applied sciences

MDPI

Article

A Multi-Classification Method Based on Optimized Binary Tree Mahalanobis-Taguchi System for Imbalanced Data

Yefang Sun, Jun Gong and Yueyi Zhang *

School of Economics and Management, China Jiliang University, Hangzhou 310018, China
* Correspondence: 02a0702026@cjlu.edu.cn

Abstract: Data imbalance is a common problem in classification tasks. The Mahalanobis-Taguchi system (MTS) has proven to be promising due to its lack of requirements for data distribution. The MTS is a binary classifier. However, multi-classification problems are more common in real life and the diversity of categories may further aggravate the difficulty of classifying imbalanced data. Imbalanced multi-classification has become an important research topic. To improve the performance of MTS in imbalanced multi-classification, we propose an algorithm called optimized binary tree MTS (Optimized BT-MTS). Mahalanobis space (MS) construction, feature selection, and threshold determination are incorporated in a unified classification framework, and joint optimization is carried out according to the principles of maximizing separability, signal-to-noise ratio, dimensionality reduction, and minimizing misclassification cost. Experimental results on several datasets show that the method can significantly reduce the overall misclassification cost and improve the performance of imbalanced data multi-classification.

Keywords: Mahalanobis-Taguchi system; imbalanced multi-classification problem; multi-class cost-sensitive; joint optimization model

check for updates

Citation: Sun, Y.; Gong, J.; Zhang, Y. A Multi-Classification Method Based on Optimized Binary Tree Mahalanobis-Taguchi System for Imbalanced Data. *Appl. Sci.* **2022**, *12*, 10179. https://doi.org/10.3390/app121910179

Academic Editor: Antonio Sarasa Cabezuelo

Received: 15 August 2022
Accepted: 8 October 2022
Published: 10 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 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

With technological advancement and industrial development, the scale of data has significantly grown and data imbalance has become a typical characteristic. Both internal and external factors can cause data imbalance. For example, in the first case, illegal and other abnormal videos only account for a small portion of internet video traffic [1], there are far fewer customers with bad credit than with good credit [2], and there are far fewer people suffering from breast cancer than healthy people [3]. The second case includes unexpected interruptions in the balanced data transmission to a database [4]. A current trend is to use machine learning methods to mine potential and hidden information from data to provide more intelligent services. Traditional classification algorithms usually assume a balanced number of samples in different categories and consistent misclassification costs for all categories. However, it is well known that patient samples are few, and their misclassification cost is much higher than that of healthy people [5]; and a high-quality production process may also produce some substandard products, whose potential cost, if they enter the market, is higher [6]. As can be seen, these assumptions are difficult to satisfy in real applications, and imbalanced data may degrade classification performance, which is specifically manifested in high overall classification accuracy, low classification accuracy of minority classes, and high overall misclassification cost.

Many imbalanced data classification methods have emerged at the data, algorithm, and ensemble learning levels. From the perspective of data distribution, data-level methods use resampling to reduce the imbalance of datasets, and can be divided into undersampling and oversampling methods. Algorithm-level methods improve performance by improving the learning process and paying more attention to minority-class samples, including cost-sensitive learning, one class learning methods, and feature methods. Ensemble learning

methods are usually combined with data- or algorithm-level methods. With the rapid development of deep networks, some methods combining ensemble learning and deep neural networks have also been proposed in recent years, which can be used to solve the imbalanced data classification problem [7,8]. Most studies have used one or two methods to improve the performance of imbalanced data classification algorithms [9–11]. Wan et al. pointed out that pre-extracted feature variables, which might be sub-optimal for subsequent classifier learning, would reduce classification performance [12]. For this reason, it is necessary to incorporate different levels of processing, such as feature extraction and algorithm improvement, into a unified pattern-recognition framework for training.

Unlike traditional machine learning classification algorithms, the MTS is a pattern-recognition technique requiring no data distribution assumptions [13]. The method combines the Mahalanobis distance (MD) for anomaly detection and the Taguchi's method for feature selection. Su et al. pointed out that the MTS is one of the most robust techniques for imbalanced data classification problems, and used it to construct a pressure ulcer prediction model, which showed high accuracy [14]. Kim et al. applied the MTS to rotating machinery fault diagnosis and showed that the higher the imbalance ratio of the data, the more significant the difference in the classification performance of the MTS compared with support vector machine, random forest, and their cost-sensitive implementations [15]. Although the imbalanced data classification performance of the MTS has been high, its feature selection and threshold determination methods still lack a complete theoretical statistical foundation [16]. In view of this, researchers have optimized the imbalanced data classification performance of the MTS in several aspects. El-Banna pointed out that despite the promise of the MTS to classify imbalanced data, it lacks an effective threshold determination method, so they proposed a nonlinear optimization model based on a ROC curve to do this [17]. Wang et al. used Fischer discriminant analysis to correct the MD and constructed a threshold optimization model; they then applied their improved MTS to the fault diagnosis of rolling bearings, whose balanced accuracy was experimentally shown to be significantly increased [18]. Hsiao et al. improved the robustness of the MTS by integrating it with a bagging algorithm [19]. Mao et al. constructed optimization models for feature selection and threshold determination [20]. All the above studies have improved the imbalanced data classification performance of the MTS, but most were proposed for binary classification problems. However, in practical applications such as disease detection, sentiment classification, and fault diagnosis, the data usually contain multiple classes, and their distribution is unbalanced. Hence, it is necessary to extend the MTS to imbalanced multi-classification problems. In addition, most studies optimize the MTS in stages, which may not be conducive to its further enhancement.

To further broaden the application field of MTS, improve its theoretical foundation, enhance its performance of imbalanced data multi-classification, and enrich the methods of imbalanced data multi-classification, we propose a more powerful imbalanced multi-classification method called optimized binary tree MTS (optimized BT-MTS). First, considering that the MS of the MTS is the only population, referring to the research of Peng et al. on BT-MTS [21], the optimized BT-MTS combines the MTS with a binary tree to realize its multi-classification application in a one-versus-all (OVA) manner. However, the approach increases the imbalance ratio of the data. Hence, MS construction, feature selection, and threshold determination are incorporated into a unified classification framework for joint optimization. The unified classification framework enables the class-selection, feature-selection, and classification process of the MS to simultaneously update the sample separability information, feature representation, and classifier information to obtain the optimal solution for the whole learning process. The unified classification framework optimizes the construction of the MS, feature selection, and threshold determination of the MTS based on the principles of maximizing the class separability, signal-to-noise ratio of normal and abnormal samples, dimensionality reduction, and minimizing the misclassification cost. Hybrid particle swarm optimization (HPSO) is used to solve what is typical mixed integer nonlinear programming (MINLP). To verify the effectiveness of the algorithm,

it was compared with BT-MTS, random forest (RF), fast correlation-based filter solution support vector machine (FCBF-SVM), synthetic minority oversampling K-nearest neighbor (SMOTE-KNN), and a backpropagation neural network (BPNN). Experimental results on several public datasets showed that the optimized BT-MTS performed better in the classification of unbalanced data as it had higher scores for macro-F1, kappa coefficient, and balanced accuracy, and lower total misclassification cost. The innovations of the optimized BT-MTS are as follows: (1) a multi-classification application of the MTS is implemented and optimized through an OVA architecture, and a multi-classification method for unbalanced data based on the MTS is explored. In addition, it is the first time MTS is used to solve the imbalanced data multi-classification problem; (2) a unified classification framework is proposed to jointly optimize the MS construction, feature selection, and classifier learning through an iterative approach, avoiding suboptimal solutions that may arise from phased optimization. (3) The number of inter-class similarity direction and the misclassification cost are embedded in the learning framework, introducing a cost-sensitive method to guide the sample MS construction, feature subset selection, and classifier threshold determination, which is conducive to improving the classification performance of the MTS for unbalanced data. It is also the first time the MTS is improved from the perspective of cost-sensitivity. (4) The signal-to-noise ratios of both normal and abnormal samples are considered, which is conducive to improving the robustness of the MTS. (5) Dimensionality reduction is incorporated in the learning framework, which makes the MTS more adaptable to big data.

