

Article

A Broad TSK Fuzzy Classifier with a Simplified Set of Fuzzy Rules for Class-Imbalanced Learning

Jinghong Zhang ¹, Yingying Li ¹, Bowen Liu ¹, Hao Chen ¹, Jie Zhou ², Hualong Yu ¹  and Bin Qin ^{1,*}¹ School of Computer, Jiangsu University of Science & Technology, Zhenjiang 212100, China² Department of Computer Science & Engineering, Shaoxing University, Shaoxing 312000, China

* Correspondence: qinbin_sd@126.com

Abstract: With the expansion of data scale and diversity, the issue of class imbalance has become increasingly salient. The current methods, including oversampling and under-sampling, exhibit limitations in handling complex data, leading to overfitting, loss of critical information, and insufficient interpretability. In response to these challenges, we propose a broad TSK fuzzy classifier with a simplified set of fuzzy rules (B-TSK-FC) that deals with classification tasks with class-imbalanced data. Firstly, we select and optimize fuzzy rules based on their adaptability to different complex data to simplify the fuzzy rules and therefore improve the interpretability of the TSK fuzzy sub-classifiers. Secondly, the fuzzy rules are weighted to protect the information demonstrated by minority classes, thereby improving the classification performance on class-imbalanced datasets. Finally, a novel loss function is designed to derive the weights for each TSK fuzzy sub-classifier. The experimental results on fifteen benchmark datasets demonstrate that B-TSK-FC is superior to the comparative methods from the aspects of classification performance and interpretability in the scenario of class imbalance.

Keywords: TSK fuzzy classifiers; fuzzy rule reduction; weighted fuzzy rule; broad ensemble learning; class-imbalanced learning

MSC: 68T05



Citation: Zhang, J.; Li, Y.; Liu, B.; Chen, H.; Zhou, J.; Yu, H.; Qin, B. A Broad TSK Fuzzy Classifier with a Simplified Set of Fuzzy Rules for Class-Imbalanced Learning. *Mathematics* **2023**, *11*, 4284. <https://doi.org/10.3390/math11204284>

Academic Editor: José Antonio Sanz

Received: 10 September 2023

Revised: 5 October 2023

Accepted: 10 October 2023

Published: 13 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The issue of class imbalance has been a challenge in the fields of data mining, computer vision, and machine learning over the past decade [1,2]. The class imbalance problem is prominently visible within machine learning datasets, where the distribution of instances across diverse classes is drastically uneven. Classical machine learning models tend to be biased towards majority classes. Conversely, minority classes can hardly determine the parameters of the classifier. However, in resolving practical issues, we typically focus more on these minority classes, which usually contain critical information. For instance, in the diagnosis of cancer, misclassifying a positive instance as a negative one may lead to severe loss. Similarly, the inability to promptly identify a minuscule number of faulty conditions in aircraft operation state fault detection could potentially trigger accidents. Therefore, with the diversification of data scenarios and the increasing complexity of data, the challenge posed by the class imbalance problem is becoming increasingly prominent. This problem finds extensive applications across multiple domains, such as detecting oil spills in satellite images, identifying the cause of power distribution failures [3], predicting potential customer churn [4], and face recognition [5].

Currently, the strategies to address the issue of class imbalance learning can be generally categorized into three types:

- (1) Sampling methods including oversampling and under-sampling techniques [2,6]. These methods aim to adjust the distribution of original data instances to approach a balanced state, thereby improving the model's predictive ability for minority classes.

- Random Over-Sampling (ROS) randomly increases the number of minority class instances [6]. Random Under-Sampling (RUS) randomly reduces the number of majority class instances [2]. Although these strategies show some effectiveness, their reliance on simple replication or deletion of original data instances can potentially lead to overfitting or loss of information. In response, an oversampling method named the Synthetic Minority Over-sampling Technique (SMOTE) has been proposed [7]. This method mitigates the risk of overfitting by performing orderly interpolation between minority class instances, thereby enhancing the capability of dealing with the class imbalance issue. In the literature [8], the combination of a Fuzzy Support Vector Machine (FSVM) with instance relative density information provides a more efficient approach for classification tasks with a complex class imbalance problem [9].
- (2) Cost-sensitive learning methods. These methods construct a cost-weight matrix by analyzing factors such as the error costs of the minority and majority classes, training costs, and instance quantities, thereby achieving an effect in dealing with class imbalance. These methods focus on the difference impact on the loss function of the misclassification of the instances in minority and majority classes. Cost-sensitive weight matrices are constructed by analyzing factors including misclassification costs, training costs, and instance numbers of minority and majority classes. With the employment of this weight matrix, the methods will protect the distribution region of minority classes instead of just pursuing high accuracy. In cost-sensitive learning methods, if minority classes are prone to misclassification, they will be assigned greater weights via a specific cost matrix [10]. Conversely, since majority classes are seldom misclassified, they will be assigned smaller weights to enhance the model's classification performance for the minority classes. For example, by incorporating the concept of cost matrix weighting into Extreme Learning Machines (ELM) [11], researchers have proposed a high-performing and computationally efficient Weighted Extreme Learning Machine (WELM) method [12]. By combining cost-sensitive thinking with ensemble learning, the literature [13] introduces a sensitive decision tree ensemble method. In particular, the advent of AdaCost [14], a cost-sensitive boosting method combined with the Boosting ensemble method, has greatly improved the prediction accuracy for minority classes by incorporating an optimized weight update strategy and the strengths of the AdaBoost method. Notably, Support Vector Machine (SVM) methods have consistently performed well in classification effects. The method proposed in the literature [15] combines a Fuzzy Support Vector Machine (FSVM) with cost sensitivity, assigning greater weights to the instances of minority classes to address class imbalance [9]. A novel approach proposed in the literature [16] combines cost sensitivity with a Broad Learning System (BLS), using weighted penalty factors to constrain each instance's contribution in different classes, allocating higher weights to the instances of smaller classes to enhance their contribution. Reference [17] presents a cost-sensitive variable selection method for Bayesian network classifiers, which optimizes the performance of multi-class classification problems with class imbalance in practical applications. In cost-sensitive methods, how to determine the weights is an open research hotspot [18].
 - (3) Hybrid methods for class imbalance problems. These methods primarily combine the above two strategies or integrate them with advanced techniques, e.g., ensemble learning, cluster learning, and deep learning, thereby enhancing the capacity to handle class imbalance problems. These methods usually employ cost-sensitive learning methods in the form of ensemble learning after, respectively, oversampling and/or under-sampling the minority classes and majority classes. In the data preprocessing stage, sampling methods such as SMOTE are used to balance the distribution of data instances [7], and then classic methods such as KNN and CART4.5 are employed to learn from these more balanced data. This has been proven to be an effective hybrid strategy. The advantages of ensemble learning techniques in enhancing the generalization performance of methods and reducing overfitting have been demonstrated

in the literature [19]. Leveraging the strengths of ensemble learning, several highly robust and generalizable methods such as SMOTEBagging [19], SMOTEBoost [20], UnderBagging [21], RUSBoost [22], and OverBoost [23] have been proposed. These approaches incorporate advanced sampling techniques into ensemble method frameworks, including Bagging and Boosting, forming advanced class imbalance ensemble frameworks. In the field of class imbalance learning, ensemble methods have shown higher robustness and foresight compared to single classifiers; hence, the method proposed in this paper also cleverly uses the unique advantages of ensemble techniques in the field of class-imbalanced learning.

Fuzzy systems are considered a specific structure of artificial neural networks [24], with their peculiarity being their rule-based learning mechanism. Due to the excellent linguistic interpretability of both the antecedent parts and consequent parts of fuzzy rules, fuzzy systems not only possess high mathematical approximation capabilities akin to neural networks but also have excellent language interpretation properties [25–28].

Fuzzy systems have unique application potential in solving class-imbalanced learning problems, mainly manifested in two aspects, i.e., integrating sampling-level strategies with fuzzy systems and the special handling of fuzzy rules [29]. The class imbalance problem is usually addressed by applying a weighting process to the fuzzy rules. For example, the fuzzy rule weighting scheme proposed in [30] is based on a collaborative voting score between instances and fuzzy rules, and the genetic method used in [31] optimizes the fuzzy rules. In addition, related studies adopted different fuzzy rule weighting generation strategies based on the fuzzy rules' adaptability to instances, effectively enhancing the performance of fuzzy rules under a class-imbalanced environment [32,33]. However, these fuzzy rule weighting methods are excessively complex in their weight generation systems, and their improvement in performance is not significant. Therefore, this study will further propose a more concise, effective, and interpretable fuzzy rule weighting optimization strategy in the environment of imbalanced data.

While the state-of-the-art methods have achieved satisfactory classification accuracy for class-imbalanced data, they still face challenges when dealing with complex data. For instance, sampling methods might lead to information loss or overfitting, and cost-sensitive methods can hardly achieve enhanced generalization performance in highly imbalanced data scenarios. Moreover, due to the dependence of fuzzy rules on data features, fuzzy systems tend to face the challenge of fuzzy rule explosion when used to tackle complex data scenarios [34].

In this study, a novel broad TSK fuzzy classifier with a simplified set of fuzzy rules for imbalanced learning (B-TSK-FC) is proposed. Thus, fuzzy systems may benefit from the selection and weighted optimization of fuzzy rules to reduce the number of fuzzy rules and solve class imbalance problems more effectively. We adopt a zero-order TSK fuzzy classifier whose fuzzy rules are significantly more flexible and linguistically interpretable than classical methods. First, we propose a method to reduce the number of fuzzy rules to enhance the interpretability and improve the classification performance of the classifier [35]. Second, we propose a concise weighting scheme that assigns different weights to the fuzzy membership of the instances from minority and majority classes. Finally, using the above mechanism, a series of zero-order TSK fuzzy sub-classifiers are generated and assembled in a broad manner, which can improve the classification accuracy of all classes in the class-imbalanced data. This strategy further improves generalization performance and effectively reduces the risk of overfitting. The principal contributions of this study are listed as follows.

- (1) Even though the random generation of fuzzy rules with equal partitions along each feature has been widely adopted by the current TSK fuzzy classifier methods [25,27,36], the exclusion of ineffective fuzzy rules has yet to be of concern, which may lead to an increase in the number of fuzzy rules and inevitably damage the interpretability of the model. Based on the adaptability of the antecedent parts and consequent parts of fuzzy rules to different complex data environments, we propose a fuzzy rule

simplification strategy that effectively reduces the number of fuzzy rules, enhances the interpretability of the TSK fuzzy classifier, and improves the classification performance of the classifier.

- (2) Different from the current methods in which all the fuzzy rules are considered indiscriminately when facing a classification task on class-imbalanced data, the fuzzy rules of the TSK fuzzy sub-classifiers in B-TSK-FC may play significantly different roles in the classification task. For the scenario of class-imbalanced learning, we recognize that fuzzy rules contain knowledge of different data distributions. By generating a weight matrix that leverages information about the number of classes in the data, we propose a concise and easy-to-implement fuzzy rule weighting scheme to modify the fuzzy system to adapt to class-imbalanced scenarios. This fuzzy rule weighting scheme is coincident with the working manner of human thinking in which different knowledge works with different magnitudes.
- (3) In a class-imbalanced data environment, guided by the objective of improving the classification accuracy of each class, we propose a dynamic weighted ensemble strategy that effectively enhances the prediction accuracy of each class. By assembling a series of zero-order TSK fuzzy sub-classifiers in a broad manner, we significantly improve the generalization performance of the system and effectively reduce the risk of overfitting while maintaining interpretability.
- (4) Comparative experimental results from fifteen benchmark datasets and the state-of-the-art comparative methods demonstrate the efficiency of our proposed B-TSK-FC fuzzy classifier in class-imbalanced scenarios in both linguistic interpretability and superior classification performance.

Therefore, the study presented in this research provides new insight into how to simplify fuzzy rules and achieve the resultant classification performance of the TSK fuzzy classifier over class-imbalanced data. The remainder of this paper is organized as follows. Section 2 provides a brief introduction to the zero-order TSK fuzzy classifier. Section 3 elaborates on our proposed B-TSK-FC broad fuzzy classifier and provides a theoretical analysis for its efficiency in classification performance enhancement. Section 4 presents the experimental results of the proposed B-TSK-FC and five comparative methods over fifteen benchmark datasets, which confirm the superiority of B-TSK-FC relative to the comparative methods. Finally, Section 5 summarizes and reviews the entire paper. The full names of the abbreviations in this study are introduced in Table A1 in Appendix A for easy reading.

2. Classical Zero-Order TSK Fuzzy Classifier

Since B-TSK-FC proposed in this study is composed of several zero-order TSK fuzzy classifiers, this section introduces the classical zero-order TSK fuzzy classifier, which contains a set of fuzzy rules expressed as follows [36]:

$$\begin{aligned} &\text{IF } x_1 \text{ is } A_1^k \wedge x_2 \text{ is } A_2^k \wedge \cdots \wedge x_d \text{ is } A_d^k \\ &\text{THEN } y^k = a^k, k = 1, 2, \dots, K. \end{aligned} \quad (1)$$

where a^k is the constant in the consequent part of the k th fuzzy rule, x_d is the d th feature of the input instance, A_j^k is the antecedent part of the k th fuzzy rule of fuzzy set on the j th feature, and K is the total number of fuzzy rules in the fuzzy system. The TSK fuzzy classifier uses the Gaussian function as the fuzzy membership, which is expressed as follows:

$$\phi_j^k(x_j) = \exp\left(-\frac{1}{2}\left(\frac{x_j - s_j^k}{\sigma_j^k}\right)^2\right) \quad (2)$$

where x_j is the j th feature of the instance, and s_j^k and σ_j^k , respectively, denote the center and width of the Gaussian membership function. Therefore, the output of the zero-order TSK classifier corresponding to the input instance \mathbf{x} can be expressed as:

$$Y = \sum_{k=1}^K \frac{\mu^k(\mathbf{x})}{\sum_{r=1}^K \mu^r(\mathbf{x})} a^k = \sum_{k=1}^K \tilde{\mu}^k(\mathbf{x}) a^k \quad (3)$$

where the antecedent part of the fuzzy rule is calculated as $\mu^k(\mathbf{x}) = \prod_{j=1}^d \phi_j^k(x_j)$, $k = 1, \dots, K$. Additionally, a^k is the consequent part of the k th fuzzy rule. For binary classification tasks, we conventionally treat class labels as $\{0, +1\}$. In this setting, based on the input of training instances, we can easily obtain the classifier's output on the testing set and intuitively distinguish between positive and negative classes. A common method is to normalize the output y to the interval 0 to 1 and consider this output the predictive probability for the +1 class label. The classification threshold is typically set at 0.5; the larger the output, the higher the probability of prediction as the +1 class, and vice versa. However, for multi-classification tasks facing multiple class labels, to ensure independence between class labels, we adopt one-hot encoding for binary coding [37], i.e., representing each class label with a C -bit binary number in which only the bit corresponding to the label is set to 1 and the other bits are set to 0. In the training of the zero-order TSK fuzzy classifier, we randomly select the center of the membership function from $[0, 0.25, 0.5, 0.75, 1]$ and represent the partition with five Gaussian functions. Notably, even though the center values are randomly selected, there still exists the linguistic interpretations "very bad", "bad", "medium", "good", "very good", etc. Despite its excellent interpretability and mathematical approximation capability, the zero-order TSK fuzzy classifier is not suitable for class-imbalanced scenarios and faces the challenge of the curse of dimensionality and fuzzy rule explosion [34].

