ACC values in CC and the REC value in RS are slightly lower than those of Gini+d_{ij}. This is attributable to the differing construction goals between γ and the Gini index. However, the classification evaluation of SG, obtained by fusing the characteristics of γ with the Gini index, surpasses those of its constituent indexes.

The ω_{best} column in Table 5 represents the optimal weights for the SG index derived from the classification traversal on each dataset, with the optimal weights for CC and RS being 0.4 and 0.9, respectively, that reflects the contribution of γ to the SG index. The optimal weight setting for datasets YS, BW, and MM demonstrate the more prominent applicability of the split difference γ in the SG index on these datasets.

Table 5: Comparison of final classification results evaluation of algorithms (%)

Dataset	Split index	F	AUC	PRE	REC	ACC	ω_{best}
YS	Gini	64.84	81.40	67.89	66.96	92.71	0.8-1
	Gini+d_{ij}	67.02	84.14	66.39	73.63	92.29	
	γ	**71.74**	**86.91**	**69.61**	**77.98**	**93.75**	
	SG	**71.74**	**86.91**	**69.61**	**77.98**	**93.75**	
CC	Gini	73.35	82.50	81.19	68.00	92.38	0.4
	Gini+d_{ij}	74.85	84.00	79.50	71.40	92.54	
	γ	75.00	84.08	78.71	72.23	92.48	
	SG	**75.62**	**84.09**	**81.20**	**72.26**	**92.83**	
BW	Gini	93.52	95.34	91.92	**95.39**	95.31	0.3-1
	Gini+d_{ij}	93.52	95.34	91.92	**95.39**	95.31	
	γ	**93.69**	**95.36**	**92.62**	94.97	**95.46**	
	SG	**93.69**	**95.36**	**92.62**	94.97	**95.46**	
MM	Gini	79.75	81.22	84.06	76.43	81.42	0.5-0.8
	Gini+d_{ij}	80.23	81.47	83.72	77.62	81.66	
	γ	82.81	83.56	84.13	**82.03**	83.71	
	SG	**82.87**	**83.65**	**84.38**	81.76	**83.83**	
RS	Gini	84.42	85.02	86.18	83.15	84.89	0.9
	Gini+d_{ij}	84.42	85.02	86.18	83.15	84.89	
	γ	84.85	85.53	88.09	82.29	85.44	
	SG	**85.72**	**86.29**	**88.33**	**83.57**	**86.22**	

Overall, the SDDT algorithm has a better classification effect than the classic CART algorithm, and the WSD-CART algorithm has a greater advantage over the SDDT algorithm in improving the accuracy of classification for imbalanced data. SDDT and WSD-CART algorithms also suffer from a common issue in DT algorithms: samples in difficult-to-classify regions can affect the classification performance, as internal and external differences of nodes cannot be easily distinguished. This is reflected in the evaluation of the REC value in the RS dataset and BW dataset. Such limitation will be further studied in our future work.

4.3 Application for space product material classification

Facing high-intensity and normalized concurrent development tasks involving multiple space products, the stable supply of manufacturing materials for space products poses challenges. Generic materials, essential for production across multiple space products with frequent demand, constitute half of the total material supply. Their availability significantly affects the smooth progress of production. Therefore, accurately identifying generic materials from all material is crucial for ensuring the success of development tasks [25].

This study selected SPM dataset to conduct space product material classification experiments and provided corresponding decision support for inventory management of space product materials. The dataset covers the inventory data of the entire process from ordering to delivery of space product materials from 2015-01-01 to 2018-05-01, totaling 310000 items.

According to the definition of the generic materials, attribute meanings, and missing values, data preprocessing is carried out, including data cleaning, classification key attribute statistics, feature selection, and data transformation [26]. A total of 15 classification related attributes are preliminarily obtained; Further calculate the correlation coefficient between attributes and classification categories, and select 10 attributes with correlation coefficients greater than 0 as the classification attribute set, as shown in Table 6. Specifically, the actual values of material types are steel rods, aluminum rods, steel plates, etc., and have been numerically processed. Finally, after data preprocessing, 921 classification sample data were formed with the material ID as the primary key, with a positive/ negative sample ratio of 1:3.77, which is an imbalanced sample.