The rest of this paper is organized as follows: Section 2 introduces basic theories, including MTS, cost-sensitive learning, and HPSO. The optimized BT-MTS is introduced in Section 3, and Section 4 reports its experimental results. Finally, Section 5 draws the conclusion and outlines the future outlook.

2. Methodology

2.1. Mahalanobis-Taguchi System

The MTS is a pattern recognition technology born in the field of quality engineering, which was proposed by Genichi Taguchi, a Japanese quality management expert. The MTS has the advantage of simple principle, easy operation, and good classification performance. It has been widely used in many fields, such as product quality inspection, logistic operating system anomaly detection, and chemical hazard level evaluation [22–25].

Assume a data set contains p feature variables and a total of n samples belonging to two categories. Then, we can set one type of samples to be normal samples, and the other type of samples to be abnormal samples. The normal and abnormal samples can be expressed as follows:

$$Y = [y_i(x_k)]_{m\times p}, i = 1, 2, \cdots m, k = 1, 2, \cdots p \text{ and } \widetilde{Y} = \left[\widetilde{y}_j(x_k)\right]_{(n-m)\times p}, j = m+1, m+2, \cdots n, k = 1, 2, \cdots p,$$

The conventional MTS implementation steps are as follows:

Step 1: Construct a MS

One class of the samples are used as normal samples to construct the MS, as shown in Figure 1. The mathematical expectation value of the *MDs* from the MS is close to unity [26]. Therefore, the MS, also known as the unit space, is regarded as a reference population. The MTS identifies anomalies by measuring the *MDs* between the unknown samples and the MS.

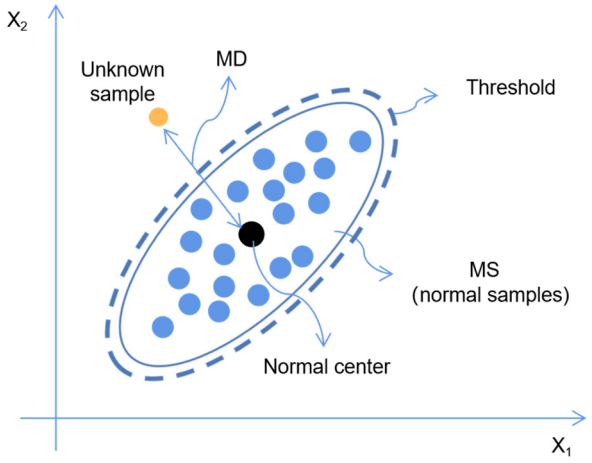


Figure 1. The measurement principle of MTS.

First, the normal samples are normalized using the following equation:

$$z_i(x_k) = \frac{y_i(x_k) - \mu(x_k)}{s(x_k)}, i = 1, 2, \cdots m, k = 1, 2, \cdots p \tag{1}$$

where $s(x_k) = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}[y_i(x_k) - \mu(x_k)]^2}, k = 1, 2, \cdots p, \mu(x_k) = \frac{1}{m}\sum_{i=1}^{m} y_i(x_k)$.

Then, the *MD* of the *i*th normal sample is obtained using the following equation:

$$MD_i = \frac{1}{\mathrm{p}} Z_i^T C^{-1} Z_i \tag{2}$$

where $Z_i^T = [z_i(x_1), z_i(x_2), \cdots z_i(x_k)]$ is the standardized data, and $C = \frac{1}{m-1}\sum_{i=1}^{m} Z_i (Z_i)^T$ is the correlation coefficient matrix.

Step 2: Verify the effectiveness of the benchmark space

First, the mean and standard deviation of the normal samples are used to normalize abnormal samples:

$$v_j(x_k) = \frac{\widetilde{y}_j(x_k) - \mu(x_k)}{s(x_k)}, j = m+1, m+2, \cdots n, k = 1, 2, \cdots p \tag{3}$$

Next, calculate the *MD* of abnormal samples according to the MS:

$$MD_j = \frac{1}{\mathrm{p}} V_j^T C^{-1} V_j \tag{4}$$

where $V_j^T = [v_j(x_1), v_j(x_2), \cdots v_j(x_k)]$ is the standardized data of abnormal samples.

Finally, the *MDs* of normal and abnormal samples are compared. If the *MD* of the abnormal samples is far greater than that of the normal samples, then the construction of this MS is reasonable.

Step 3: Select feature variables

First, on the basis of the number of feature variables, a corresponding two-level orthogonal table is selected. Level-1 means using the feature, while level-2 means not using the feature. Next, the *MD* of the abnormal samples is calculated at each experimental design. Finally, the larger-the-better signal-to-noise ratio (SNR_{LB}) of each experiment and the SNR_{LB} gain value are calculated. Assume that the two-level orthogonal table

of an arranged sample is $L_n(2^q)$. Then, the equation for calculating the SNR_{LB} of the $i(i = 1, 2, \cdots n)$th experiment of the orthogonal experimental table is as follows:

$$\eta_i = -10\lg\left(\frac{1}{n-m}\sum_{j=m+1}^{n}\frac{1}{MD_{ij}}\right) \tag{5}$$

where j is the number of abnormal samples, and MD_{ij} is the MD of $j(j = m+1, m+2, \cdots n)$th abnormal sample in the ith experiment regarding the MS. The SNR_{LB} gain value of the feature variable x_k is calculated using the following equation:

$$Gain = \overline{\eta_k^+} - \overline{\eta_k^-} \tag{6}$$

where $\overline{\eta_k^+}$ indicates the mean value of the SNR_{LB} of the feature variable that the orthogonal experiment uses, and $\overline{\eta_k^-}$ indicates that the orthogonal experiment does not use this. When the gain value of the SNR_{LB} is positive, then the feature variable is retained; otherwise, the feature variable is eliminated.

Step 4: Optimize the MS

The selected feature variables are used to reconstruct the MS. According to the optimized MS, *MDs* of normal and abnormal samples are calculated. When the MS is optimized, the differences in the *MDs* of the normal and abnormal samples are greater.