In the following sections, we will first introduce the proposed B-TSK-FC broad ensemble fuzzy classifier, discussing in detail how to improve the classic zero-order TSK classifier to adapt it to imbalanced data and how to effectively reduce the number of fuzzy rules while ensuring the enhancement of the classification performance of the zero-order TSK classifier.

3. The Proposed Method

In this section, we provide a detailed description of the proposed B-TSK-FC based on the dynamic selection and weighted optimization of fuzzy rules. This work is inspired by the following three points.

- (1) In order to solve the fuzzy rule explosion problem, which is encountered by current fuzzy systems in complex and variable data environments [34], we adopt a strategy for simplifying fuzzy rules and improving the quality of fuzzy rules. While the random selection of the centers of fuzzy rule antecedent parts offers interpretability, some initially generated fuzzy rules may not align well with data characteristics, indicating low fuzzy rule quality. Since the adaptability of a fuzzy rule to specific data scenarios is primarily reflected in its antecedent parts and consequent parts, we simplify the fuzzy rules according to the antecedent parts and consequent parts to improve the quality of fuzzy rules.
- (2) Although the current class-imbalanced learning techniques have achieved significant progress in classification performance, they are not interpretable. As a result, we choose the zero-order TSK fuzzy classifier, known for its excellent interpretability, incorporate cost-sensitive reasoning, and propose a simple yet effective fuzzy rule weighting method. This allows the TSK fuzzy classifier to tackle class-imbalanced data more efficiently.
- (3) After the above improvements, the TSK fuzzy classifier has a strong class imbalance classification performance. However, the generated TSK fuzzy sub-classifiers are similar to each other. Using conventional simple voting for the ensemble would

restrict performance enhancement. Hence, we employ the idea of a class-imbalanced G-mean metric with the objective of optimizing per-class classification accuracy. This approach allows for the reasonable weighting of individual fuzzy sub-classifiers within the ensemble. This not only enhances the generalization performance of the ensemble classifier but also reduces the risk of overfitting.

In the subsequent sections, we will delve into the B-TSK-FC fuzzy classification system in three steps in Section 3.1. In Section 3.2, we will theoretically analyze and prove its advantages in terms of performance and interpretability. Finally, in Section 3.3, we will analyze its time complexity.

3.1. Structure of B-TSK-FC

The core components of B-TSK-FC primarily encompass the three strategies introduced: fuzzy rule selection, fuzzy rule weighted optimization, and the final “G-mean” broad-weighted ensemble. We will begin by elucidating the fuzzy rule selection strategy.

In Equation (2), we can distinctly observe the correlation between the membership function values of instances in each dimension and the antecedent parts of fuzzy rules: a higher membership function value corresponds to a larger antecedent part of the fuzzy rule. This suggests that if we can identify an ideal membership function along with its central parameter value, it would offer a valuable reference standard for fuzzy rule selection. Similarly, the greater the value of a fuzzy rule’s consequent part, the higher the weight attributed to the corresponding fuzzy rule. During the decision-making process for instances, this translates to assigning higher weights. Figure 1 depicts the architecture for fuzzy rule selection and fuzzy rule weighted optimization. In Figure 1, Step (a) uses training data to generate K fuzzy rules. The quality and activation levels of these fuzzy rules vary with respect to instances. Differences in activation levels are chiefly reflected in the fuzzy rule’s antecedent parts and consequent parts: the higher their values, the stronger the activation towards the instance, suggesting better adaptability of the fuzzy rule to specific data. Therefore, in Step (b), based on the product of the fuzzy rule’s antecedent part and consequent part, we annotate the fuzzy rules with larger products with solid lines and those with smaller products with dashed lines. Subsequently, in Step (c), we select the high-quality fuzzy rules indicated by solid lines to form a new set of fuzzy rules, building the classifier. While these fuzzy rules exhibit superior classification capacity for specific data scenarios, they do not specifically deal with class-imbalanced data. Hence, in Step (d), we categorize each fuzzy rule from top to bottom into different class-related parts. The portions of the fuzzy rule corresponding to the minority class are assigned a higher weight, whereas the portions corresponding to the majority class are given a lesser weight, thereby enhancing the training of minority class information in the fuzzy rule. This effectively amplifies the capability of the fuzzy rule to handle imbalanced data.

Next, we provide a detailed description of the training process for the t th fuzzy sub-classifier in the B-TSK-FC ensemble fuzzy classification framework, following the improvement of fuzzy rules. To begin, we divide the dataset into three subsets: a training set, validation set, and testing set. We then compute the membership functions across each dimension for the training data. The training process of a TSK fuzzy sub-classifier is introduced in Algorithm 1. Based on Step 2 of Algorithm 1, we generate the antecedent part matrix Φ_t for the t th sub-classifier. Each column in the Φ_t matrix is the antecedent part membership function information of an individual fuzzy rule. Consequently, we obtain the antecedent part matrix Φ_t corresponding to the initially derived K fuzzy rules.

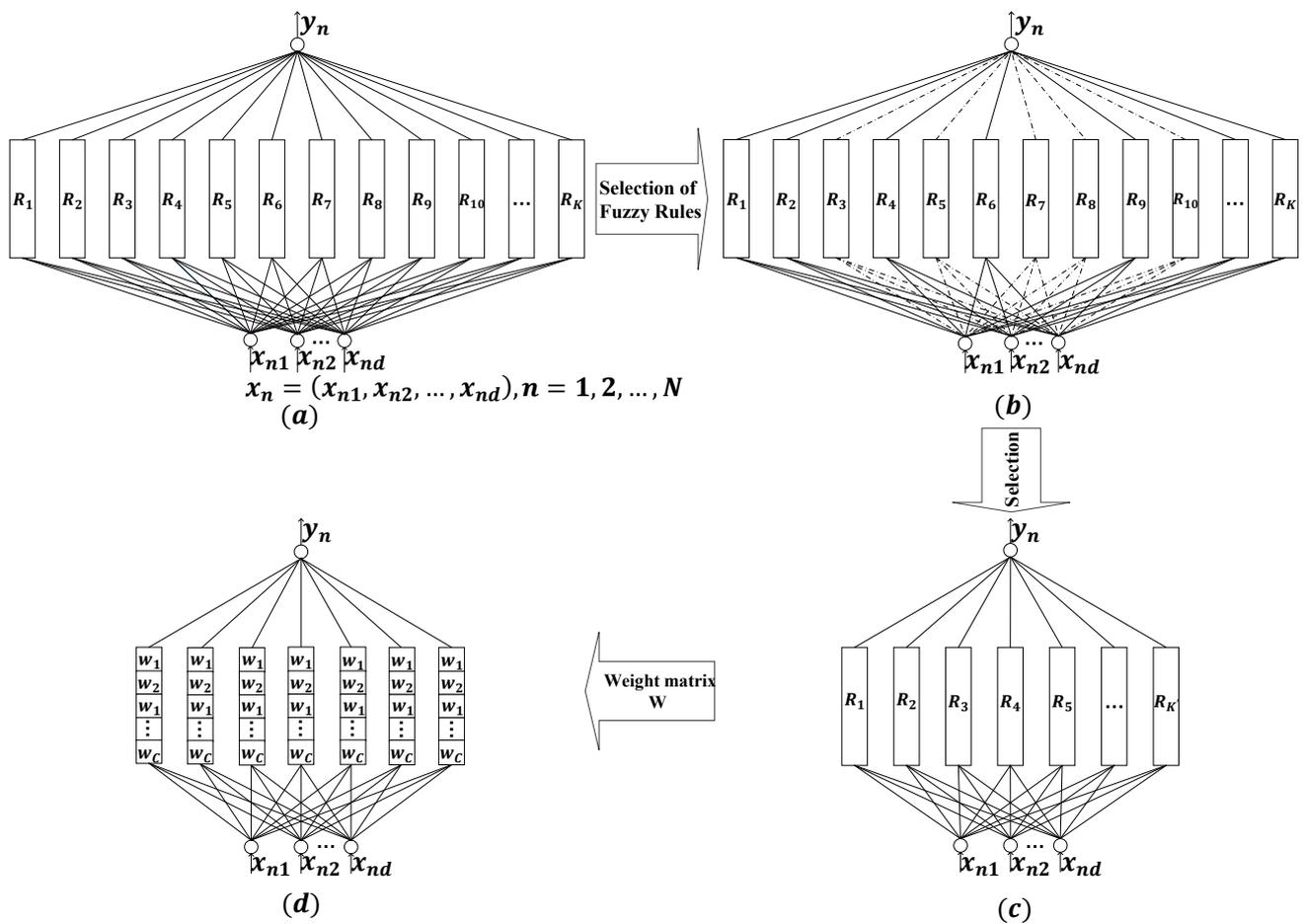


Figure 1. An illustrative architecture of the fuzzy rule reduction and weighted optimization of fuzzy rules: (a) Training of initial fuzzy rules. (b) Solid lines represent fuzzy rules with larger products of antecedent parts and consequent parts, while dashed lines represent those with smaller products. (c) Selecting fuzzy rules with larger products of antecedent parts and consequent parts. (d) Using a weight matrix to apply weights to different class sections of each fuzzy rule.

Algorithm 1 Training process of the t th fuzzy sub-classifier

Input: Training dataset $D = [X, Y]$, consisting of $X = [x_1, x_2, \dots, x_N]^T$, and the corresponding class labels $Y = [y_1, y_2, \dots, y_N]^T$. Here, $x_n = [x_1, x_2, \dots, x_d]$, $n = 1, 2, \dots, N$ is the number of training instances, and d is the total dimension of the instance. For binary classification, $y_n \in \{0, +1\}$. For multi-class classification, y_n is transformed into a one-hot encoded binary vector, as outlined in [37]. The method requires a pre-set initial fuzzy rule count K , optimized fuzzy rule count K' where $K' < K$, $k' = 1, 2, \dots, K'$, a regularization constant parameter λ , and the width of the Gaussian function, denoted as σ_{tj}^k , in which $t = 1, 2, \dots, T$, $j = 1, 2, \dots, d$, $k = 1, 2, \dots, K$.

Output: $a'_t = [a'_t{}^1, a'_t{}^2, \dots, a'_t{}^{K'}]^T$, the consequent part parameters of the learned fuzzy rules in the t th zero-order TSK fuzzy sub-classifier, the antecedent part matrix $\Phi'_t = [\phi'_t(x_n)w_c]_{N \times K'}$ after fuzzy rule improvement, where w_c is the weight of the corresponding fuzzy rule.

Procedure:

Step 1 Using distribution information across classes, construct a diagonal weight matrix W . Let the number of instances of class c in the training dataset be denoted as N_c , where $c \in \{1, 2, \dots, C\}$ is the class label of the instances. The total number of training instances is N , and the weight diagonal matrix W is defined as $W = \text{diag}(w_1, w_2, \dots, w_c, \dots, w_C)$, where $w_c = N/N_c$. Here, “diag” denotes a diagonal matrix where the diagonal elements are the provided values, and all off-diagonal elements are zero. In this context, W is an $N \times N$ diagonal matrix.

Algorithm 1 Count.

Step 2 Compute the Gaussian membership function for each feature of the instance, defined as follows for the k th fuzzy rule and j th input feature.

$$\phi_{tj}^k(x_j) = \exp\left(-\frac{1}{2}\left(\frac{x_j - s_{tj}^k}{\sigma_{tj}^k}\right)^2\right) \tag{4}$$

where $j = 1, 2, \dots, d$ and $s_{tj}^k \in \{0, 0.25, 0.5, 0.75, 1\}$ denotes the center of the k th fuzzy rule along the j th feature. Here, t denotes the t th fuzzy sub-classifier, and σ_{tj}^k is determined either manually or using the method described in [36]. Then, compute the normalized membership function value for the instance \mathbf{x}_n under the k th fuzzy rule.

$$\phi_t^k(\mathbf{x}_n) = \frac{\sum_{j=1}^d \phi_{tj}^k(x_{nj})}{\sum_{r=1}^K \prod_{j=1}^d \phi_{tj}^r(x_{nj})} \tag{5}$$

where $n = 1, 2, \dots, N$.

Step 3 Computing the consequent part of the fuzzy rule. Initially, the number of fuzzy rules is set to K . The consequent part parameter matrix of the fuzzy rule is defined as \mathbf{a}_t . Subsequently, based on [35,36], it can be transformed into a linear equation form.

$$\Phi_t \mathbf{a}_t = \mathbf{Y} \tag{6}$$

By introducing the identity matrix $\mathbf{I}_{K \times K}$ and using the LLM [38–40], the consequent part parameter of the fuzzy rule can be determined as

$$\mathbf{a}_t = \left(\frac{1}{2\lambda} \mathbf{I} + \Phi_t^T \Phi_t\right)^{-1} \Phi_t^T \mathbf{Y} \tag{7}$$

Step 4 Calculate the matrix \mathbf{E} of the antecedent parts and consequent parts of the fuzzy rules.

$$\mathbf{E} = \Phi_t \mathbf{a}_t = \left[\phi_t^k(\mathbf{x}_n) a^k\right]_{N \times K} \tag{8}$$

Step 5 Select K' fuzzy rules corresponding to the K' columns in \mathbf{E} that have the largest K' average values and construct matrix $\Phi'_t = \left[\phi_t^{k'}(\mathbf{x}_n) w_c\right]_{N \times K'}$.