Table 6: Classification attribute set of space product materials

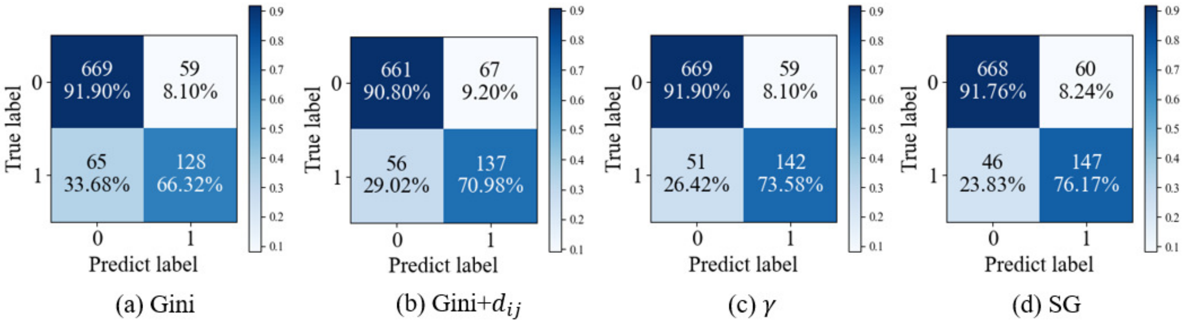
Material ID	Attribute	Correlation coefficient	Example	Type
A_1	Material Type	0.53	7	Numerical
A_2	Kinds of engineering used	0.47	2	Numerical
A_3	Total Outbound frequency	0.38	1	Numerical
A_4	Average of order lead time	0.27	71	Numerical
A_5	Total order times	0.15	3	Numerical
A_6	Variance of monthly Outbound frequency	0.12	1.88	Numerical
A_7	Variance of order lead time	0.11	25	Numerical
A_8	Average of inventory	0.014	1224	Numerical
A_9	Total outbound quantity	0.012	345	Numerical
A_10	Total amount of outbound	0.0064	28080.3	Numerical

Four types of node split indexes were used to classify material, and the results were evaluated as shown in Table 7. The optimal weight range of the SG index was [0.7, 0.9], and its classification results showed the best performance, with a significant improvement in overall classification performance compared to the Gini index and Gini+d_{ij}.

Table 7: Evaluation of classification results for space product materials

Split index	F	AUC	PRE	REC	ACC	ω_{best}
Gini	67.30	79.27	66.58	68.90	86.54	0.7-0.9
Gini+d_{ij}	69.12	81.25	67.56	71.67	86.65	
γ	72.22	82.94	70.73	74.02	88.05	
SG	73.47	84.19	70.88	76.66	88.49	
SG's improvement to Gini	9.17	6.21	6.46	11.26	2.25	

Classify the material samples and obtain the confusion matrix results under four indexes: Gini index, Gini+d_{ij}, γ, and SG, as well as the prediction accuracy of categories as shown in (a), (b), (c), and (d) of Figure 3, respectively. There are a total of 193 positive samples actually, and the SG index predicts 19 more true positive samples compared to the Gini index. It can be clearly seen that the true positive rate (TP) of SG has improved from 66.32% to 76.17%. Such an improvement of close to 10% is very significant for predicting the minority class. At the same time, the SG index also reflects predictive ability for TN and FP that is not weaker than the Gini coefficient, making it more likely to accurately predict the generic material as the majority class of space product materials.



(a) Gini (b) Gini+d_{ij} (c) γ (d) SG

Figure 3: Confusion matrices comparison for space product material classification

Based on the SG index and with weights of $[0.7, 0.9]$, a classification decision tree (DT) is constructed for material data to generate corresponding space product material classification rules, as shown in Figure 4. Among them, there are 8 rules for determining the types of generic materials. By dividing the positive and negative samples on the root node, it can be seen that material type has the greatest impact on splitting the material classes. Under such one-step division, 86.40% of non-generic material samples are divided into the left node, and 81.87% of generic material samples are divided into the right node. The other important factors are ranked in order of the kinds of engineering used, total order times, average of order lead time, total outbound quantity, and other attributes.

Due to the limited accumulation time for some material sample data and the human experience noise across the dataset, there are still numerous misclassified samples overall. Addressing this issue

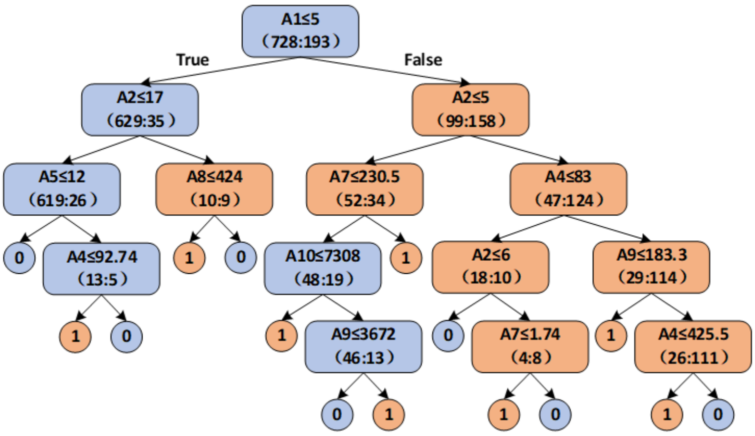


Figure 4: Classification rules of space product material

necessitates standardizing the procurement process, implementing rigorous data entry procedures in subsequent management, or incorporating additional distinctive attributes from a technical perspective for data mining.