Step 5: Calculate threshold value and predict classification

In the classic MTS, the Japanese quality management expert Genich Taguchi used a quadratic loss function to obtain the threshold value. In its actual application, this method requires the subjective evaluation of experts from relevant fields, resulting in this method's limitations and operational difficulties. Thus, this study introduces the probabilistic thresholding model (PTM) to solve the threshold value problem [27]. The method uses the probability distribution of *MD*, is objective, and is easily obtainable. The calculation method is as follows:

$$\lambda = \overline{x}_{MD} + \sqrt{\frac{1}{1+\delta-\theta}} * s_{MD} \tag{7}$$

where δ is a small parameter typically set as 0.05; θ is the ratio of the number of normal samples with an *MD* less than the minimum value of the *MD* of the abnormal samples in the normal samples; and s_{MD} is the standard deviation of the *MD* of the normal samples. Then, the *MD* of the samples to be classified regarding the optimized MS is calculated. When the *MD* of the samples to be classified is smaller than or equal to the threshold value λ, then the sample is identified as a normal sample; otherwise, the sample is abnormal.

2.2. Multi-Class Cost-Sensitive Learning

Cost-sensitive learning is an important research component in the field of machine learning, addressing many real-life problems such as software defect prediction [28], medical diagnosis [29], and fraud detection [30]. The core of cost-sensitive learning lies in the construction of the cost matrix, which is shown in Figure 2 for the multi-classification problem. In this paper, the multi-classification cost matrix is determined by the ratio of the number of samples between different categories. C_{ij} is the cost of misclassifying a sample from class c_i as belonging to class c_j, the calculation equation is as follows:

$$C_{ij} = \frac{N_j}{N_i} \tag{8}$$

where N_j is the number of samples with label c_j, N_i is the number of samples with label c_i.

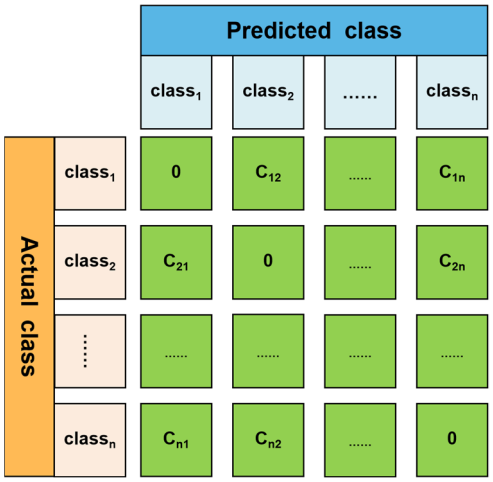


Figure 2. Schematic diagram of the cost matrix for multi-classification tasks.

BT-MTS is an OVA multi-classification method, so how to calculate the corresponding cost matrix for each MTS classifier based on the known multi-classification cost matrix is a key issue in this paper. This problem can be better solved by using binary cost-sensitive error correcting output codes (BC-SECOC) [31]; the general calculation methods are as follows.

Suppose a base classifier f_1, S_L denotes positive class, S_R denotes negative class, and both S_L,S_R contain at least one class, then the cost when a sample belonging to positive class is misclassified as negative class (cost for a false negative, *CFN*), and the contrary one (cost for a false positive, *CFP*) can be calculated as follows:

$$CFN = \sum_{c_i \in S_L} P(c_i) \sum_{c_j \in S_R} P(c_i, c_j) \omega(c_i, c_j) \tag{9}$$

$$CFP = \sum_{c_j \in S_R} P(c_j) \sum_{c_i \in S_L} P(c_i, c_j) \omega(c_j, c_i) \tag{10}$$

where $P(c_i), P(c_j)$ is the prior probability of class c_i and class c_j, can be estimated according to the number of samples contained in each class, and they satisfy the condition $\sum_{c_i \in S_L} P(c_i) = 1$ and $\sum_{c_j \in S_R} P(c_j) = 1$, respectively. $\omega()$ is the misclassification cost, which can be found from the multi-classification cost matrix. $P(c_i, c_j)$ is the probability of misclassifying a sample from class c_i as belonging to class c_j, which can be estimated from the class separability. In general, if the class separability between two classes is larger, the misclassification probability will be smaller. So, $P(c_i, c_j)$ can be computed as follows:

$$P(c_i, c_j) = \frac{1}{J(c_i, c_j)} \tag{11}$$

where $J()$ are the measures of class separability. In this paper, we used the number of inter-class similarity directions (NISD) to measure it.

2.3. Hybrid Particle Swarm Optimization

The joint optimization model for this study is a typical MINLP problem that contains both 0–1 variables, continuous variables, and nonlinear constraints. Thus, we use the HPSO to solve the optimization model.

Particle swarm optimization (PSO), proposed by Kennedy and Eberhart, is advantageous for its rapid conversion speed, few parameters, and easy operation and realization. Therefore, PSO has been widely applied in parameter optimization settings [32], process

parameter design [33], and job-shop scheduling [34]. Its particle speed updating equation and location updating equation are as follows:

$$V_{ab}^{t+1} = W \cdot V_{ab}^{t} + C_1 \cdot rand() \cdot \left(P_{pbest,ab} - X_{ab}^{t}\right) + C_2 \cdot rand() \cdot \left(P_{gbest,b} - X_{ab}^{t}\right) \quad (12)$$

$$X_{ab}^{t+1} = X_{ab}^{t} + V_{ab}^{t+1} \quad (13)$$

where W is the weight of inertia; V_{ab}^{t}, X_{ab}^{t} are the particle speed and particle location of the ath particle at the bth dimension undergoing tth iteration, respectively; C_1, C_2 are acceleration coefficients; $rand()$ represents a random number generated between one and zero; and $P_{pbest,ab}$, $P_{gbest,b}$ represent the local best and global best positions, respectively.

To resolve the problem of discreteness, PSO was optimized, and binary particle swarm optimization (BPSO) was proposed. Its particle speed updating equation is as follows:

$$S\left(V_{ab}^{t+1}\right) = \frac{1}{1 + \exp\left(-V_{ab}^{t+1}\right)} \quad (14)$$

$S\left(V_{ab}^{t+1}\right)$ is the probability of particle a having a b dimension of one. Finally, Equation (15) is used to update the particle's location in the next iteration.

$$X_{ab}^{t+1} \begin{cases} 1, if\ rand() \le S\left(V_{ab}^{t+1}\right) \\ 0, otherwise \end{cases} \quad (15)$$

The evolutionary algorithm using Equations (12)–(15), namely the HPSO, can solve the MINLP problem.

3. Proposed Method

The main task of the optimized BT-MTS is to obtain a reasonable sample classification order, according to which we construct the MS at all non-leaf nodes from top to bottom, and then perform dimensionality reduction, threshold determination, and classification prediction of samples according to the threshold. The binary tree approach is used, and its structure greatly affects classification accuracy. The classification order affects the classification accuracy of each node, and eventually affects the classification result. Misclassification accumulates along the binary tree from top to bottom and is irreversible; the closer the misclassification occurs to the root, the less ideal the classification performance is. Therefore, the MS is first constructed and classified for categories with high discrimination, which can reduce the misclassification of root nodes and improve overall classification performance. The traditional MTS used the Taguchi's method for feature selection, and a quality loss function to determine the threshold, which had many problems. This paper uses a joint optimization model to help the optimized BT-MTS to determine the sample classification order, construct the MS, filter features, and determine the threshold. The optimization objectives are as follows.