Step 6 Let the consequent part of the optimized fuzzy rules be denoted as $\mathbf{a}'_t = [a^1, a^2, \dots, a^{K'}]^T$. Using the weighted matrix Φ'_t , recalculate the consequent part \mathbf{a}'_t of the fuzzy rules, and again transform it into a linear equation form as suggested by [35,36].

$$\Phi'_t \mathbf{a}'_t = \mathbf{Y} \tag{9}$$

Introduce the identity matrix $\mathbf{I}_{K' \times K'}$ and use the LLM to derive the parameters for the improved consequent part of the fuzzy rule [38–40].

$$\mathbf{a}'_t = \left(\frac{1}{2\lambda} \mathbf{I} + \Phi'^T_t \Phi'_t\right)^{-1} \Phi'^T_t \mathbf{Y} \tag{10}$$

Step 7 Return \mathbf{a}'_t, Φ'_t .

$$\Phi_t = \left[\phi_t^k(\mathbf{x}_n)\right]_{N \times K} \tag{11}$$

Subsequent to Method 1's Steps 3 and 4, using the antecedent part matrix Φ_t , we derive the consequent parts of the fuzzy rules, forming the consequent part given by

$$\Phi_t \mathbf{a}_t = \mathbf{Y} \tag{12}$$

where,

$$\mathbf{a}_t = [a^1, a^2, \dots, a^K]^T = \left(\frac{1}{2\lambda} \mathbf{I} + \Phi_t^T \Phi_t \right)^{-1} \Phi_t^T \mathbf{Y} \tag{13}$$

where \mathbf{a}_t denotes the matrix of fuzzy rule consequent parts, $\mathbf{I}_{K \times K}$ is the identity matrix, λ is the regularization constant parameter, and \mathbf{Y} is the set of instance labels. Subsequently, we obtain the product matrix \mathbf{E} of the antecedent parts and consequent parts by multiplying the corresponding columns of the antecedent part matrix with the consequent part matrix.

$$\mathbf{E} = \Phi_t \mathbf{a}_t = [\phi_t^k(\mathbf{x}_n) a^k]_{N \times K} \tag{14}$$

The greater the degree of membership of the antecedent parts and the value of the consequent parts, the higher the adaptability of the fuzzy rule to a specific data scenario. A larger product of the antecedent part and consequent part implies a higher weight in fuzzy decision making. Thus, by computing the product of antecedent parts and consequent parts, we select the columns with larger average values from \mathbf{E} as high-quality fuzzy rules, eliminating other fuzzy rules. Following Step 4 of Algorithm 1, we pick K' fuzzy rules, obtaining the optimized product matrix \mathbf{E}' of the fuzzy rule antecedent parts and consequent parts.

$$\mathbf{E}' = [\phi_t^{k'}(\mathbf{x}_n) a^{k'}]_{N \times K'} \tag{15}$$

where $k' = 1, 2, \dots, K'$ and $K' < K$. We have successfully chosen K' higher-quality fuzzy rules from the initial K fuzzy rules in this way. The chosen center values of the antecedent parts better match the distribution of specific instance data, thus enhancing the overall fuzzy rule quality. We can better adapt to specific data scenarios by optimizing the quality of the fuzzy rules, enhancing the rationality of classification boundaries and thereby strengthening the overall performance of the fuzzy classifier.

It is widely known that too many fuzzy rules may damage the interpretability and barely provide any improvement for the classification performance of the TSK fuzzy classifier. In the iterative training process of the sub-classifiers of B-TSK-FC, the number of fuzzy rules may increase without limit. Based on this consideration, the fuzzy rules of each TSK fuzzy sub-classifier are randomly selected from the same candidate set prepared in advance. The antecedent part of the fuzzy rules in the candidate set is generated through random selection from $\{0, 0.25, 0.5, 0.75, 1\}$ along each feature. The number of fuzzy rules is enough for the TSK fuzzy classifier to achieve unsatisfactory but acceptable classification accuracies. There are many duplicate fuzzy rules between different TSK fuzzy sub-classifiers, which limits the number of fuzzy rules in the resultant B-TSK-FC. There also exists a small part of different fuzzy rules between the TSK fuzzy sub-classifiers. One fuzzy rule that cooperated with different fuzzy rules in different TSK fuzzy sub-classifiers may play different roles. Thus, the differences between TSK fuzzy sub-classifiers are determined by these different fuzzy rules. The numbers of fuzzy rules in the candidate set for the fifteen datasets are introduced in Table 1. In order to guarantee the classification performance of TSK fuzzy sub-classifiers while keeping differences between each pair of TSK fuzzy sub-classifiers, the 15 benchmark datasets used in Table 1 are introduced in detail in Section 4.1.

Table 1. The number of fuzzy rules in the candidate set for the fifteen datasets.

#	Datasets	The Number of Fuzzy Rules in the Candidate Set for Each TSK Fuzzy Sub-Classifier
1	penbased(PEN)	255
2	marketing(MAR)	200
3	turkiyestudentevaluationRSpecific(TUR)	90

Table 1. *Cont.*

#	Datasets	The Number of Fuzzy Rules in the Candidate Set for Each TSK Fuzzy Sub-Classifer
4	DNA(DNA)	400
5	skin(SKI)	220
6	usps(USP)	220
7	musk(MUS)	535
8	vowel0(VOW)	200
9	car-good(CAG)	90
10	thyroid(THY)	280
11	poker-8-9_vs_6(P96)	145
12	shuttle-2_vs_5(SHU)	180
13	poker-8_vs_6(P86)	135
14	letterA(LET)	220
15	page_blocks(PAB)	155

While the above dynamic fuzzy rule selection method enhances the overall performance of the fuzzy classifier, it does not specifically address scenarios with imbalanced data. To improve the performance under such circumstances, we propose a concise and efficient fuzzy rule weighting scheme. Specifically, the weight is determined by the ratio of the total number of training instances N to the number of instances N_c of a particular class in the training set, which are then used to construct the diagonal elements of the weighted diagonal matrix. The weight matrix is defined as an $N \times N$ diagonal matrix, where each diagonal element of a row corresponds to the weighting coefficient of the fuzzy rule generated by the 1 to N training instances. Considering the fuzzy rule matrix rows, which correspond to the 1 to N training instances in sequential order, if a row's instance belongs to a particular class, the diagonal position in the corresponding row of the weight diagonal matrix is set to the ratio of N to N_c . In this way, the values on the diagonal of the matrix represent the ratio of the total instance size to the number of instances in the corresponding class. This approach conveniently constructs the fuzzy rule weighting diagonal matrix, exemplified by the weight matrix \mathbf{W} in Algorithm 1, Step 1.

After generating K initial fuzzy rules for the zero-order TSK fuzzy classifier, we select K' "high-quality" fuzzy rules from them. Then, we apply a weighting mechanism to the antecedent part portion of these K' fuzzy rules, as detailed in Step 5 of Algorithm 1. The antecedent part matrix is given by

$$\Phi'_t = \left[\phi_t^{k'}(\mathbf{x}_n) \right]_{N \times K'} \tag{16}$$

Each column is the membership information of the fuzzy rule after normalization in matrix Φ'_t , while each row depicts the antecedent part knowledge of a specific instance across all fuzzy rules. The row corresponding to a particular instance is multiplied by the weight of the diagonal element in the corresponding row of the weight matrix. This implements the antecedent part weighting based on the ratio of the total number of instances to the number of instances in that specific category. By multiplying the weight matrix with the antecedent part matrix and applying an appropriate transposition, the weighted antecedent part matrix for the fuzzy rules is obtained.

$$\Phi'_t = \left[\phi_t^{k'}(\mathbf{x}_n)w_c \right]_{N \times K'} \tag{17}$$

The weighted fuzzy rules can be described in the following Equation (18):

$$\begin{aligned} &\text{IF } x_1 \text{ is } A_1^k \wedge x_2 \text{ is } A_2^k \wedge \dots \wedge x_d \text{ is } A_d^k \\ &\text{THEN } y^k = w_c a^k, k = 1, 2, \dots, K. \end{aligned} \tag{18}$$

The output of the TSK fuzzy sub-classifiers, which is composed of the weighted fuzzy rules expressed in Equation (9), can be calculated as

$$Y = \sum_{k=1}^K \phi_t^k(\mathbf{x})(w_c a^k) \tag{19}$$

Equations (18) and (19) indicate that the weights do not alter the form of fuzzy rules and the output, which guarantees the interpretability of the TSK fuzzy sub-classifiers.

The weighting process of the fuzzy rules has been completed. Through this mechanism, when a specific row in matrix Φ'_t encompasses membership function knowledge from minority class instances, the respective portion of the fuzzy rule is assigned a larger weight. Conversely, the parts of the fuzzy rule containing membership function information from majority class instances are given reduced weight. This strategy amplifies the weight of the minority class within the fuzzy rules while diminishing that of the majority class, thereby enhancing the classification performance and adaptability for imbalanced data scenarios. Subsequently, after fuzzy rule weighting, we obtain the renewed antecedent part matrix Φ'_t , and further represent it in a linear form [36].

$$\Phi'_t \mathbf{a}'_t = \mathbf{Y} \tag{20}$$

where \mathbf{a}'_t denotes the consequent part parameter vector that the classifier aims to learn, and \mathbf{Y} is the label of the training set. Via pseudoinverse computation, it is derived as

$$\mathbf{a}'_t = \left(\frac{1}{2\lambda} \mathbf{I} + \Phi'^T_t \Phi'_t \right)^{-1} \Phi'^T_t \mathbf{Y} \tag{21}$$

where $I_{K' \times K'}$ signifies the identity matrix, and λ is the introduced large amount. Consequently, we have constructed a zero-order TSK fuzzy classifier optimized through fuzzy rule selection and fuzzy rule weighting. The detailed training process for the t th fuzzy rule-optimized sub-classifier is elaborated in Algorithm 1.

To improve the imbalanced classification performance of B-TSK-FC, we leverage the idea of the Geometric Mean (G-mean) to formulate a loss function encompassing all classes. We then employ gradient descent to optimize this function over a predefined set of validation ensemble instances, thereby obtaining a weight matrix for the sub-classifiers that minimizes class-specific errors. This step allows us to effectively ensemble individual sub-classifiers to better fit the actual imbalanced instance distribution. Algorithm 2 illustrates the entire training process for the B-TSK-FC fuzzy classifier. As shown in step 3 of Algorithm 2, we devised a scheme to weight ensemble sub-classifier outputs based on distinct weights, with the overall output being as follows:

$$H(\mathbf{x}) = w'_1 h_1(\mathbf{x}) + w'_2 h_2(\mathbf{x}) + \dots + w'_T h_T(\mathbf{x}) \tag{22}$$

where $H(\mathbf{x})$ denotes the output of the ensemble classifier on a given instance \mathbf{x} , h_t (where $t = 1, 2, \dots, T$) is the t th sub-classifier, and $\mathbf{w}' = [w'_1, w'_2, \dots, w'_T]^T$ is the ensemble weights for each sub-classifier. To determine the optimal ensemble coefficients under class-imbalanced data, we consider the prediction accuracy across all classes, particularly emphasizing the impact of minority class prediction accuracy on the overall prediction outcome. We have introduced a novel loss function based on the product of mean

squared errors for each class. For instance, in the binary classification scenario, it is defined as follows:

$$l(\mathbf{w}') = \frac{1}{P} \sum_{m=1}^P (\mathbf{w}'^T \mathbf{z}_m - y_m)^2 \times \frac{1}{Q} \sum_{m=P+1}^Q (\mathbf{w}'^T \mathbf{z}_m - y_m)^2 \tag{23}$$

where P and Q , respectively, denote the number of positive and negative instances in the validation set and \mathbf{z}_m is the predicted output of the m th instance across all sub-classifiers. The gradient of this loss function, upon differentiation, is given by

$$\begin{aligned} \nabla l(\mathbf{w}') &= \frac{1}{PQ} \left[\sum_{m=1}^P \frac{\partial(\mathbf{w}'^T \mathbf{z}_m - y_m)^2}{\partial \mathbf{w}'} \sum_{m=P+1}^Q (\mathbf{w}'^T \mathbf{z}_m - y_m)^2 \right. \\ &\quad \left. + \sum_{m=1}^P (\mathbf{w}'^T \mathbf{z}_m - y_m)^2 \sum_{m=P+1}^Q \frac{\partial(\mathbf{w}'^T \mathbf{z}_m - y_m)^2}{\partial \mathbf{w}'} \right] \\ &= \frac{1}{PQ} \left[\sum_{m=1}^P 2(\mathbf{w}'^T \mathbf{z}_m - y_m) \mathbf{z}_m \sum_{m=P+1}^Q (\mathbf{w}'^T \mathbf{z}_m - y_m)^2 \right. \\ &\quad \left. + \sum_{m=1}^P (\mathbf{w}'^T \mathbf{z}_m - y_m)^2 \sum_{m=P+1}^Q 2(\mathbf{w}'^T \mathbf{z}_m - y_m) \mathbf{z}_m \right] \end{aligned} \tag{24}$$

Let $\mathbf{Z}_+ = [z_{tm}]_{P \times T}$, $\mathbf{Z}_- = [z_{tm}]_{Q \times T}$, $\mathbf{y}_+ = [y_1, y_2, \dots, y_P]^T$, $\mathbf{y}_- = [y_1, y_2, \dots, y_Q]^T$, then

$$\begin{aligned} \nabla l(\mathbf{w}') &= \frac{2}{PQ} [\mathbf{Z}_+^T (\mathbf{Z}_+ \mathbf{w}' - \mathbf{y}_+) * (\mathbf{Z}_- \mathbf{w}' - \mathbf{y}_-)^T (\mathbf{Z}_- \mathbf{w}' - \mathbf{y}_-) \\ &\quad + (\mathbf{Z}_+ \mathbf{w}' - \mathbf{y}_+)^T (\mathbf{Z}_+ \mathbf{w}' - \mathbf{y}_+) * \mathbf{Z}_-^T (\mathbf{Z}_- \mathbf{w}' - \mathbf{y}_-)] \end{aligned} \tag{25}$$

where \mathbf{Z}_+ and \mathbf{Z}_- are the outputs of all sub-classifiers for positive and negative instances in the validation set, respectively. Meanwhile, \mathbf{y}_+ and \mathbf{y}_- denote the labels of these positive and negative instances, respectively. Setting Equation (25) to zero and using gradient descent, we identify the weight matrix \mathbf{w}' corresponding to the minimized loss function as

$$\mathbf{w}' = \mathbf{w}' - \eta \nabla l(\mathbf{w}') \tag{26}$$

where η is the learning rate, as illustrated in Step 3 of Algorithm 2. Based on the foundation established in Step 2, we use the training set \mathbf{D}'' to train, constituted by the outputs of all sub-classifiers on the validation set. Through gradient descent, we compute the optimal weight matrix $\mathbf{w}' = [w'_1, w'_2, \dots, w'_T]^T$. Conveniently, we term this class-imbalanced ensemble scheme, which considers the testing accuracy of each class, as the G-mean weighted ensemble. For multi-class data, we ascertain the weights of the sub-classifiers using a methodology analogous to binary classification. The output post-weighting of the sub-classifiers is

$$H(\mathbf{x}) = w'_1{}^T h_1(\mathbf{x}) + w'_2{}^T h_2(\mathbf{x}) + \dots + w'_T{}^T h_T(\mathbf{x}) \tag{27}$$

Within the G-mean weighted ensemble strategy, the outputs of the sub-classifiers on the validation set are treated as features for new training instances. Simultaneously, the labels of the validation instances are used as labels for these new instances, forming the new weight-training dataset \mathbf{D}'' . This dataset facilitates the training of ensemble weights for each sub-classifier.