By applying the WSD_CART algorithm to accurately classify materials, significant differences between non-generic and generic materials can be clearly identified. Non-generic materials are diverse and primarily targeted at a few exclusive products, with relatively low frequency of use. Therefore, we recommend continuing to use the traditional small-batch, multi-batch ordering method to ensure timely and flexible availability of such materials to meet specific needs [27].

For the mainstream branches of generic materials, their characteristics are more distinct: stable quality and performance ($A_1 > 5$), targeting multiple models ($A_2 > 5$), controllable delivery cycles ($83 < A_4 \leq 425.5$), and high frequency of use ($A_9 > 183.3$). These traits align with the industry's definition of generic materials and offer opportunities for enterprises to optimize their ordering strategies. We strongly recommend adopting a more efficient and customized combination batch ordering strategy to replace the traditional cumbersome ordering method for such materials [28]. By integrating demand and optimizing procurement plans, enterprises can reduce procurement costs and improve overall supply chain efficiency [29].

Furthermore, it's important to note that general materials commonly experience extended lead times in their ordering process due to high and frequent demand, necessitating longer preparation times for suppliers. To address this issue effectively, we suggest that enterprises establish closer partnerships with suppliers. By collaboratively enhancing the ordering process, they can efficiently reduce lead times and ensure timely supply of general materials [30].

5 Conclusions

This study introduces a novel approach to addressing imbalanced classification in DTs through the development of the split difference index. Our proposed algorithms, SDDT and WSD-CART, demonstrate significant improvements in majority class recognition while maintaining high overall classification accuracy across various datasets.

The experiments on UCI datasets with different levels of imbalance validate the robustness of our approach. Notably, in the space product material classification task, our method increased the true positive rate for majority class identification from 66.32% to 76.17%, showcasing its practical applicability in real-world scenarios.

The strength of our approach lies in its ability to provide both improved classification performance and interpretable decision rules. This dual advantage makes it particularly valuable for domains where understanding the decision-making process is as crucial as the accuracy of the classification itself.

However, it's important to acknowledge the limitations of this study. The computational complexity of our algorithms, especially for large datasets, needs further investigation. Additionally, while our method shows improvements across various imbalance levels, its performance in difficult classification areas within some data samples requires more extensive testing.

Future research directions include integrating this approach with ensemble methods to potentially further enhance performance. Exploring the applicability of the split difference concept to other machine learning paradigms beyond DTs could also yield interesting insights.

In conclusion, this work contributes to the ongoing efforts to address the challenges of imbalanced data in machine learning. By offering a new perspective on DT split criteria, we hope to inspire further innovations in this critical area of research, ultimately leading to more robust and fair classification systems across diverse applications.

References

[1] Mjahed, O.; Hadaj, S.E.; Guarmah, E.M.E.; Mjahed, S. (2022). Bio-Inspired hybridization of artificial neural networks for various classification tasks, *Studies in Informatics and Control*, 31(3), 21–30, 2022.

[2] Du, H.; Zhang, Y.; Zhang, L.; Chen, Y. (2023). Selective ensemble learning algorithm for imbalanced dataset, *Computer Science and Information Systems*, 20(2), 831–856, 2023.

[3] Lai, W. (2023). Default prediction of internet finance users based on imbalance-xgboost, *Technical Gazette*, 30(3), 779–786, 2023.

[4] Kamaladevi M.; Venkatraman V. (2021). Tversky Similarity based Under Sampling with Gaussian Kernelized Decision Stump Adaboost Algorithm for Imbalanced Medical Data Classification, *International Journal of Computers Communications & Control*, 16(6), 4291, 2021.

[5] Zhang, K. (2023). Using deep learning to automatic inspection system of printed circuit board in manufacturing industry under the internet of things, *Computer Science and Information Systems*, 20(2), 723–741, 2023.

[6] Pang, J.L. (2023). Adaptive fault prediction and maintenance in production lines using deep learning, *International Journal of Simulation Modelling*, 22(4), 734–745, 2023.

[7] Li, Z.; Huang, M.; Liu, G.; Jiang, C.(2021). A hybrid method with dynamic weighted entropy for handling the problem of class imbalance with overlap in credit card fraud detection, *Expert Systems with Applications*, 175, 114750, 2021.

[8] Huang S.; Lei K. (2020). IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks, *Ad Hoc Networks*, 105, 102177, 2020.

[9] Goyal, P.; Verma, D.K.; Kumar, S. (2023). Diagnosis of Plant Leaf Diseases Using Image Based Detection and Prediction Using Machine Learning Approach, *Economic Computation and Economic Cybernetics Studies and Research*, 57(4), 293–312, 2023.