3.1. Optimization Objectives

3.1.1. Divisibility

We use the NISD to measure separability between different categories, and the larger NISD means that the two samples are less similar and easily separated [35]. Prioritizing categories that can be easily classified for classification can reduce the accumulation of errors, thereby increasing classification accuracy. The calculation method is as follows.

First, the center vector of class c_i can be calculated by using Equation (16):

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (16)$$

The weights assigned to the samples $x_1^i, x_2^i, \cdots x_{n_i}^i$ in c_i according to the Euclidean distance from the center vector, can be calculated by using Equation (17):

$$1/d\left(x_1^i, \overline{x}\right), 1/d\left(x_2^i, \overline{x}\right), \cdots 1/d\left(x_{n_i}^i, \overline{x}\right) \tag{17}$$

The intra-class similarity direction of c_i can be calculated by using Equation (18):

$$M_{\mathrm{i}} = \left(x_1^i - \overline{x}\right)/d\left(x_1^i, \overline{x}\right) + \left(x_2^i - \overline{x}\right)/d\left(x_2^i, \overline{x}\right) + \cdots \left(x_{n_i}^i - \overline{x}\right)/d\left(x_{n_i}^i, \overline{x}\right) \tag{18}$$

Then, the inter-class similarity direction between c_i and c_j can be calculated by using Equation (19):

$$P_{i,j} = q_i + q_j \tag{19}$$

where $q_i = \cos\left(M_i, d_{i,j}\right) = \left(M_i, d_{i,j}\right)/\left(\|M_i\| \bullet \|d_{i,j}\|\right)$, $q_j = \cos\left(M_j, d_{j,i}\right) = \left(M_j, d_{j,i}\right)/\left(\|M_j\| \bullet \|d_{j,i}\|\right)$, $\|d_{i,j}\|$ and $\|d_{j,i}\|$ are the vector norm from the center of class c_i to the center of class c_j.

Finally, the NISD between class c_i and c_j can be calculated by using Equation (20):

$$\widetilde{P}_{i,j} = \left|d_{i,j}\right| - \left(q_i r_i + q_j r_j\right) \tag{20}$$

where r_i and r_j represent the radius of category c_i and category c_j, respectively. Thus, the first objective of optimization model is to maximize the $\widetilde{P}_{i,j}$.

3.1.2. SNR

SNR is defined as the ratio of signal strength and noise strength. In the classic MTS, the SNR has the following functions: (1) serves as the basis for selecting key variables; (2) measures the performance of the MTS; and (3) increases the performance of the MTS. SNR is divided into three categories, namely nominal-the-best SNR, smaller-the-better SNR, and $\mathrm{SNR_{LB}}$. Conventional MTS only calculates the SNR for abnormal samples. Considering that abnormal samples with a large MD and small variance are easily identified, $\mathrm{SNR_{LB}}$ is used. The calculation method for abnormal samples is as follows:

$$\eta_j = -10\lg\left(\frac{1}{n-m}\sum_{j=m+1}^{n}\frac{1}{MD_j}\right) \tag{21}$$

Moreover, scholars maintain that the SNR of normal samples must be considered [36]. The calculation equation is as follows:

$$\eta_i = -10\lg\left(\frac{1}{m}\sum_{i=1}^{m}\frac{1}{MD_i}\right) \tag{22}$$

According to the characteristics of the MTS, the *MD* of normal samples must satisfy the following equation:

$$\frac{1}{m}\sum_{i=1}^{m} MD_i = 1 \tag{23}$$

Then $\frac{m}{\frac{1}{MD_1}+\frac{1}{MD_2}+\cdots\frac{1}{MD_m}} \leq \frac{MD_1+MD_2+\cdots+MD_m}{m} = 1$; i.e., $\frac{1}{MD_1} + \frac{1}{MD_2} + \cdots + \frac{1}{MD_m} \geq m$. Therefore, Equation (22) is smaller than or equal to zero. According to the aforementioned analysis, we learned that abnormal samples and normal samples with a large SNR lead to favorable classification performance. Thus, the second and third objectives of model optimization are to maximize Equations (21) and (22).

3.1.3. Dimensionality Reduction Rate

Feature variables conducive to identifying abnormal samples are selected. Through the reduction of the number of feature variables, classification efficiency is increased, and the

impact from noise variables is reduced, thereby further increasing classification accuracy. Taking p feature variables, the dimensionality reduction rate (DRR) is as follows:

$$DRR = \frac{p - p'}{p} \tag{24}$$

where $p\prime$ denotes the number of feature variables after dimensionality reduction. A high dimensionality reduction rate leads to few retained feature variables. Thus, the fourth objective of the optimization model is to maximize the DRR.

3.1.4. Misclassification Cost

Assume that a classifier at a non-leaf node is MTS1, and $S_L = \{c_1\}, S_R = \{c_2, c_3, c_4\}$, then the misclassification cost *CFN* of the left subtree can be simplified as:

$$CFN = \sum\nolimits_{c_{j \in S_N}} P(c_i, c_j)\omega(c_i, c_j) \tag{25}$$

The misclassification cost CFP of the right subtree can be expressed as:

$$CFP = \sum\nolimits_{c_i \in S_N} P(c_i) \sum\nolimits_{c_{j \in S_L}} P(c_i, c_j)\omega(c_i, c_j) \tag{26}$$

The classification confusion matrix of MTS1 can be expressed as:

$$\begin{array}{c} \\ P \\ N \end{array} \begin{array}{c} \begin{array}{cc} P & N \end{array} \\ \left[\begin{array}{cc} TP & FN \\ FP & TN \end{array}\right] \end{array} \tag{27}$$

Then the total classification cost of errors generated by MTS1 is:

$$\text{Cost} = CFP \times FP + CFN \times FN \tag{28}$$

Therefore, the last objective of the optimization model is to minimize Equation (28).

3.2. Joint Optimization Model

Assume the samples contain q categories, p feature variables, and a sample size of n; the MS selected during the training of the MTS is class $c_i (i = 1, 2, \cdots q)$; $(x_1, x_2, \cdots, x_p, c_i, \lambda)$ is the decision variable of the $p + 2$ dimension; and λ is the threshold of the MS. Based on the above optimization objectives, the joint optimized model can be expressed as follows:

$$\max f(x_1, x_2, \cdots, x_p, c_i, \lambda) = \widetilde{P_{i,j}} + \eta_i + \eta_j + DRR - FN_{\cos t} - FP_{\cos t} \tag{29}$$

3.3. Model Optimization Method

The joint optimization model for the benchmark space construction, the feature variables selection, and the threshold determination for this study was an MINLP problem with a loop nested. Thus, a traversal algorithm combined with HPSO can be used to solve this model. In this study, we assumed that the negative number of the target function Equation (29) is the fitness function. The smaller the function value, the more optimal the result. The concrete steps for obtaining solutions are as follows Algorithm 1.