Algorithm 2 Training process of B-TSK-FC

Input: Training set $\mathbf{D} = [(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)]^T$, where $x_n = [x_1, x_2, \dots, x_d]$, $n = 1, 2, \dots, N$ denotes the number of instances in the training set, and d is the total dimension of the instance. Validation set $\mathbf{D}' = [(x_1, y_1), (x_2, y_2), \dots, (x_M, y_M)]^T$, where $x_m = [x_1, x_2, \dots, x_d]$, $m = 1, 2, \dots, M$ is the number of instances in the validation set. For binary classification, $y_n \in \{0, +1\}$. For multi-class classification, y_n is encoded using one-hot encoding into a binary vector following [37]. Number of sub-classifiers, T .

Output: Fuzzy rule-improved fuzzy sub-classifier $h_t(x)$, B-TSK-FC broad ensemble classifier $H(x) = w_1^T h_1(x) + w_2^T h_2(x) + \dots + w_T^T h_T(x)$, where w_t^T is the ensemble weight of the t th TSK fuzzy sub-classifier, $t = 1, 2, \dots, T$.

Procedure:

Step 1 Using T and \mathbf{D} , invoke Algorithm 1 to generate T sub-classifiers post-fuzzy rule improvement. Let h_t denote the t th zero-order TSK fuzzy sub-classifier, which has finished fuzzy rule selection and weight optimization, where $t = 1, 2, \dots, T$. Algorithm 1 is denoted as \mathcal{L}_t . Execute the following iterative procedure.

for $t = 1$ *to* T *do*

$h_t = \mathcal{L}_t(\mathbf{D})$;

end for

Step 2 Use the validation set \mathbf{D}' to generate the training set \mathbf{D}'' for training the ensemble weightings. Initially set $\mathbf{D}'' = \emptyset$. Subsequently, execute the following iterative procedure.

for $m = 1$ *to* M *do*

for $t = 1$ *to* T *do*

$z_{tm} = h_t(x_m)$;

end for

$z_m = (z_{1m}, z_{2m}, \dots, z_{Tm})$

$\mathbf{D}'' = \mathbf{D}'' \cup (z_m, y_m)$;

end for

Step 3 Use the gradient descent method on \mathbf{D}'' to compute the ensemble weighting matrix for each sub-classifier, denoted as $\mathbf{w}' = [w_1', w_2', \dots, w_T']^T$, where each weight w_t' is the ensemble weight for the t th sub-classifier to facilitate the weighted ensemble of all sub-classifiers, with $t = 1, 2, \dots, T$.

Step 3.1 Define Loss Function.

Drawing inspiration from the G-mean metric, design a loss function $l(\mathbf{w}')$ using the mean square error for each instance class.

$$l(\mathbf{w}') = \frac{1}{P} \sum_{m=1}^P (\mathbf{w}'^T z_m - y_m)^2 \frac{1}{Q} \sum_{m=P+1}^Q (\mathbf{w}'^T z_m - y_m)^2 \tag{28}$$

where P and Q , respectively, denote the count of positive and negative instances in the validation set, such that $P + Q = M$. z_m is the predictive output for the m th instance in the validation set across all sub-classifiers, serving as the training feature set for the ensemble weights. To minimize this loss, compute the gradient of the loss function $l(\mathbf{w}')$.

$$\nabla l(\mathbf{w}') = \frac{2}{PQ} [Z_+^T (Z_+ \mathbf{w}' - \mathbf{y}_+) * (Z_- \mathbf{w}' - \mathbf{y}_-)^T (Z_- \mathbf{w}' - \mathbf{y}_-) + (Z_+ \mathbf{w}' - \mathbf{y}_+)^T (Z_+ \mathbf{w}' - \mathbf{y}_+) * Z_-^T (Z_- \mathbf{w}' - \mathbf{y}_-)] \tag{29}$$

where Z_+ and Z_- , respectively, denote the predicted outputs for all positive and negative instances across all sub-classifiers. Meanwhile, \mathbf{y}_+ and \mathbf{y}_- correspondingly represent the actual classes of all positive and negative instances.

Step 3.2 Set

$$\mathbf{w}' = \mathbf{w}' - \eta \nabla l(\mathbf{w}') \tag{30}$$

where η is the learning rate, thus deriving the optimal solution matrix \mathbf{w}' for the ensemble weights on data \mathbf{D}'' using the gradient descent method.

Step 3.3 Use the obtained weight matrix \mathbf{w}' to ensemble all TSK fuzzy sub-classifiers, obtaining the total output for the classifier post-ensemble as

$$H(x) = w_1^T h_1(x) + w_2^T h_2(x) + \dots + w_T^T h_T(x) \tag{31}$$

Step 4 Output $h_t(x)$, $H(x)$.

The reduction of fuzzy rules in the training process of TSK fuzzy sub-classifiers may lead to the weakness of the related fuzzy rules in the ensemble structure. The effect of a fuzzy rule in B-TSK-FC can be evaluated by its total output in the following Equation (32):

$$Y = \sum_{i=1}^T w'_i w_k^t a_k^t \phi_k^t(\mathbf{x}) \tag{32}$$

Equation (32) indicates that the effect of a fuzzy rule in B-TSK-FC is jointly determined by the weight w_k^t of the fuzzy rule and the weight w'_i of the TSK fuzzy sub-classifier. If a fuzzy rule is neglected in many TSK fuzzy sub-classifiers (corresponding to $w_k^t = 0$), it may impact the prediction of B-TSK-FC weakly. In other words, besides the interpretable fuzzy rules, B-TSK-FC is provided with two other understandable parameters, w'_i and w_k^t , which demonstrate the importance of the corresponding fuzzy rules.

Specifically, we initially used Algorithm 1 to generate T zero-order TSK fuzzy sub-classifiers improved through fuzzy rules based on the training set. In Step 2 of Algorithm 2, we used the outputs of the sub-classifiers on the validation set as features for new instances, and the validation set labels served as the labels for these new instances, culminating in the formation of a new training dataset \mathbf{D}'' . Subsequently, in Step 3, we deployed gradient descent on \mathbf{D}'' to calculate the ensemble weights of the sub-classifiers, accomplishing the system's weighted ensemble. The selection of the number of sub-classifiers T is contingent upon the best generalization performance on the testing set. Figure 2 delineates the training and prediction procedure of B-TSK-FC. We partitioned the original data into training, validation, and testing sets. The sub-classifiers, generated and improved based on the training set, predicted outputs on the validation set, serving as new input features, essentially representing the probabilities of predicted instance categories. Thereafter, employing the validation set labels as new instance labels, we trained the ensemble weights. Once trained, we used the ensemble classifier to predict on the testing set, deriving the final outcomes.

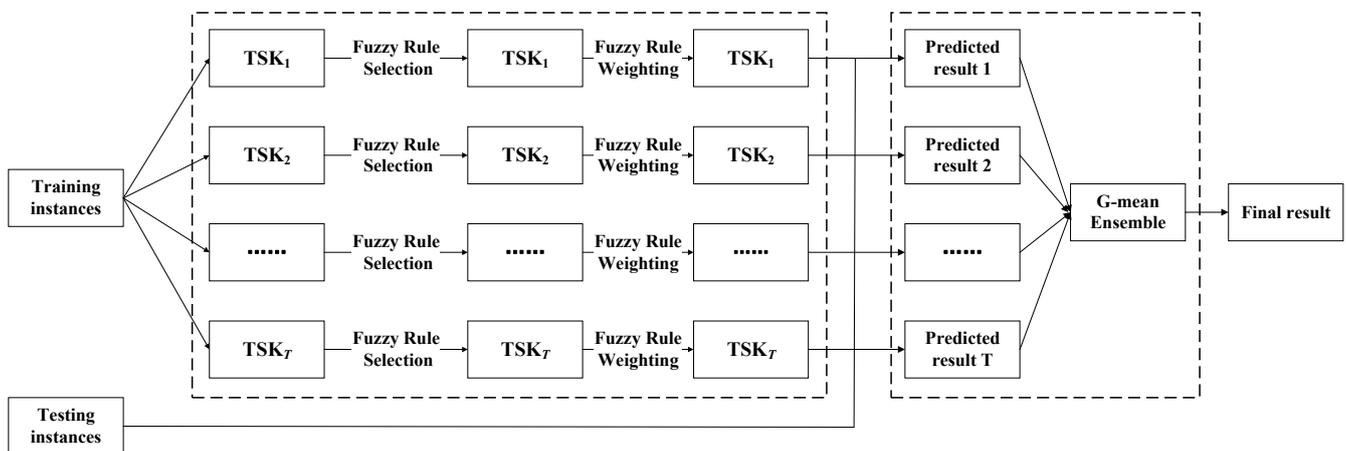


Figure 2. An illustrative architecture of B-TSK-FC.

The newly formulated loss function we have introduced places significant emphasis on the prediction accuracy of each class in imbalanced data, with a heightened focus on minority classes. This is instrumental in obtaining ensemble weights that ameliorate the performance of class-imbalanced learning, which, in turn, augments the classifier's performance in scenarios with imbalanced class distributions.

3.2. Theoretical Analysis and Proof of the Principle behind B-TSK-FC

This study presents the training process of the B-TSK-FC broad fuzzy classifier, which can be segmented into three phases: (1) Selection of fuzzy rules in the zero-order TSK

fuzzy classifier; (2) Weighted optimization of the fuzzy rules; and (3) Final G-mean weighted ensemble.

- (1) In the selection phase, the quality of fuzzy rules varies depending on the chosen centers of the antecedent parts. Some fuzzy rules align exceptionally well with particular data, while others do not. It is worth noting that fuzzy rules with larger antecedent part values typically produce higher values of the membership functions, suggesting the appropriate selection of the antecedent part centers. As a result, the constructed fuzzy rules better adhere to the original data distribution. The consequent part of a fuzzy rule is its weight; the larger its value, the closer its decision boundary is to the real data boundary. By affecting the output through a linear combination of antecedent parts and consequent parts (Equation (8)), the selection method proposed in this study optimizes the overall quality of fuzzy rules, reduces their number, enhances interpretability, and significantly boosts classification performance. This approach also tackles the fuzzy rule explosion issue induced by increasing data complexity, dynamically adapting to complex and variable data environments.
- (2) In the weighting phase, traditional zero-order TSK fuzzy classifiers underperform when dealing with imbalanced data. Hence, leveraging class-specific information, we generate a concise weight matrix that assigns higher weights to fuzzy rules encompassing minority class membership function knowledge, enhancing training efficacy. In contrast, fuzzy rules with majority class membership function information are assigned lower weights, which improves the capability to handle class imbalances.
- (3) At the ensemble stage, we define a new loss function inspired by the G-mean metric from imbalanced learning to compute the rational weights of each fuzzy sub-classifier in imbalanced scenarios. This G-mean weighted ensemble scheme effectively ameliorates prediction performance, precludes the negligence of minority classes, and mitigates overfitting risks.

Through the design of these three stages, the B-TSK-FC broad fuzzy classifier not only bolsters its capability to process imbalanced classes but also refines the overall classification accuracy and efficiency by meticulously adjusting fuzzy rules, making it more adaptable to complex and variable data environments.

In what follows, we will prove that the fuzzy rule selection strategy proposed in this study can significantly enhance the generalization performance of the fuzzy classifier. This proof is based on the testing set and uses the cross-entropy loss function for evaluation.

Taking a binary classification as an example, let the positive class label of an instance be $y = 1$ and the negative class label be $y = 0$. The output expression for the zero-order TSK fuzzy classifier is given by

$$y = \frac{\sum_{k=1}^K \mu^k(\mathbf{x}) a^k}{\sum_{r=1}^K \mu^r(\mathbf{x})} \quad (33)$$

where $\mu^k(\mathbf{x})$ denotes the antecedent part of the fuzzy rule for the testing instance under the k th fuzzy rule, and a^k is the consequent part of the k th fuzzy rule. The classifier output y resides in the $(0,1)$ range after normalization. Typically, this output is perceived as the probability of predicting the instance as the positive class ($y = 1$). Consequently, the probability of predicting the instance as the negative class ($y = 0$) can be represented by $(1 - y)$. If $y > 0.5$, it becomes straightforward to classify the input instance as the positive class; otherwise, it is classified as the negative class [41]. Equation (33) suggests that the larger the product of the antecedent part and the consequent part of the fuzzy rule, the closer the output y is to 1, implying a higher probability for the classifier to predict the instance as the positive class; conversely, the higher the likelihood of predicting it as the negative class.