[10] Sun T.; Zhou Z. (2018). Structural diversity for decision tree ensemble learning, *Frontiers of Computer Science*, 12, 560–570, 2018.

[11] Wang, J.; Zhu, B.; Liu, P.; Jia, R.; Jia, L.; Chen, W.; Feng, C.; Li, J. (2021). Screening Key Indicators for Acute Kidney Injury Prediction Using Machine Learning, *International Journal of Computers Communications & Control*, 16(3), 4180, 2021.

[12] Aaboub F.; Chamlal H.; Ouaderhman T. (2023). Statistical analysis of various splitting criteria for decision trees, *Journal of Algorithms & Computational Technology*, 17, 17483026231198181, 2023.

[13] Dietterich, T.; Kearns, M.; Mansour Y. (1996, July). Applying the weak learning framework to understand and improve C4.5, In *Proc. 13th Int'l Conf. Machine Learning*, 96–104, 1996.

[14] Cieslak, D.; Chawla, N. (2008). Learning decision trees for unbalanced data, In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2008*, Springer Berlin Heidelberg, 241–256, 2008.

[15] Park Y.; Ghosh J. (2012). Ensembles of α-Trees for Imbalanced Classification Problems, *IEEE Transactions on Knowledge and Data Engineering*, 26(1), 131–143, 2012.

[16] Boonchuay K.; Sinapiromsaran K.; Lursinsap C. (2017). Decision tree induction based on majority entropy for the class imbalance problem, *Pattern Anal Applic*, 20, 769–782, 2017.

[17] Liu W.; Chawla S.; Cieslak D.; Chawla, N.V. (2010, April). A robust decision tree algorithm for imbalanced data sets, In *Proceedings of the 2010 SIAM International Conference on Data Mining*, 766–777, 2010.

[18] Hong, J.S.; Lee, J.; Sim, M.K. (2024). Concise rule induction algorithm based on one-sided maximum decision tree approach, *Expert Systems with Applications*, 237, 121365, 2024.

[19] Lv X.; Liu C.; Zhu J. (2011). Improved Algorithm of Decision Tree Based on Key Decision Factor and Its Applications in Railway Transportation, *Journal of the China Railway Society*, 33(09), 62–67, 2011.

[20] Chandra B.; Kothari R.; Paul P. (2010). A new node splitting measure for decision tree construction, *Pattern Recognition*, 43(8), 2725–2731, 2010.

[21] Zhang S.C. (2012). Decision tree classifiers sensitive to heterogeneous costs, *Journal of Systems and Software*, 85(4), 771-779, 2012.

[22] Rodríguez, J.J.; Díez-Pastor, J.F.; García-Osorio, C. (2011). Ensembles of decision trees for imbalanced data, In *International workshop on multiple classifier systems*, Berlin, Heidelberg: Springer Berlin Heidelberg, 76-85, 2011.

[23] Yang, H. (2023). A random forest approach to appraise personal credit risk of internet loans, *Technical Gazette*, 30(2), 492-498, 2023.

[24] Japkowicz, N. (2013). Assessment metrics for imbalanced learning, *Imbalanced learning: Foundations, algorithms, and applications*, 187-206, 2013.

[25] Blakey-Milner, B.; Gradl, P.; Snedden, G.; Brooks, M.; Pitot, J.; Lopez E.; Leary M.; Berto F.; Du Plessis A. (2021). Metal additive manufacturing in aerospace: A review, *Materials & Design*, 209, 110008, 2021.

[26] Djari, A. (2023) Influence of the membership functions number of fuzzy logic controller on the performances of dynamic systems; *Romanian Journal of Information Technology & Automatic Control/Revista Română de Informatică și Automatică*, 33(1), 93–106. 2023.

[27] Li, Z.P. (2022). Management decisions in multi-variety small-batch product manufacturing process, *International Journal of Simulation Modelling*, 21(4), 537-547, 2022.

[28] Clempner, J.B. (2023). An Ergodic and Transient Markov Model for Penalty Regularised Portfolio, *Economic Computation and Economic Cybernetics Studies and Research*, 57(4), 275-292, 2023.

[29] Zhang, Y.M.; Song, Y.F.; Meng, X.; Liu, Z.G. (2023). Optimizing supply chain efficiency with fuzzy critic-edas, *International Journal of Simulation Modelling*, 22(4), 723-733, 2023.

[30] Negoiţă, R.F.; Borangiu, T. (2023). Robotic Process Automation of Inventory Demand with Intelligent Reservation, *Studies in Informatics and Control*, 32(2), 5-14. 2023.

Cite this paper as:

Zhou, T.; Gao, X.; Sun, X.; Han, L. (2024). Split Difference Weighting: An Enhanced Decision Tree Approach for Imbalanced Classification, *International Journal of Computers Communications & Control*, 19(6), 6702, 2024.

https://doi.org/10.15837/ijccc.2024.6.6702