$$
s.t.\begin{cases}
\widetilde{P}_{S_L,S_R} = \left|d_{S_L,S_R}\right| - \left(q_{S_L} r_{S_L} + q_{S_R} r_{S_R}\right) \\
\eta_t = -10\lg\left(\frac{1}{n-m}\sum\limits_{t=m+1}^{n}\frac{1}{MD_t}\right) \\
\eta_s = -10\lg\left(\frac{1}{m}\sum\limits_{s=1}^{m}\frac{1}{MD_s}\right) \\
DRR = \frac{p-p'}{p} \\
\sum\limits_{k=1}^{p} x_k = p\prime < p \\
\sum\limits_{k=1}^{p} x_k = p\prime > 0 \\
MD_s = \frac{1}{\mathrm{p}} Z_s^T C^{-1} Z_s, s = 1, 2, \cdots m \\
MD_t = \frac{1}{\mathrm{p}} Z_t^T C^{-1} Z_t, t = m+1, m+2, \cdots n \\
x_k = \begin{cases} 0, x_k \text{ is selected for classification} \\ 1, x_k \text{ is not selected for classification} \end{cases}, k = 1, 2, \cdots p \\
CFN = \sum_{c_i \in S_L} P(c_i) \sum_{c_{j \in} S_R} P(c_i, c_j) \omega(c_i, c_j) \\
CFP = \sum_{c_i \in S_R} P(c_i) \sum_{c_{j \in} S_L} P(c_i, c_j) \omega(c_i, c_j) \\
FN_{\cos t} = CFN \times FN \\
FP_{\cos t} = CFP \times FP \\
P(c_i, c_j) = \frac{1}{\widetilde{P}_{i,j}}
\end{cases} \tag{30}
$$

Algorithm 1: The concrete steps.

Input: dataset $S = \{c_1, c_2, \cdots c_i\}$, individual learning factor $C_1 = 2$, social learning factor $C_2 = 2$, inertia factor $w = 0.6$, population size $N = 30$, maximum particle velocity $v_{\max} = 1$, minimum particle velocity $v_{\min} = -1$, maximum iteration time $iterate_{\max} = 200$, and particle length $L = p + 1$

Output: fitness function values of different categories f_i, selected feature subsets fs

1. **For** $c_i = 1$ *to* q **do**:
2. Initialize the population to preliminarily obtain the personal best and global best solution of the feature variable selection and the threshold, as well as the corresponding fitness function value f_i.
3. **For** $iterate = 1$ *to* 200 **do**
4. Use Equations (12)–(15) to update the velocity and location of each particle
5. Update the personal best and global best solution of the population
6. **End for**
7. **End for**

3.4. Concrete Implementation Steps of the Optimized BT-MTS

The Optimized BT-MTS, shown in Figure 3, can select a suitable classification order for data sets. Simultaneously, the optimal feature variable subsets and thresholds of the MS are arranged. The concrete implementation steps are as follows:

Step 1: Divide data set. Data are randomly divided into training set (80%) and testing set (20%). The training set data are used for the Optimized BT-MTS model training, whereas the testing set data are used for verifying the classification performance of the model.

Step 2: Construct benchmark space, select feature subsets, and calculate the threshold values. From the top down, the solution method detailed in Section 3.3 is used. The category corresponding to the minimum fitness function value is the MS, and the corresponding feature variable subsets and thresholds are obtained.

Step 3: Remove categories that have been determined to be the MS. Then, repeat Step 2 until all of the MS of all inner nodes are determined.

Step 4: Make classification prediction. The testing set is imported into the trained model for classification prediction.

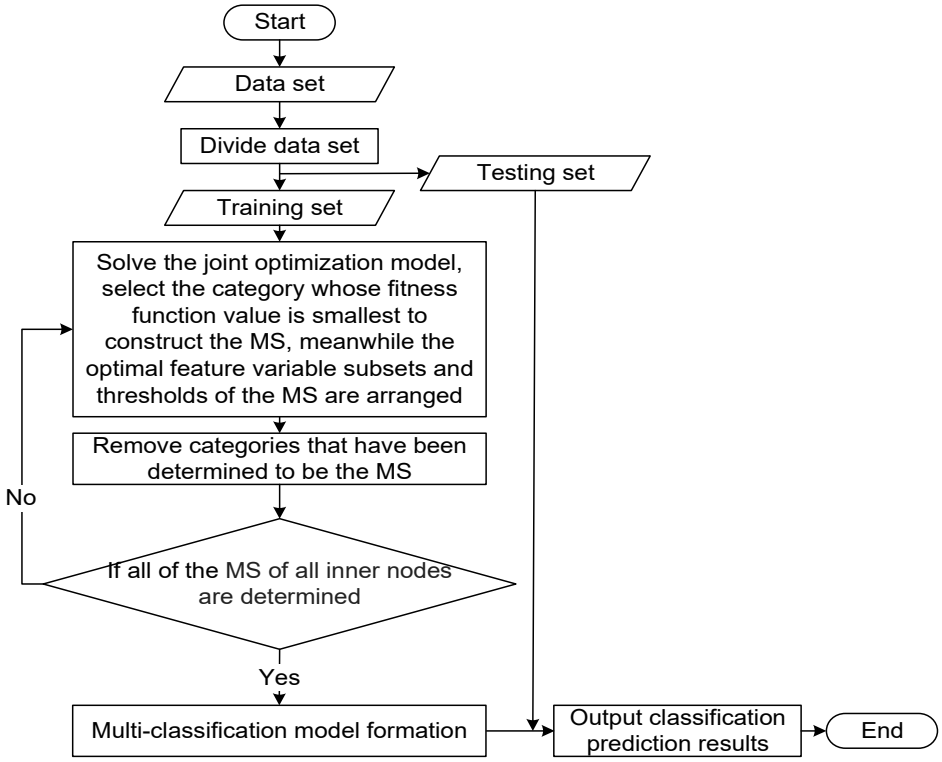


Figure 3. The schematic of the optimized BT-MTS.

4. Experimental Results and Analysis

4.1. Experimental Data and Research Method

To verify the effectiveness of the optimized BT-MTS, we selected four multi-classification data sets from the KEEL database for the experiment. The data sets are selected as follows: (1) imbalance ratio: the imbalance ratio of the dataset varies widely from 5.88 to 439.6, which better reflects the imbalance data processing ability of different methods; (2) divisibility: the data sets have different degrees of difficulty in classification, which is conducive to compare the classification performance of different methods comprehensively, and the advantages and shortcomings are more obvious; (3) sample size: considering that small sample data tend to undertrain a machine learning algorithm and render it ineffective [37], the total sample size of the data sets selected in this paper greatly varies, and the number of instances of each category within the data sets also greatly varies, which tests the performance of the algorithm more. The data sets used are listed in Table 1.

Table 1. Data set information.

Data Set	Number of Samples	Number of Feature Variables	Number of Categories	Number of Samples by Category	Imbalance Ratio
Balance	625	4	3	49/288/288	5.9
Wine quality-white	4898	11	7	20/163/1457/2198/880/175/5	439.6
Glass	214	8	6	70/76/17/13/9/29	8.4
Page-blocks	5437	10	5	4913/329/28/88/115	175.5

In this study, BT-MTS is first used as one of the comparison methods to verify the effectiveness of the optimization algorithm. The BT-MTS method uses the following processes to determine the classification sequence: in sequence, a category of the training set is used as the MS. The rest of the categories are regarded as abnormal samples. In

addition, the classification accuracy rate of the testing set can be calculated. The category with the highest classification accuracy is used as the root node of the MS. In the training and testing sets, the sample data of the root node are removed. The aforementioned steps are repeated for the remaining categories to calculate the classification accuracy rate. The category with the highest classification rate is used as the MS of the Level 2 node. This process is repeated until the MS of all non-leaf nodes has been identified. Finally, the conventional MTS implementation steps are used to train the classifiers of the inner nodes.