Assume that before the selection of the fuzzy rule, the product of the antecedent part and the consequent part of the fuzzy rule is given by $\mu^{k1}(\mathbf{x}) a^{k1}$, leading to an output from the classifier as \hat{y}_p^1 . At this juncture, the probability of the classifier predicting the

label as $y = 1$ is \hat{y}_p^1 , and for the label $y = 0$, the probability is $1 - \hat{y}_p^1$. After the selection process, the product of the antecedent part and the consequent part of the fuzzy rule changes to $\mu^{k2}(\mathbf{x})a^{k2}$. As the selection is based on the value of this product, it follows that $\mu^{k2}(\mathbf{x})a^{k2} > \mu^{k1}(\mathbf{x})a^{k1}$. Consequently, the classifier output shifts to \hat{y}_p^2 . At this point, the classifier's probability for predicting the label as $y = 1$ is \hat{y}_p^2 , while for the label $y = 0$, it is $1 - \hat{y}_p^2$.

To evaluate performance, the cross-entropy loss function is employed, and its expression is

$$L = -\frac{1}{I} \sum_{i=1}^I [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{34}$$

where I denotes the number of testing set instances, y_i is the label of the i th testing instance (1 for the positive class and 0 for the negative class), and \hat{y}_i is the predicted probability, where \hat{y}_i belongs to the interval $(0, 1)$.

- (1) Assuming the label of the i th instance is $y_i = 1$ (i.e., the instance belongs to the positive class), the cross-entropy loss function becomes

$$L = -\log(\hat{y}_i) \tag{35}$$

Its derivative is

$$\nabla L = -\frac{1}{\hat{y}_i} < 0 \tag{36}$$

where $\hat{y}_i \in (0, 1)$. It is evident that the cross-entropy loss function is monotonically decreasing within this interval.

The classifier's probability for predicting the instance as the positive class is \hat{y}_i^1 before the fuzzy rule selection, and its output corresponds to \hat{y}_{ip}^1 . The loss for the classifier in this instance is

$$L^1 = -\log(\hat{y}_i^1) \tag{37}$$

After the reduction through fuzzy rule selection, the classifier's probability for predicting the instance as positive becomes \hat{y}_i^2 , leading to an output of \hat{y}_{ip}^2 . At this juncture, the classifier's loss for this instance is

$$L^2 = -\log(\hat{y}_i^2) \tag{38}$$

When the true label $y_i = 1$, we have

$$\hat{y}_i^1 = \hat{y}_{ip}^1 \tag{39}$$

$$\hat{y}_i^2 = \hat{y}_{ip}^2 \tag{40}$$

Furthermore, we understand that

$$\mu^{k2}(\mathbf{x})a^{k2} > \mu^{k1}(\mathbf{x})a^{k1} \tag{41}$$

referring to the output expression of the fuzzy classifier in Equation (33).

Thus,

$$\hat{y}_{ip}^2 > \hat{y}_{ip}^1 \tag{42}$$

It follows that

$$\hat{y}_i^2 > \hat{y}_i^1 \tag{43}$$

Given that the loss function is monotonically decreasing for $\hat{y}_i \in (0, 1)$, it leads to

$$L^2 < L^1 \tag{44}$$

which implies

$$L^2 - L^1 < 0 \tag{45}$$

This demonstrates that when the label $y_i = 1$, the classifier’s loss after fuzzy rule selection improvement is lower than before the selection.

- (2) Assuming the label of the i th instance is $y_i = 0$ (i.e., the instance belongs to the negative class), the cross-entropy loss function is expressed as

$$L = -\log(1 - \hat{y}_i) \tag{46}$$

Its derivative becomes

$$\nabla L = \frac{1}{1 - \hat{y}_i} > 0 \tag{47}$$

Given that \hat{y}_i lies within the interval $(0, 1)$, it is evident that the cross-entropy loss function is monotonically increasing within this range.

Before the fuzzy rule selection, the classifier’s probability of predicting the instance as the negative class is \hat{y}_i^1 , and the output of the classifier corresponds to \hat{y}_{ip}^1 . The loss incurred by the classifier for this instance at this juncture is

$$L^1 = -\log(\hat{y}_i^1) \tag{48}$$

The classifier’s probability of predicting the instance as the negative class becomes \hat{y}_i^2 after the fuzzy rule selection, with the corresponding classifier output being \hat{y}_{ip}^2 . Consequently, the classifier’s loss for this instance is

$$L^2 = -\log(\hat{y}_i^2) \tag{49}$$

In the context where the true label $y_i = 0$, we have

$$\hat{y}_i^1 = 1 - \hat{y}_{ip}^1 \tag{50}$$

$$\hat{y}_i^2 = 1 - \hat{y}_{ip}^2 \tag{51}$$

Moreover, the following holds true:

$$\mu^{k2}(\mathbf{x})a^{k2} > \mu^{k1}(\mathbf{x})a^{k1} \tag{52}$$

referencing the output formula of the fuzzy classifier, Equation (37).

It follows that

$$\hat{y}_{ip}^2 > \hat{y}_{ip}^1 \tag{53}$$

Substituting from Equations (50) and (51), we deduce

$$\hat{y}_i^2 < \hat{y}_i^1 \tag{54}$$

Given that the loss function is monotonically increasing for \hat{y}_i within the interval $(0, 1)$, it results in

$$L^2 < L^1 \tag{55}$$

which implies

$$L^2 - L^1 < 0 \tag{56}$$

This establishes that the loss of the classifier post-fuzzy rule selection improvement is less than its loss prior to the improvement when the label $y_i = 0$.

In summary, we have demonstrated that under the same number of fuzzy rules, the fuzzy rule selection improvement scheme proposed in this study effectively reduces the

classification loss of the fuzzy classifier, thereby significantly enhancing its generalization performance. This validates our hypothesis.

3.3. Complexity Analysis

The time complexity of the B-TSK-FC method proposed in this paper mainly comprises two parts: the first is the training time complexity of each fuzzy sub-classifier in Algorithm 1, and the second is overall time complexity during the ensemble phase in Algorithm 2.

In the first phase, assuming there are T sub-classifiers, the number of training instances for the t th sub-classifier is N . According to Equations (5), (7) and (8) in Algorithm 1, the time complexity at this stage can be inferred as $O(Nd^2K + K^3)$. Concurrently, the time complexity for fuzzy rule selection is $O(NK + K)$. Furthermore, the time complexity for fuzzy rule weighting is $O(3NK')$, and for calculating the consequent part from Equation (10), it is $O(K'^3)$. Hence, the total time complexity for each fuzzy sub-classifier can be expressed as $O(Nd^2K + K^3 + 3NK' + NK + K + K'^3)$, which simplifies to $O(Nd^2K + K^3 + K'^3)$. Here, K is the initial number of fuzzy rules, K' is the number of fuzzy rules after selection ($K' < K$), and d is the dimension of the instances. It is noteworthy that after fuzzy rule selection and weighting, the time complexity of the improved zero-order TSK fuzzy classifier only increases by $O(K'^3)$ compared to the original zero-order TSK fuzzy classifier $O(Nd^2K + K^3)$. This is mainly due to the improved techniques involving mostly low-dimensional matrix operations, signifying that our fuzzy rule improvement approach does not significantly increase the time complexity yet markedly enhances the performance of the fuzzy classifier to address imbalanced data. Following Algorithm 2, each sub-classifier has the same training time complexity. Therefore, the collective time complexity for all sub-classifiers is $O(\sum_{t=1}^T (Nd^2K + K^3 + K'^3))$. Adding the complexity of the weighted ensemble, which is $O(T)$, the resulting overall time complexity after the ensemble is $O(\sum_{t=1}^T (Nd^2K + K^3 + K'^3) + O(T)) = O(\sum_{t=1}^T (Nd^2K + K^3 + K'^3))$. In summary, the complete time complexity of the method is $O(\sum_{t=1}^T (Nd^2K + K^3 + K'^3))$.

4. Experimental Results

In this section, we conduct an in-depth experimental evaluation of the newly proposed B-TSK-FC method. The experimental data are sourced from benchmark datasets of KEEL [42] and UCI [43]. Moreover, we have also compared the B-TSK-FC method with five other leading imbalanced class classifiers and ensemble classifiers. The layout of this section is as follows: Sections 4.1–4.3 provide detailed descriptions of the datasets used in the experiment, the comparative methods, parameter settings, and evaluation criteria, respectively. In Section 4.4, we present and analyze the experimental results in detail. Finally, in Section 4.5, we conduct a statistical analysis of the results.

4.1. Datasets

To ensure the fairness and comprehensiveness of our study, we selected 15 benchmark datasets from KEEL [42] and UCI [43] to conduct a consistent evaluation of the B-TSK-FC method compared to other comparative methods. In order to thoroughly evaluate the classification performance of the proposed B-TSK-FC, fifteen datasets of various numbers of dimensionalities, classes, instances and imbalance ratios in a wide range are adopted in this experiment. The class imbalance ratios of these datasets range from 2 to 175, encompassing a wide spectrum from mildly imbalanced to highly imbalanced data environments. In addition, the dimensionality of the datasets varies from 3 to 241, with instance sizes ranging from 1000 to 240,000 and the number of classes ranging between 2 and 10. Hence, these datasets adequately represent imbalanced data scenarios across different dimensions, instance counts, and class counts. Table 2 provides a detailed list of the attributes of these 15 datasets, where IR denotes the imbalance ratio of the instances.

Table 2. Summary of 15 datasets.

#	Datasets	IR	No. of Instances	No. of Features	No. of Classes
1	penbased(PEN)	1.95	1100	16	10
2	marketing(MAR)	2.49	6877	13	9
3	turkiyestudentevaluationRspecific(TUR)	3.03	5820	33	5
4	DNA(DNA)	3.29	3186	180	2
5	skin(SKI)	3.82	245,057	3	2
6	usps(USP)	4.00	1500	241	2
7	musk(MUS)	5.49	6598	166	2
8	vowel0(VOW)	9.98	988	13	2
9	car-good(CAG)	24.04	1728	6	2
10	thyroid(THY)	40.16	7200	21	3
11	poker-8-9_vs_6(P96)	58.40	1485	10	2
12	shuttle-2_vs_5(SHU)	66.67	3316	9	2
13	poker-8_vs_6(P86)	85.88	1477	10	2
14	letterA(LET)	112.64	2000	21	2
15	page_blocks(PAB)	175.46	5473	10	5

4.2. Comparative Methods

We adopt seven class-imbalanced learning methods for comparison against the proposed method in this experiment. Since the B-TSK-FC method is premised on improving the zero-order TSK fuzzy system, we selected the original zero-order TSK fuzzy system as our initial comparative method, aiming to highlight the performance improvement of B-TSK-FC over the base TSK fuzzy classifier. However, considering that the original TSK fuzzy system was not specifically designed to handle class-imbalanced data, we first used the SMOTE [7] method to balance the imbalanced dataset and labeled this comparative method as SMOTE+TSK. The subsequent method for comparison is the Loss-Weighted TSK method (W-TSK), which is widely regarded as an effective method within the field of class-imbalanced learning. W-TSK builds upon the cost-sensitive approach, allocating distinct weights to the losses of minority and majority class instances, assigning higher weights to the losses of minority class instances and correspondingly reducing the weights for majority class instances, thus effectively enhancing the attention paid to the minority class. We also chose the classical K-Nearest Neighbors method, called KNN, as the third comparative method. Since KNN does not inherently address class imbalance, we again used the SMOTE method for data preprocessing and called this method SMOTE+KNN.

Considering the ensemble learning strategy adopted by the B-TSK-FC method, we further selected four methods renowned for their robustness and superior performance in the imbalanced ensemble field: RUSBoost [18], OverBoost [23], SMOTEBagging [19], and SMOTEBoost [20]. The RUSBoost method combines under-sampling strategies with the Boosting ensemble method, establishing itself as a hallmark method for imbalanced ensemble learning. Similarly, the OverBoost method integrates over-sampling with the Boosting ensemble, using random over-sampling techniques to balance data and further harnessing the potent capabilities of Boosting, thereby offering notable robustness and accuracy when handling class-imbalanced data. SMOTEBagging is a combination of SMOTE and Bagging (Bootstrap Aggregating) [7]. Before each sub-sample training, it first uses the SMOTE method to increase minority class instances and then trains the base classifier on this expanded dataset. This method aims to enhance the base classifier's ability to identify minority classes, thereby making the entire bagging ensemble perform better in the face of class imbalance problems. SMOTEBoost is a combination of SMOTE and Boosting. In

each round of Boosting, SMOTEBoost first uses the SMOTE method to oversample the currently misclassified minority class instances and then trains a new base classifier on this expanded dataset. Compared with traditional Boosting methods, this method not only focuses on misclassified instances but also particularly emphasizes the importance of minority classes. In this way, with the iteration of Boosting, the model's sensitivity to minority classes will gradually increase. SMOTEBagging and SMOTEBoost are strategies that combine SMOTE with ensemble learning methods to improve the classification model's ability to identify minority classes. They are all the most advanced and robust methods in the field of class imbalance learning. Therefore, we used these two ensemble methods combined with SMOTE as comparative methods

4.3. Parameter Settings and Evaluation Metrics

To ensure the comparability of the experimental results, we used a grid search approach to find the optimal parameter combinations for the B-TSK-FC method and its comparative methods across all datasets. Within the B-TSK-FC method, there are three primary parameters: the center μ of the Gaussian function in the antecedent part of the TSK fuzzy system, the number of fuzzy rules K in the TSK fuzzy classifier, and the number T of TSK sub-classifiers in the ensemble. Firstly, for the center value μ of the Gaussian membership function, we constrained it to randomly select from the range [0, 0.25, 0.5, 0.75, 1] drawing on recommendations from [44]. This configuration also ensures the language interpretability of the TSK fuzzy classifier, e.g., interpretations such as very bad, bad, medium, good, and very good. Secondly, for the number of fuzzy rules K in each TSK fuzzy sub-classifier, we searched for the optimal fuzzy rule count in the range of 5 to 500 in increments of 5. This approach ensures a thorough exploration of the parameter space to identify the best fuzzy rule parameter. Lastly, for the number T of TSK fuzzy sub-classifiers in the B-TSK-FC ensemble system, we also adopted a search range of 5 to 500, with increments of 5, until we found the optimal parameter settings under the G-mean evaluation metric. Such parameter-setting strategies are designed to ensure exhaustive exploration of the parameter space while identifying the parameter combination that achieves optimal performance. Detailed parameter settings for B-TSK-FC can be found in Table 3.