Furthermore, this paper selects four algorithms, RF, FCBF-SVM, SMOTE-KNN, and BPNN, as the comparison algorithms from four levels: integrated learning, feature method, data level, and neural network method. The parameter set of the algorithm is crucial for the performance of the algorithm, and each critical parameter should be optimized by using training data [38]. For the four machine learning algorithms, this study adopted the grid search method to optimize their critical parameters using MATLAB R2019a software. Regarding RF, this study optimized two critical parameters, namely the number of decision trees (Ntree) and the number of variables to select at random for each decision split (NumPredictorsToSample). The search scope of Ntree is 50–500, and the length of the step is 50. The search scope of NumPredictorsToSample is from one to the number of feature variables; the length of the step is one. For FCBF-SVM, the experiment adopts the Gaussian radial basis function kernel and optimizes the kernel parameter (KernelScale) and penalty factor (BoxConstraint). The search scope of both parameters is from 2^{-8} to 2^{8}, and the length of their step is one. Regarding KNN and BPNN, the experiment separately optimizes the k value of KNN and the number of neurons in the hidden layer (Hiddensize). Among them, the search scope for the KNN and for the number of neurons in the hidden layer is 1–20 and $\sqrt{n+m}+a$, respectively, where n is the number of input nodes, m is the number of output nodes, a is 1–10, and the search step length is one.

4.2. Select Evaluation Index

This study uses classification accuracy Equation (31), the macro-F1 Equation (32), the class balance accuracy *CBA*, Equation (33), the total cost of misclassification Equation (34), and the kappa coefficient Equation (35) as the evaluation indices for the model. The accuracy and the macro-F1 are evaluation metrics for multi-classification problems, reflecting the accuracy of the model, the kappa coefficient demonstrates the credibility of the model, and the *CBA* is proposed for imbalanced data, reflecting the classification performance of the model for imbalanced data. In addition, this paper explores a cost-sensitive classification method, for which the total misclassification cost is used as an evaluation index, which can reflect the loss caused by misclassification in different ways.

$$Accuracy = \frac{\sum_{i=1}^{q} c_{ii}}{\sum_{i,j=1}^{q} c_{ij}} \tag{31}$$

$$Macro - F_1 = \frac{2 \times macro - P \times macro - R}{macro - P + macro - R} \tag{32}$$

$$CBA = \frac{1}{q}\sum_{i=1}^{q} \frac{c_{ii}}{\max(c_{i\cdot}, c_{\cdot i})} \tag{33}$$

$$\cos t = \sum_{i=1}^{q}\sum_{j=1}^{q} c_{ij} \times \omega_{ij} \tag{34}$$

where $macro - P = \frac{1}{q}\sum_{i=1}^{q} pricision_i, macro - R = \frac{1}{q}\sum_{i=1}^{q} recall_i; pricision_i = \frac{c_{ii}}{c_{\cdot i}}, recall_i = \frac{c_{ii}}{c_{i\cdot}}$; q is the number of categories; c_{ii} is the number of samples with actual labels of i and the classification prediction results are also i; $c_{i\cdot}$ is the number of the samples with actual labels

of i; and $c_{.i}$ is the number of the samples with classification prediction results of i; c_{ij} is the number of samples with actual labels of i and the prediction results are j.

$$Kappa = \frac{p_0 - p_c}{1 - p_c} \tag{35}$$

where $p_0 = \frac{\sum_{i=1}^{q} c_{ii}}{\sum_{i,j=1}^{q} c_{ij}}, p_c = \frac{\sum_{i=1}^{q} c_{i.} \times c_{.i}}{\left(\sum_{i,j=1}^{q} c_{ij}\right)^2}$.

4.3. Analysis of Experimental Results

4.3.1. Balance Data Set

According to the steps of optimized BT-MTS, fitness function values can be obtained, as shown in Table 2, from which it can be seen that the fitness function value of class 2 is the smallest, which implies that, if the MTS constructs the MS with this class as the positive class and treats the remaining two classes as the negative class, it will have a smaller misclassification cost and better class separability, while the MS has better robustness. After removing this class, the fitness values of the remaining categories are calculated, and it can be found that class 3 has a smaller fitness function value than class 1. Therefore, class 3 is used as the benchmark at the second node to construct the MS. Finally, the classification order of the balanced dataset is 2, 3, and 1. The feature subsets of base classifiers MTS1 and MTS2 obtained according to the proposed method are shown in Table 3, with threshold values for the base classifier of 1.6316 and 0.9470, respectively.

Table 2. Values of fitness function at different stages of balance data set.

	c_1	c_2	c_3
Stage 1	418.0586	−11.4987	−3.8107
Stage 2	37.4477	/	16.1342

Table 3. Feature variable selection results of the Optimized BT-MTS of the balance data set.

	x_1	x_2	x_3	x_4
MTS1	1	1	1	1
MTS2	0	1	1	0

Note. "1"means using the variable, "0" means not using the variable.

The classification order of BT-MTS is 2, 3, and 1. The threshold values for MTS1 and MTS2 are 1.5816 and 1.5813, respectively, and their feature selection results are shown in Table 4. Figure 4 compares the DRR of the two MTS methods.

Table 4. Feature variable selection results of BT-MTS of of the balance data set.

	x_1	x_2	x_3	x_4
MTS1	1	1	1	1
MTS2	1	0	1	1

Note. "1"means using the variable, "0" means not using the variable.

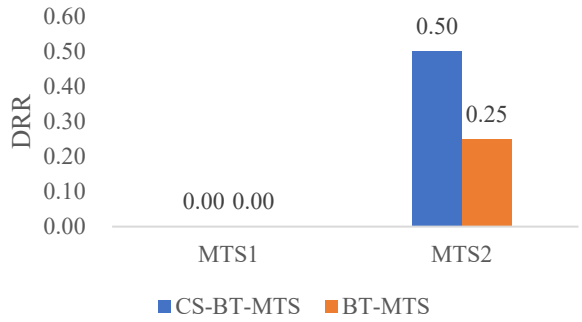


Figure 4. Comparison of DRR of the balance data set.

The balance data set comparison results of the six methods are presented in Table 5.

Table 5. Comparison of the classification results of the balance data set.

Method	Parameter Set	Evaluation Index				
		Accuracy	Macro-F1	Kappa Coefficient	CBA	Cost
Optimized BT-MTS	Joint optimization model	0.8880	0.7939	0.8102	0.7473	31.7489
BT-MTS	Use SNR_{LB} gain value to select feature variables, PTM	0.7680	0.6590	0.6151	0.5866	114.3063
RF	Ntree = 50, NumPredictorsToSample = 2	0.8560	0.6076	0.7411	0.5873	74.0410
FCBF-SVM	BoxConstraint = 3.0625, KernelScale = 0.0625	0.7600	0.5321	0.5622	0.5074	88.5306
SMOTE-KNN	K = 4	0.7500	0.7549	0.6223	0.6970	83.1721
BPNN	Hiddensize = 4	0.8960	0.8186	0.8238	0.7695	26.0712

The feature variable selection result of FCBF-SVM is x_1, x_3, x_4. As shown in Table 5, from the comprehensive view of accuracy, macro-F1, kappa coefficient, CBA, and cost, the classification performance of optimized BT-MTS is significantly better than that of BT-MTS. The dimensionality reduction of the proposed method is slightly higher than that of BT-MTS, and the results indicate that it effectively improves the multi-classification performance of BT-MTS.

Furthermore, comparison of the proposed method with four machine learning algorithms shows that the algorithm in this paper and BPNN performed better on all five evaluation metrics than the ensemble learning algorithms RF, FCBF-SVM, and SMOTE-KNN.

4.3.2. Wine Quality-White Data Set

According to the steps of optimized BT-MTS, fitness function values can be obtained, as shown in Table 6, from which it can be seen that the fitness function value of class 5 is the smallest. Therefore, this class should be used as the left subtree of the root node to construct the MS. After removing this class, the fitness values of the remaining categories are calculated. Finally, the classification order of the data set is category 5, 6, 3, 4, 8, 9, and 7. The threshold values for MTS1–MTS6 are 0.8429, 1.7956,2.7071, 1.3308, 2.6967, and 2.0202. The feature selection results are shown in Table 7.

Table 6. Values of fitness function at different stages of the Wine quality-white data set.

	c_3	c_4	c_5	c_6	c_7	c_8	c_9
Stage 1	−26,928	−260	−2,110,498	−6114	−128,959	−141,914	−2,015,752
Stage 2	−10,899	−10,439	/	−42,569	−11,102	−3185	−1208
Stage 3	−487,094	−60,492	/	/	−54,679	−233,864	−1835
Stage 4	/	−676,814	/	/	−201	−16,650	−21,435
Stage 5	/	/	/	/	−764	−106,886	−27,410
Stage 6	/	/	/	/	−115,434		−158,500

Table 7. Feature variable selection results of Optimized BT-MTS of the Wine quality-white data set.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
MTS1	0	0	0	0	1	0	0	1	1	0	0
MTS2	0	1	0	0	1	0	0	0	0	0	0
MTS3	1	1	0	0	0	0	0	1	1	0	1
MTS4	0	1	1	0	0	0	0	0	0	0	1
MTS5	0	1	1	1	1	1	0	0	0	1	1
MTS6	0	0	1	0	1	0	0	0	0	0	1

Note. "1"means using the variable, "0" means not using the variable.

According to the BT-MTS method, the classification sequence is category 5, 9, 8, 3, 6, 7, and 4. The MTS classifiers were thus trained in sequence, resulting in the feature variable selection results in Table 8. The threshold values for MTS1–MTS6 are 1.6811, 0.6667, 1.5734, 1.2118, 3.3241, and 1.7486, respectively. The DRR of the optimized BT-MTS and BT-MTS are shown in Figure 5, and it can be seen that the dimensionality reduction performance of this paper's method is better than that of BT-MTS.

Table 8. Feature variable selection results of BT-MTS of the Wine quality-white data set.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
MTS1	0	1	0	1	0	1	1	1	1	1	1
MTS2	1	1	1	1	1	1	1	1	1	1	1
MTS3	1	1	1	1	1	1	1	1	1	1	1
MTS4	1	1	1	1	0	0	1	1	1	1	1
MTS5	0	1	0	1	0	1	1	1	1	1	1
MTS6	0	1	1	1	0	1	1	1	1	0	1

Note. "1"means using the variable, "0" means not using the variable.

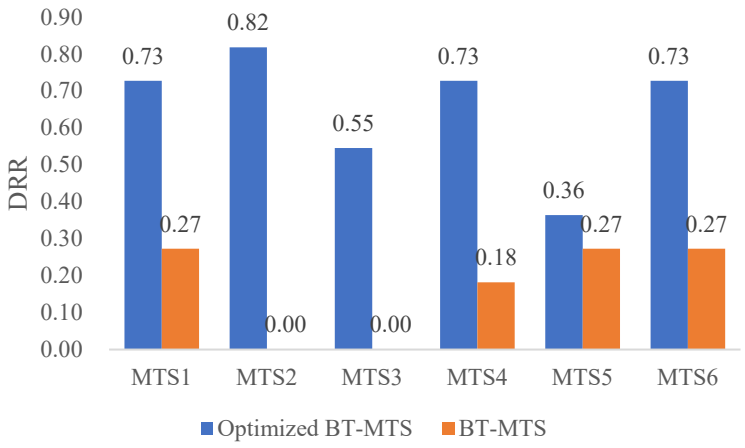


Figure 5. Comparison of DRR of the Wine quality-white data set.

The balance data set comparison results of the six methods are presented in Table 9.

Table 9. Comparison of the classification results of the Wine quality-white data set.

Method	Parameter Set	Evaluation Index				
		Accuracy	Macro-F1	Kappa Coefficient	CBA	Cost
Optimized BT-MTS	synergistic optimization model	0.6285	0.3101	0.4306	0.3097	1,278.4892
BT-MTS	Use SNRLB gain value to select feature variables, PTM	0.3102	0.1115	0.0607	0.1062	6,338.7487
RF	Ntree = 150, NumPredictorsToSample = 3	0.7140	0.4155	0.5571	0.3578	2,059.7415
FCBF-SVM	BoxConstraint = 0.0625, KernelScale = 4.0625	0.4333	0.0879	/	0.0635	3,013.3218
SMOTE-KNN	K = 5	0.5955	0.3717	0.3888	0.3398	1,759.4939
BPNN	Hiddensize = 5	0.5414	0.2560	0.2611	0.2102	2,446.9699

As can be seen from the table, the method of optimized BT-MTS in this paper outperforms the algorithms of BT-MTS, FCBF-SVM, SMOTE-KNN, and BPNN, indicating that the algorithm in this paper can better adapt to the data environment with a high imbalance ratio. Comparing the method of this paper with RF, we can find that the accuracy, macro-F1, kappa coefficient, and CBA of this paper's algorithm are slightly lower, but the total misclassification cost of the optimized BT-MTS is significantly lower, which also indicates that this paper considers the cost-sensitive joint optimization model to improve the focus on minority class samples in the classification process.

4.3.3. Glass Data Set

According to the steps of optimized BT-MTS, fitness function values can be obtained, as shown in Table 10, from which it can be seen that the classification sequence of optimized BT-MTS was category 7, 2, 3, 5, 1, and 6. The thresholds for the MTS1-MTS5 are 5.6075, 5.1663, 29.8456, 14.2530, and 10.1904. The feature selection results are shown in Table 11. It should be noted that the number of samples in category 6 is small, and after the random division of training and testing samples, the number of samples involved in training is even smaller and insufficient for calculation, so this paper directly places it in the last position for classification.

Table 10. Values of fitness function at different stages of the glass data set.

	c_1	c_2	c_3	c_5	c_6	c_7
Stage1	−2852.7039	−270.3196	−12,654.4039	−1,103.9032	NaN	−411,598.6086
Stage2	−160.2149	−2771.1798	−1482.8692	−261.2249	NaN	/
Stage3	−305.5313	/	−1104.4824	−250.4687	NaN	/
Stage4	−20.9576	/	/	−211.3834	NaN	/
Stage5	−192.1399	/	/	/	NaN	/

Table 11. Feature variable selection results of optimized BT-MTS of the glass data set.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
MTS1	1	0	0	0	0	0	0	0
MTS2	1	0	0	0	0	0	0	0
MTS3	1	0	0	0	0	0	0	1
MTS4	0	1	1	0	1	0	0	0
MTS5	0	1	1	0	0	0	0	1

Note. "1"means using the variable, "0" means not using the variable.