Table 3. Parameter settings for the B-TSK-FC method.

Parameters	Ranges and Intervals
μ : Center value of the Gaussian membership function	[0, 0.25, 0.5, 0.75, 1]
K : Number of fuzzy rules for the TSK fuzzy sub-classifier	5:5:500
T : Number of sub-classifiers in the ensemble	5:5:500

Table 4 provides the parameter settings for the comparative methods used in the experiments. In this context, μ denotes the center value of the Gaussian membership function for the antecedent part of the fuzzy system, while K is the number of fuzzy rules in the TSK fuzzy classifier. To maintain experimental fairness, we set the parameters for the SMOTE+TSK comparative method based on the default values listed in Table 4. Similarly, we determined the center μ of the Gaussian membership function and the fuzzy rule number K for the TSK fuzzy system by a grid search approach, ensuring that the search range and step size matched the settings for the μ parameter and fuzzy rule number K in the B-TSK-FC method as presented in Table 3. For the Weighted TSK method (W-TSK), the parameter settings for the components μ and K were aligned with those of the B-TSK-FC method. As for the SMOTE+KNN comparative method, its principal parameter was the number of neighbors in the KNN method. We conducted a search within the range of 2 to 200 with a step size of 1 to fully exploit its performance potential. Regarding the ensemble methods designed for class imbalance, RUSBoost and OverBoost, we adhered to the general parameter settings provided in Table 4. Consistent with the B-TSK-FC method, we looked for the optimal ensemble size of sub-classifiers in the range of 2 to 500, with

a step size of 1, using a grid search approach to derive the best parameter values. This setup was implemented to guarantee the reliability and impartiality of the experimental results. Through these rigorous parameter settings and optimization processes, we ensured that the experimental results for all methods and datasets accurately reflected performance under the optimal hyperparameter combinations. The parameter settings of the last two ensemble comparative methods, SMOTEBagging and SMOTEBoost, are listed in detail in Table 4. The number of their sub-classifiers also uses the grid search method to range from 5 to 500 with a step size of 5 and is searched step by step.

For learning from class-imbalanced data, traditional overall accuracy metrics often fail to comprehensively reflect the method’s predictive performance for the minority class. Hence, we opted for the G-mean as the primary metric for evaluating the performance of each method in this study. The G-mean effectively gauges the balanced accuracy of predictions across classes in the context of class imbalance. Thus, using it as a criterion provides a more objective and comprehensive reflection of the performance of various methods in solving the class imbalance issue. Additionally, the confusion matrix defines various metrics for evaluating method performance, with detailed information provided in Table 5. The specific formula for G-mean is as follows:

$$G - \text{mean} = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} \tag{57}$$

Table 4. Parameter settings of the comparative methods.

Approaches	Default Values of Parameters	Ranges and Intervals of Parameters
SMOTE+TSK	sampling_strategy = ‘auto’, random_state = None, k_neighbors = 5	μ : [0, 0.25, 0.5, 0.75, 1] K: 5:5:500
W-TSK	-	μ : [0, 0.25, 0.5, 0.75, 1] K: 5:5:500
SMOTE+KNN	sampling_strategy = ‘auto’, random_state = None, k_neighbors = 5(KNN)	k_neighbors(KNN): 2:1:100
RUSBoost	learning_rate = 1.0, random_state = None	n_estimators: 5:5:500
OverBoost	random_state = None, k_neighbors = 5, early_termination; False	n_estimators: 5:5:500
SMOTEBagging	random_state = None, k_neighbors = 5, sampling_strateg = ‘auto’	n_estimators: 5:5:500
SMOTEBoost	random_state = None, learning_rate = 1.0, k_neighbors = 5	n_estimators: 5:5:500

Table 5. Confusion matrix for binary classification.

True Condition	Predicted Result	
	Positive	Negative
Positive	TP (True Positive)	FN (False Negative)
Negative	FP (False Positive)	TN (True Negative)

4.4. Comparative Experimental Study

To thoroughly evaluate the performance of the B-TSK-FC method for class-imbalanced learning proposed in this research, we designed and conducted a series of comparative experiments. In these experiments, we selected two class-imbalanced learning methods based on the TSK fuzzy system (i.e., SMOTE+TSK and W-TSK), one SMOTE+KNN method using the K-nearest neighbors approach, and four advanced methods in the class-imbalanced ensemble learning field (RUSBoost, OverBoost, SMOTEBagging, and SMOTEBoost) as comparative methods. Since the proposed B-TSK-FC is constructed through an ensemble manner, the ensemble methods RUSBoost, OverBoost, SMOTEBagging, and SMOTEBoost are adopted. TSK fuzzy classifiers SMOTE+TSK and W-TSK, which tackle imbalanced data by different means from B-TSK-FC, are adopted to evaluate the efficiency of the methodology of B-TSK-FC. RusBoost, OverBoost, and SMOTE+KNN are the state-of-the-art or typical methods; thus, the experimental comparison between B-TSK-FC and them may be helpful for fair evaluation. Following the parameter-setting strategies mentioned above, we optimized each parameter using grid search to ensure fairness in experimentation. All experiments were conducted on 15 benchmark datasets sourced from KEEL and UCI using G-mean as the evaluation metric. Following our parameter optimization strategy, we executed the B-TSK-FC method and the seven comparative methods ten times each on every dataset. Subsequently, we recorded the G-mean scores of each method on the training and testing sets and calculated the average of the ten runs for each method on the respective training and testing sets, representing the final G-mean score. Table 6 presents the detailed experimental results, where K denotes the fuzzy rule count in the TSK fuzzy system, and T is the number of sub-classifiers in the ensemble learning method. The best performance of the methods carried out on each dataset is marked in bold in Table 6. To offer a more intuitive performance comparison, we visualized the average final G-mean scores of B-TSK-FC and the comparative methods on each testing dataset in a bar chart format (as illustrated in Figure 3). This representation vividly highlights the performance advantage of the B-TSK-FC method over other methods across various datasets.

Table 6. Average number of fuzzy rules, number of sub-classifiers in the ensemble framework, average G-mean value, and standard deviation for TSK classifier classification performance on benchmark datasets.

DAS	B-TSK-FC			SMOTE+TSK			W-TSK			SMOTE+KNN			RusBoost			OverBoost			SMOTEBagging			SMOTEBoost												
	Training ± Std	Testing ± Std		K	T	Training ± Std	Testing ± Std	K	T	Training ± Std	Testing ± Std	K	T	Training ± Std	Testing ± Std	K	T	Training ± Std	Testing ± Std	K	T	Training ± Std	Testing ± Std	K	T									
PEN	0.9714 ± 0.0001	0.9811 ± 0.0001		230	10	0.9517 ± 0.0000	0.9434 ± 0.0003	200		0.9774 ± 0.0000	0.9672 ± 0.0001	300		0.9778 ± 0.0000	0.9612 ± 0.0001			0.7791 ± 0.0006	0.7209 ± 0.0015			350		0.8810 ± 0.0003	0.8804 ± 0.0005	55		1.0000 ± 0.0000	0.9783 ± 0.0000	500		0.1408 ± 0.0470	0.1364 ± 0.0443	100
MAR	0.2455 ± 0.0001	0.2518 ± 0.0001		180	10	0.2731 ± 0.0001	0.2227 ± 0.0003	400		0.3143 ± 0.0000	0.2306 ± 0.0001	200		0.4155 ± 0.0000	0.2274 ± 0.0001			0.2205 ± 0.0002	0.1977 ± 0.0005			170		0.2523 ± 0.0000	0.2353 ± 0.0002	90		0.9133 ± 0.0000	0.2324 ± 0.0000	50		0.2436 ± 0.0086	0.2264 ± 0.0124	25
TUR	0.8706 ± 0.0001	0.8297 ± 0.0002		80	50	0.8579 ± 0.0000	0.8170 ± 0.0001	250		0.8678 ± 0.0000	0.8178 ± 0.0001	275		0.8601 ± 0.0000	0.8272 ± 0.0001			0.8234 ± 0.0009	0.8241 ± 0.0012			300		0.8485 ± 0.0001	0.8294 ± 0.0002	250		0.9990 ± 0.0000	0.8448 ± 0.0001	25		0.8468 ± 0.0095	0.8432 ± 0.0114	30
DNA	0.8206 ± 0.0002	0.7487 ± 0.0003		360	30	0.7614 ± 0.0000	0.6300 ± 0.0004	450		0.7793 ± 0.0001	0.6223 ± 0.0007	450		0.4978 ± 0.0000	0.3622 ± 0.0003			0.8409 ± 0.0000	0.7994 ± 0.0002			100		0.8526 ± 0.0000	0.8294 ± 0.0002	40		0.9740 ± 0.0000	0.7121 ± 0.0005	5		0.8303 ± 0.0092	0.8200 ± 0.0137	5
SKI	0.9793 ± 0.0000	0.9814 ± 0.0000		200	10	0.9669 ± 0.0000	0.9665 ± 0.0000	10		0.9670 ± 0.0000	0.9672 ± 0.0000	19		0.9791 ± 0.0000	0.9782 ± 0.0000			0.9652 ± 0.0001	0.9629 ± 0.0001			24		0.9643 ± 0.0000	0.9614 ± 0.0001	18		0.9999 ± 0.0000	0.9993 ± 0.0000	20		0.9438 ± 0.0007	0.9434 ± 0.0014	10
USP	0.9569 ± 0.0001	0.9444 ± 0.0003		200	10	0.9536 ± 0.0001	0.8938 ± 0.0006	240		0.9481 ± 0.0000	0.8710 ± 0.0014	325		0.9468 ± 0.0000	0.9467 ± 0.0004			0.8795 ± 0.0001	0.8293 ± 0.0005			20		0.9479 ± 0.0001	0.9371 ± 0.0002	10		1.0000 ± 0.0000	0.8550 ± 0.0004	500		0.9996 ± 0.0007	0.8251 ± 0.0008	100
MUS	0.9696 ± 0.0000	0.9653 ± 0.0000		480	50	0.9582 ± 0.0000	0.9396 ± 0.0001	600		0.9579 ± 0.0000	0.9464 ± 0.0001	500		0.9687 ± 0.0000	0.9311 ± 0.0000			0.9649 ± 0.0000	0.9385 ± 0.0002			80		0.9559 ± 0.0000	0.9250 ± 0.0001	50		0.9995 ± 0.0000	0.9355 ± 0.0002	25		0.9999 ± 0.0000	0.9601 ± 0.0000	250
VOW	0.9979 ± 0.0000	0.9969 ± 0.0000		180	300	0.9811 ± 0.0000	0.9794 ± 0.0001	90		0.9822 ± 0.0001	0.9795 ± 0.0000	85		0.9890 ± 0.0000	0.9868 ± 0.0000			0.9650 ± 0.0025	0.9386 ± 0.0025			29		0.9944 ± 0.0000	0.9529 ± 0.0006	10		0.9958 ± 0.0000	0.9702 ± 0.0008	5		1.0000 ± 0.0000	0.9789 ± 0.0168	80
CAG	0.9832 ± 0.0000	0.9854 ± 0.0000		80	150	0.9247 ± 0.0000	0.9117 ± 0.0009	170		0.9210 ± 0.0000	0.9203 ± 0.0001	130		0.9589 ± 0.0000	0.9361 ± 0.0003			0.8671 ± 0.0093	0.8842 ± 0.0059			475		0.9625 ± 0.0000	0.9612 ± 0.0000	25		1.0000 ± 0.0000	0.8908 ± 0.0036	20		0.9572 ± 0.0153	0.9049 ± 0.0773	30
THY	0.7511 ± 0.0006	0.7580 ± 0.0004		250	10	0.7450 ± 0.0000	0.7227 ± 0.0007	475		0.7571 ± 0.0000	0.7291 ± 0.0004	300		0.7263 ± 0.0000	0.7034 ± 0.0006			0.8442 ± 0.0366	0.8097 ± 0.0469			15		0.9910 ± 0.0000	0.9897 ± 0.0000	50		0.9978 ± 0.0000	0.9820 ± 0.0001	5		0.9925 ± 0.0020	0.9916 ± 0.0029	5
P96	0.9776 ± 0.0008	0.9573 ± 0.0031		130	10	0.9529 ± 0.0010	0.8928 ± 0.0083	475		0.9770 ± 0.0004	0.8875 ± 0.0061	180		0.9045 ± 0.0000	0.8960 ± 0.0002			0.6709 ± 0.0009	0.4454 ± 0.0312			17		0.9483 ± 0.0003	0.4573 ± 0.0279	375		0.9295 ± 0.0021	0.4812 ± 0.0468	5		0.6205 ± 0.0544	0.2446 ± 0.2092	20
SHU	0.9991 ± 0.0000	0.9984 ± 0.0000		160	10	0.9693 ± 0.0002	0.9689 ± 0.0028	250		0.9622 ± 0.0000	0.9831 ± 0.0007	180		0.9980 ± 0.0000	0.9965 ± 0.0000			0.9982 ± 0.0000	0.9810 ± 0.0024			20		1.0000 ± 0.0000	0.9827 ± 0.0027	50		1.0000 ± 0.0000	1.0000 ± 0.0000	5		1.0000 ± 0.0000	1.0000 ± 0.0000	5

Table 6. Cont.

DAS	B-TSK-FC		SMOTE+TSK		W-TSK		SMOTE+KNN		RusBoost		OverBoost		SMOTEBagging		SMOTEBoost	
	Training ± Std Testing ± Std	K T	Training ± Std Testing ± Std	K	Training ± Std Testing ± Std	K	Training ± Std Testing ± Std	K	Training ± Std Testing ± Std	T						
P86	0.9887 ± 0.0001 0.9595 ± 0.0029	120 10	0.9811 ± 0.0000 0.9373 ± 0.0034	200	0.9811 ± 0.0001 0.8519 ± 0.0135	170	0.9667 ± 0.0000 0.9045 ± 0.0085		0.6304 ± 0.0195 0.4575 ± 0.0772	22	0.9746 ± 0.0001 0.3647 ± 0.0416	100	0.8858 ± 0.0047 0.0947 ± 0.0360	5	0.6149 ± 0.0617 0.4122 ± 0.2328	5
LET	0.9492 ± 0.0000 0.9519 ± 0.0000	200 10	0.9541 ± 0.0000 0.9437 ± 0.0001	250	0.9554 ± 0.0000 0.9387 ± 0.0002	275	0.9468 ± 0.0000 0.9055 ± 0.0003		0.8895 ± 0.0083 0.8813 ± 0.0062	15	0.9888 ± 0.0000 0.9218 ± 0.0007	50	0.9579 ± 0.0001 0.7320 ± 0.0013	5	0.9455 ± 0.0122 0.9229 ± 0.0197	10
PAB	0.8498 ± 0.0010 0.8543 ± 0.0013	140 200	0.8263 ± 0.0002 0.8092 ± 0.0014	25	0.8369 ± 0.0002 0.7976 ± 0.0025	40	0.8150 ± 0.0001 0.8039 ± 0.0006		0.8206 ± 0.0051 0.7951 ± 0.0033	300	0.3754 ± 0.1055 0.3912 ± 0.1154	375	0.9942 ± 0.0000 0.8306 ± 0.0020	300	0.3818 ± 0.2045 0.2936 ± 0.2520	60

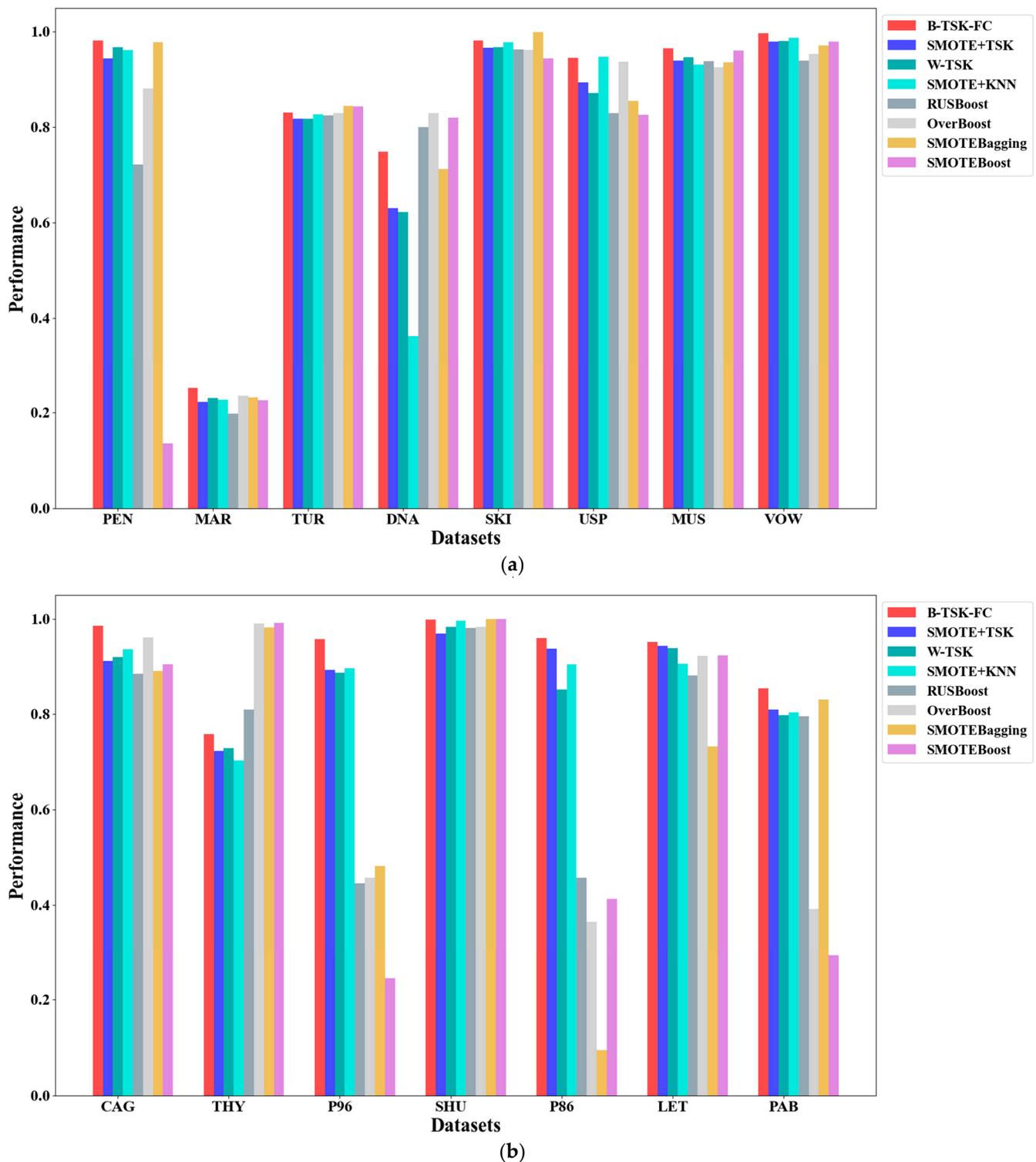


Figure 3. Comparative testing performance of B-TSK-FC and comparative methods across the fifteen datasets, in which, Subfigure (a) includes the comparison results of eight data sets, and subfigure (b) includes the comparison results of seven data sets.

From the experimental results presented in Table 6 and Figure 3, we can draw the following conclusions:

- (1) On the majority of the datasets, the proposed B-TSK-FC method exhibits superior generalization performance. Particularly on datasets such as *PEN*, *MAR*, *MUS*, *VOW*, *CAG*, *P96*, *P86*, *LET*, and *PAB*, B-TSK-FC significantly outperforms the comparative

- methods, indicating its exceptional effectiveness in handling class-imbalanced scenarios.
- (2) For certain datasets, including *TUR*, *DNA*, and *THY*, the performance of B-TSK-FC is comparable to some of the comparative methods yet remains competitive. This suggests that the B-TSK-FC method maintains consistent generalization performance across various datasets.
 - (3) On some datasets, especially those with higher imbalance ratios, B-TSK-FC achieves the best results. This demonstrates that in specific settings, such as highly imbalanced complex datasets, B-TSK-FC can provide superior classification outcomes. This also indicates that our designed mechanism for the selection of fuzzy rules is not overly dependent on data distribution but can obtain higher quality fuzzy rules accordingly based on the specific data distribution, thereby efficiently tackling diverse and complex datasets.
 - (4) Please note that some comparative methods achieve much lower testing accuracies than training accuracies when carried out on some datasets, e.g., SMOTEBagging carried out on the datasets *MAR*, *TUR*, *DNA*, *USP*, *P96*, *P86*, *LET*, *PAB*, and SMOTEboost implemented on the datasets *USP*, *CAG*, *P96*, *P86*, and *PAB*. The big margin between testing accuracies and training accuracies indicates that the method has seriously overfitted the training instances and lost the generalization capability for the testing instances. In contrast, the testing accuracies and the training accuracies of B-TSK-FC are much closer than that of the comparative methods, which demonstrates the advantage of the generalization capability of B-TSK-FC.

In summary, the experimental results robustly affirm the superior and consistent performance of the B-TSK-FC method when dealing with most imbalanced datasets. This provides a testament to the efficacy of our designed fuzzy rule weighting technique for imbalanced data. While the method may slightly underperform on some datasets, we aim to further optimize it in future research to enhance its performance. Additionally, the results emphasize the effectiveness of selecting fuzzy rules based on the specific characteristics of datasets in practical applications.

From another perspective, the results from Table 6 also reveal that, compared to the advanced methods, our B-TSK-FC method has a smaller number of fuzzy rules and ensemble classifiers. The former implies that B-TSK-FC ensures superior generalization while retaining good linguistic interpretability. This is attributed to our reduction of fuzzy rules, enhancing the overall quality of the fuzzy rules by refining them. The latter indicates that, in comparison to conventional imbalanced ensemble learning methods, B-TSK-FC achieves better classification with fewer classifiers, significantly reducing the method's overall complexity and consequently saving time and memory resources.

To more explicitly highlight the advantages of our proposed B-TSK-FC method in performance compared to both individual class-imbalanced processing methods and ensemble imbalance processing techniques, we categorize the comparative methods into two groups. The first group encompasses the SMOTE+TSK, W-TSK, and SMOTE+KNN methods, representing individual class imbalance processing techniques. The second group includes the RUSBoost, OverBoost, SMOTEBagging, and SMOTEBoost methods, representing ensemble imbalance processing techniques. Based on the average G-mean experimental results of each method across different testing datasets in Table 6, we calculate the percentage performance improvement of B-TSK-FC over each comparative method for all 15 testing datasets. Subsequently, we determine the average percentage improvement for each category of methods. The related statistical results are presented in Table 7.

From the analysis of Table 7, it is evident that whether compared to individual class-imbalanced processing methods or ensemble imbalance processing techniques, our proposed B-TSK-FC method achieves a notable performance boost in addressing class imbalance issues. This further underscores the comprehensive superiority and efficiency of the B-TSK-FC method when tackling imbalanced data problems.

Table 7. Average percentage improvement in generalization performance of B-TSK-FC method compared to individual and ensemble imbalanced methods across different datasets.

Method Type	Performance Improvement Percentage (%)
Single-Class-Imbalanced Method	5.44
Class-Imbalanced Ensemble Method	15.48

4.5. Statistical Test

In this subsection, we conducted a statistical test on the performance of eight class-imbalanced learning methods, including our proposed B-TSK-FC method and seven advanced existing imbalanced learning methods, including SMOTE+TSK, W-TSK, SMOTE+KNN, RUSBoost, OverBoost, SMOTEBagging, and SMOTEBoost. The experimental data comprised 15 benchmark imbalanced datasets sourced from KEEL [42] and UCI [43]. To assess the performance of these methods, we conducted the Friedman test.

The Friedman test [45] is a classic non-parametric test designed to determine whether significant differences exist among multiple variables. The test procedure is as follows.

Firstly, using the experimental results from Table 6 related to the average G-mean metric, we compute the rank $\{R_v^u\}_{U \times V}$ for each method on every dataset. Here, $\{R_v^u\}_{U \times V}$ is the rank of the u th method on the v th dataset, where $u \in \{1, \dots, U\}$, $v \in \{1, \dots, V\}$. U is the number of methods, while V is the count of datasets used for experimentation. The test statistic, Ω , is then computed as

$$\Omega = \frac{12V}{U(U+1)} \left[\sum_u \bar{R}_u^2 - \frac{U(U+1)^2}{4} \right] \tag{58}$$

where the average rank \bar{R}_u is given by

$$\bar{R}_u = 1/V \sum_{v=1}^V R_u^v \tag{59}$$

When $U > 4$ and $V > 15$, the distribution of Ω approximates a chi-squared χ^2 distribution with $U - 1$ degrees of freedom. Setting the significance level, $\alpha = 0.05$, the p -value is calculated as

$$p = P(\chi_{\alpha[U-1]}^2 \geq \Omega) \tag{60}$$

Should the p -value be less than the predetermined significance level α , the null hypothesis asserting no significant difference is rejected. In such cases, the significant difference is validated. Typically, multiple post hoc tests are conducted to compare the control method with other comparative methods. This encompasses tests including Hochberg, Holm, Hommel, and adjusted p -value tests.

In this study, we conducted experiments on 15 benchmark class-imbalanced datasets from UCI and KEEL repositories to evaluate the performance of our proposed B-TSK-FC method in comparison to state-of-the-art comparative methods. These comparative methods include SMOTE+TSK, W-TSK, SMOTE+KNN, RUSBoost, OverBoost, SMOTEBagging, and SMOTEBoost. For statistical validation, the Friedman test was used. Table 8 presents the average ranking results of the B-TSK-FC method based on the Friedman test. The Friedman statistic, which follows a chi-squared distribution with 7 degrees of freedom, was found to be 26.694444. The p -value obtained from the Friedman test was 0.000378.

The Friedman statistic was recorded at 26.694444 and corresponds to a chi-squared distribution with 7 degrees of freedom. The p -value obtained from the Friedman test was 0.000378. This result rejects the null hypothesis, suggesting that there is a significant difference in performance between at least one method and the others. As can be clearly observed from Table 8, the B-TSK-FC method proposed in this study is ranked first.

Table 8. Average rankings of the methods (Friedman).

Method	Ranking
B-TSK-FC	1.8667
SMOTE+TSK	4.8667
W-TSK	4.4667
SMOTE+KNN	4.3333
RUSBoost	6.2667
OverBoost	4.8667
SMOTEBagging	4.3
SMOTEBoost	5.0333

To delve deeper into the performance differences between the methods, we conducted post hoc comparisons using the Friedman test. At a significance level of $\alpha = 0.05$, we present Table 9, which showcases the p -values obtained by applying post hoc methods over the results of the Friedman procedure.

Table 9. Post hoc comparison table for $\alpha = 0.05$ (Friedman).

i	Method	$z=(R_0 - R_i)/SE$	p	Holm Hommel
7	RUSBoost	4.91935	0.000001	0.007143
6	SMOTEBoost	3.540441	0.000399	0.008333
5	SMOTE+TSK	3.354102	0.000796	0.01
4	OverBoost	3.354102	0.000796	0.0125
3	W-TSK	2.906888	0.00365	0.016667
2	SMOTE+KNN	2.757817	0.005819	0.025
1	SMOTEBagging	2.720549	0.006517	0.05

The outcomes of the post hoc tests using the Holm–Hommel procedure lead to the rejection of all null hypotheses and indicate significant differences between B-TSK-FC and the seven comparative methods. Subsequently, to control the family-wise error rate, we adjusted the original p -values. Table 10 details the adjusted p -values acquired through the application of the post hoc methods based on the Friedman test.

Table 10. Adjusted p -values (Friedman).

i	Method	Unadjusted p	p_{Holm}	p_{Hommel}
1	RUSBoost	0.000001	0.000006	0.000006
2	SMOTEBoost	0.000399	0.002397	0.001991
3	SMOTE+TSK	0.000796	0.003981	0.003185
4	OverBoost	0.000796	0.003981	0.003185
5	W-TSK	0.00365	0.010951	0.006517
6	SMOTE+KNN	0.005819	0.011638	0.006517
7	SMOTEBagging	0.006517	0.011638	0.006517

The adjusted p -values in Table 10 further reinforce the significant superiority of the proposed B-TSK-FC in the field of classification performance to the seven comparative methods. In summary, based on the statistical insights garnered from the Friedman test and the subsequent post hoc evaluations, we deduce that our proposed B-TSK-FC method

demonstrates pronounced superiority in addressing class-imbalanced datasets relative to current advanced methods.

5. Conclusions

In this study, we propose a novel approach for handling class imbalance issues based on the zero-order TSK fuzzy classifier, wherein the fuzzy rules have a reduction improvement process. Adapting to the dynamic changes in data scenarios, we select pertinent fuzzy rules. By acknowledging the differences in the antecedent parts of the fuzzy rules generated by different class instances, we apply varied weights accordingly. With these two steps, we have constructed an array of enhanced zero-order TSK fuzzy sub-classifiers. Lastly, by defining the mean squared loss product of various class instances as the learning loss function, we deduce the output weighting coefficients for each sub-classifier, hence achieving a dynamically weighted ensemble output.

The approach proposed in this paper not only refines the overall quality of the fuzzy rules, thereby augmenting the intrinsic classification performance of the fuzzy classifier, but also ensures commendable linguistic interpretability by reducing the number of fuzzy rules. Furthermore, by selecting fuzzy rules based on specific data scenarios, we have endowed the classifier with the performance to cater to more complex and changeable data. Our proposed weighted fuzzy rule scheme considerably elevates the fuzzy classifier's efficacy in handling imbalanced data. Additionally, our introduction of a weighted ensemble strategy, which guarantees the prediction precision of each class, further augments the overall ensemble system's classification generalization on imbalanced data. Experimental outcomes on 15 benchmark class-imbalanced datasets manifest that the performance of our proposed B-TSK-FC method surpasses other avant-garde popular comparative methods. These results emphatically validate the effectiveness and superiority of our approach in grappling with class imbalance challenges.

Nevertheless, despite the significant advancements achieved in our research, there remain copious potential avenues for future exploration. Initially, our methodology for optimizing fuzzy rules possesses further scope for refinement to cater to more sophisticated datasets and severe class imbalance scenarios. Moreover, our approach can be broadened to other types of fuzzy classifiers or distinct machine learning models to further optimize their performance in solving class imbalance issues. Finally, amalgamating our technique with other pre-existing solutions for class imbalances could pave the way for a more robust and flexible strategy.

Author Contributions: Conceptualization, H.Y. and B.Q.; methodology, B.Q.; software, J.Z. (Jinghong Zhang); validation, J.Z. (Jinghong Zhang) and B.L.; formal analysis, J.Z. (Jinghong Zhang); investigation, J.Z. (Jinghong Zhang); resources, H.C.; data curation, Y.L.; writing—original draft preparation, J.Z. (Jinghong Zhang); writing—review and editing, B.Q.; visualization, J.Z. (Jie Zhou); supervision, J.Z. (Jie Zhou); project administration, B.Q. and H.Y.; funding acquisition, H.Y., B.Q. and J.Z. (Jie Zhou). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation (NNSF) of China under Grant (62376109, 62176107, 62101645), Zhejiang Provincial Natural Science Foundation of China under Grant (LQ22F020024), and the Postgraduate Research and Practice Innovation Program of Jiangsu Province of China under Grant No. SJCX23_2124.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Introduction of the Abbreviations.

Abbreviation	Full Form
TSK	Takagi–Sugeno–Kang
IR	Imbalanced Ratio
diag	Diagonal Matrix
G-mean	Geometric Mean
ROS	Random Over-Sampling
RUS	Random Under-Sampling
SMOTE	Synthetic Minority Over-sampling Technique
LLM	Lazy Learning Machine
KEEL	Knowledge Extraction based on Evolutionary Learning
UCI	University of California, Irvine
W-TSK	Loss-Weighted TSK
KNN	K-Nearest Neighbors
RUSBoost	Random Under-Sampling Boosting
OverBoost	Over-Sampling Boosting
SMOTEBagging	Synthetic Minority Over-Sampling Technique Bootstrap Aggregating
SMOTEBoost	Synthetic Minority Over-Sampling Technique Boosting

References

- Chawla, N.V.; Japkowicz, N.; Kolcz, A. Editorial: Special issue on learning from imbalanced datasets. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 1–6. [[CrossRef](#)]
- He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284.
- Xu, L.; Chow, M.; Taylor, L. Power distribution fault cause identification with imbalanced data using the data mining-based fuzzy classification e-method. *IEEE Trans. Power Syst.* **2007**, *22*, 164–171. [[CrossRef](#)]
- Verbraken, T.; Verbeke, W.; Baesens, B. A novel profit maximizing metric for measuring classification performance of customer churn prediction models. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 961–973. [[CrossRef](#)]
- Pozzolo, A.D.; Boracchi, G.; Caelen, O.; Alippi, C.; Bontempi, G. Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *28*, 3784–3797.
- Cao, H.; Li, X.L.; Woon, D.Y.K.; Ng, S.K. Integrated oversampling for imbalanced time series classification. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 2809–2822. [[CrossRef](#)]
- Chawla, N.; Bowyer, K.; Hall, L.; Kegelmeyer, W. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [[CrossRef](#)]
- Yu, H.; Sun, C.; Yang, X.; Zheng, S.; Zou, H. Fuzzy support vector machine with relative density information for classifying imbalanced data. *IEEE Trans. Fuzzy Syst.* **2019**, *27*, 2353–2367. [[CrossRef](#)]
- Lin, C.F.; Wang, S.D. Fuzzy support vector machines. *IEEE Trans. Neural Netw.* **2002**, *13*, 464–471.
- Sun, Y.; Kamel, M.S.; Wong, A.K.C.; Wang, Y. Cost-Sensitive Boosting for Classification of Imbalanced Data. *Pattern Recognit.* **2007**, *40*, 3358–3378. [[CrossRef](#)]
- Li, K.; Kong, X.; Lu, Z.; Wenyin, L.; Yin, J. Boosting weighted ELM for imbalanced learning. *Neurocomputing* **2014**, *128*, 15–21. [[CrossRef](#)]
- Zong, W.; Huang, G.B.; Chen, Y. Weighted extreme learning machine for imbalance learning. *Neurocomputing* **2013**, *101*, 229–242. [[CrossRef](#)]
- Krawczyk, M.W.B.; Schaefer, G. Cost-sensitive decision tree ensembles for effective imbalanced classification. *Appl. Soft Comput.* **2014**, *14*, 554–562. [[CrossRef](#)]
- Fan, W.; Stolfo, S.J.; Zhang, J.; Chan, P.K. Adacost: Misclassification Cost-Sensitive Boosting. In Proceedings of the International Conference on Machine Learning, Bled, Slovenia, 27–30 June 1999; pp. 97–105.
- Batuwita, R.; Palade, V. FSVM-CIL: Fuzzy support vector machines for class imbalance learning. *IEEE Trans. Fuzzy Syst.* **2010**, *18*, 558–571. [[CrossRef](#)]

16. Yao, L.; Wong, P.K.; Zhao, B.; Wang, Z.; Lei, L.; Wang, X.; Hu, Y. Cost-Sensitive Broad Learning System for Imbalanced Classification and Its Medical Application. *Mathematics* **2022**, *10*, 829. [[CrossRef](#)]
17. Ramos-López, D.; Maldonado, A.D. Cost-Sensitive Variable Selection for Multi-Class Imbalanced Datasets Using Bayesian Networks. *Mathematics* **2021**, *9*, 156. [[CrossRef](#)]
18. Loyola-Gonzalez, O.; Martinez-Trinidad, J.F.C.O.; Carrasco-Ochoaand, J.A.; Garcia-Borroto, M. Cost-Sensitive Pattern-Based classification for Class Imbalance problems. *IEEE Access* **2019**, *7*, 60411–60427. [[CrossRef](#)]
19. Wang, S.; Yao, X. Diversity analysis on imbalanced data sets by using ensemble models. In Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, Nashville, TN, USA, 30 March–2 April 2009; pp. 324–331.
20. Chawla, N.V.; Lazarevic, A.; Hall, L.O.; KBowyer, W.K. SMOTEBoost: Improving prediction of the minority class in boosting. In Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Database, Cavtat-Dubrovnik, Croatia, 22–26 September 2003; pp. 107–119.
21. Seiffert, C.; Khoshgoftaar, T.M.; Hulse, J.V.A.; Napolitano, A. RUSBoost: A Hybrid Approach to Alleviating Class Imbalance. *IEEE Trans. Syst. Man Cybern.-Part A Syst. Hum.* **2010**, *40*, 185–197. [[CrossRef](#)]
22. Liu, X.; Wu, J.; Zhou, Z. Exploratory Undersampling for Class-Imbalance Learning. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2009**, *39*, 539–550.
23. Seiffert, C.; Khoshgoftaar, T.M.; Hulse, J.V.; Napolitano, A. Resampling or Reweighting: A Comparison of Boosting Implementations. In Proceedings of the 2008 20th IEEE International Conference on Tools with Artificial Intelligence, Dayton, OH, USA, 3–5 November 2008; pp. 445–451. [[CrossRef](#)]
24. Zhang, X.; Nojima, Y.; Ishibuchi, H.; Hu, W.; Wang, S. Prediction by Fuzzy Clustering and KNN on Validation Data With Parallel Ensemble of Interpretable TSK Fuzzy Classifiers. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *52*, 400–414. [[CrossRef](#)]
25. Qin, B.; Chung, F.-L.; Wang, S. Biologically Plausible Fuzzy-Knowledge-Out and Its Induced Wide Learning of Interpretable TSK Fuzzy Classifiers. *IEEE Trans. Fuzzy Syst.* **2020**, *28*, 1276–1290. [[CrossRef](#)]
26. Zhou, W.; Li, H.; Bao, M. Stochastic Configuration Based Fuzzy Inference System with Interpretable Fuzzy Rules and Intelligence Search Process. *Mathematics* **2023**, *11*, 614. [[CrossRef](#)]
27. Qin, B.; Chung, F.-L.; Wang, S. KAT: A Knowledge Adversarial Training Method for Zero-Order Takagi–Sugeno–Kang Fuzzy Classifiers. *IEEE Trans. Cybern.* **2021**, *52*, 6857–6871. [[CrossRef](#)]
28. Qin, B.; Chung, F.-L.; Nojima, Y.; Ishibuchi, H.; Wang, S. Fuzzy rule dropout with dynamic compensation for wide learning algorithm of TSK fuzzy classifier. *Appl. Soft Comput.* **2022**, *127*, 109410. [[CrossRef](#)]
29. Fernández, A.; García, S.; del Jesus, M.J.; Herrera, F. A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets Syst.* **2008**, *159*, 2378–2398. [[CrossRef](#)]
30. Cordon, O.; del Jesus, M.J.; Herrera, F. A proposal on reasoning methods in fuzzy rule-based classification systems. *Int. J. Approx. Reason.* **1999**, *20*, 21–45. [[CrossRef](#)]
31. Soler, V.; Cerquides, J.; Sabria, J.; Roig, J.; Prim, M. Imbalanced datasets classification by fuzzy rule extraction and genetic methods. In Proceedings of the Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06), Hong Kong, China, 18–22 December 2006; pp. 330–336.
32. Ishibuchi, H.; Yamamoto, T. Fuzzy rule selection by multi-objective genetic local search methods and rule evaluation measures in data mining. *Fuzzy Sets Syst.* **2004**, *141*, 59–88. [[CrossRef](#)]
33. Ishibuchi, H.; Yamamoto, T. Rule weight specification in fuzzy rule-based classification systems. *IEEE Trans. Fuzzy Syst.* **2005**, *13*, 428–435. [[CrossRef](#)]
34. Information Resources Management Association USA. *Fuzzy Systems: Concepts, Methodologies, Tools, and Applications*; Springer: Heidelberg, Germany, 2017.
35. Qin, B.; Nojima, Y.; Ishibuchi, H.; Wang, S. Realizing Deep High-Order TSK Fuzzy Classifier by Ensembling Interpretable Zero-Order TSK Fuzzy Subclassifiers. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 3441–3455. [[CrossRef](#)]
36. Sonbol, A.H.; Fadali, M.S.; Jafarzadeh, S. TSK fuzzy function approximators: Design and accuracy analysis. *IEEE Trans. Syst. Man Cybern. B Cybern.* **2012**, *42*, 702–712. [[CrossRef](#)] [[PubMed](#)]
37. Min, Y.; Abbe, E. Communication-computation efficient gradient coding. International Conference on Machine Learning. *PMLR* **2018**, *80*, 5610–5619.
38. Wang, S.; Chung, K.F.-L. On least learning machine. *J. Jiangnan Univ. (Natural Sci. Ed.)* **2010**, *9*, 505–510.
39. Wang, S.; Jiang, Y.; Chung, F.-L.; Qian, P. Feedforward kernel neural networks, generalized least learning machine, and its deep learning with application to image classification. *Appl. Soft Comput.* **2015**, *37*, 125–141. [[CrossRef](#)]
40. Wang, S.; Chung, F.-L.; Wu, J.; Wang, J. Least learning machine and its experimental studies on regression capability. *Appl. Soft Comput.* **2014**, *21*, 677–684. [[CrossRef](#)]
41. Zhou, T.; Ishibuchi, H.; Wang, S. Stacked Blockwise Combination of Interpretable TSK Fuzzy Classifiers by Negative Correlation Learning. *IEEE Trans. Fuzzy Syst.* **2018**, *26*, 3327–3341. [[CrossRef](#)]
42. Alcal-Fdez, J.; Fernandez, A.; Luengo, J.; Derrac, J.; Garcia, S. KEEL Data-Mining Software Tool: Dataset Repository, Integration of Methods and Experimental Analysis Framework. *J. Mult.-Valued Log. Soft Comput.* **2011**, *17*, 255–287.
43. Lichman, M. UCI Machine Learning Repository. 2013. Available online: <http://archive.ics.uci.edu/ml> (accessed on 15 March 2023).

44. Zhang, Y.; Ishibuchi, H.; Wang, S. Deep Takagi-Sugeno-Kang fuzzy classifier with shared linguistic fuzzy rules. *IEEE Trans. Fuzzy Syst.* **2018**, *26*, 1535–1549. [[CrossRef](#)]
45. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.