The classification order of BT-MTS is category 5, 2, 7, 1, 3, and 6, and the thresholds of MTS1-MTS5 are 2.5646, 2.2468, 2.7533, 1.9409, and 2.7274, respectively. The corresponding feature selection results are shown in Table 12, and the DRR of the two MTS methods can be obtained, as shown in Figure 6.

Table 12. Feature variable selection results of BT-MTS of the glass data set.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
MTS1	1	0	1	1	1	1	1	0
MTS2	1	1	1	1	1	0	1	1
MTS3	0	1	1	1	0	1	1	0
MTS4	1	1	0	1	1	1	0	1
MTS5	0	1	1	1	0	1	1	0

Note. "1"means using the variable, "0" means not using the variable.

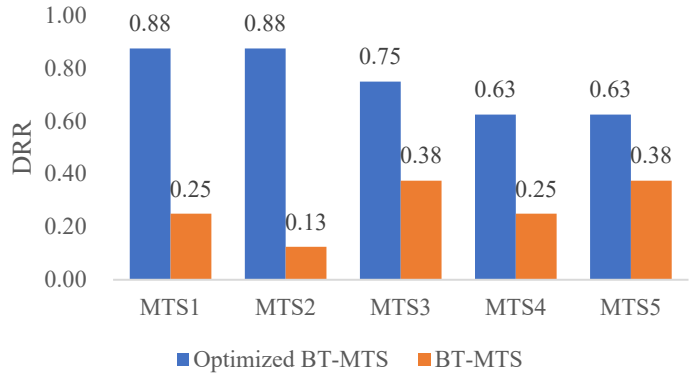


Figure 6. Comparison of DRR of the glass data set.

The balance data set comparison results of the six methods are presented in Table 13. The feature selection results of FCBF-SVM is x_4, x_8. As can be seen from the table, the accuracy, macro-F1, kappa coefficient, CBA, and cost of the method in this paper are better than those of BT-MTS, FCBF-SVM, and BPNN. Compared with SMOTE-KNN and RF, the accuracy, macro-F1, and CBA of this paper's method are slightly lower, but the advantage of this paper's method is obvious in terms of the total cost of misclassification.

Table 13. Comparison of the classification results of the glass data set of the glass data set.

Method	Parameter Set	Evaluation Index				
		Accuracy	Macro-F1	Kappa Coefficient	CBA	Cost
Optimized BT-MTS	joint optimization model	0.6976	0.4023	0.5841	0.4479	22.6081
BT-MTS	Use SNRLB gain value to select feature variables, PTM	0.4523	0.3687	0.3065	0.3157	108.3620
RF	Ntree = 100, NumPredictorsToSample = 3	0.7857	0.6536	0.7025	0.5972	39.3054
FCBF-SVM	BoxConstraint = 1.0625, KernelScale = 2.0625	0.5952	0.3306	0.4251	0.2822	61.1883
SMOTE-KNN	K = 6	0.6428	0.5255	0.5007	0.4979	50.3315
BPNN	Hiddensize = 5	0.5813	0.3396	0.4068	0.1629	63.1848

4.3.4. Page-Blocks Data Set

The values of the fitness functions for different categories on the page-blocks dataset are shown in Table 14, so the final classification order is 3, 5, 4, 2, and 1. The threshold values of MTS1-MTS4 are 6.1112, 5.3295, 1.6599, and 7.5448, respectively, and the feature selection results of the joint optimization model in this paper are shown in Table 15.

Table 14. Values of fitness function at different stages of the Page-blocks data set.

	c_1	c_2	c_3	c_4	c_5
Stage1	−45.4349	−365.7163	−123,953.3926	−4116.0841	−40,928.4861
Stage2	−26.5124	841.9145	/	2696.5264	−1993.7109
Stage3	−1211.3206	462.6110	/	−4297.0959	/
Stage4	−193.7301	−360.7507	/	/	/

Table 15. Feature variable selection results of Optimized BT-MTS of the Page-blocks data set.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
MTS1	0	0	1	0	1	1	0	0	1	0
MTS2	0	0	0	0	1	1	1	0	0	1
MTS3	0	0	0	0	1	0	1	0	0	0
MTS4	0	0	0	0	1	0	0	1	0	0

Note. "1"means using the variable, "0" means not using the variable.

The classification order of the BT-MTS method is category 1, 4, 2, 3, and 5, and the threshold values are 8.7889, 4.0251, 3.5549, and 3.4798, respectively. The feature subsets obtained according to the Taguchi's method are shown in Table 16, and the DRR of the two MTS methods are shown in Figure 7.

Table 16. Feature variable selection results of BT-MTS of the Page-blocks data set.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
MTS1	1	0	0	1	1	1	1	0	0	0
MTS2	1	1	0	1	1	1	0	0	0	1
MTS3	1	0	0	0	1	1	0	0	0	0
MTS4	1	1	1	1	1	1	1	1	1	0

Note. "1"means using the variable, "0" means not using the variable.

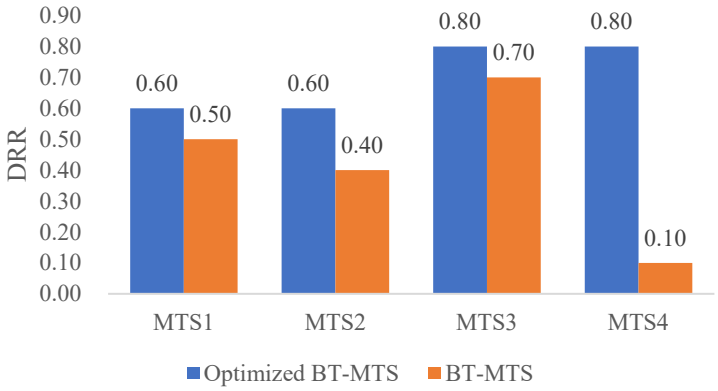


Figure 7. The DRR of MTS classifiers of the Page-blocks data set.

The Page-blocks data set comparison results of the six methods are presented in Table 17, and the feature selection result of FCBF-SVM is x_4, x_8.

Table 17. Comparison of the classification results of the Page-blocks data set.

Method	Parameter Set	Evaluation Index				
		Accuracy	Macro-F1	Kappa Coefficient	CBA	Cost
CS-BT-MTS	synergistic optimization model	0.9662	0.8156	0.8145	0.7368	813.4696
BT-MTS	Use SNR_{LB} gain value to select feature variables, PTM	0.9251	0.4635	0.4939	0.3837	4.801.1563
RF	Ntree = 100, NumPredictorsToSample = 3	0.9744	0.8375	0.8681	0.8070	560.3284
FCBF-SVM	BoxConstraint = 2.0625, KernelScale = 1.0625	0.9269	0.5044	0.5423	0.4537	2,740.8484
SMOTE-KNN	K = 6	0.9759	0.9757	0.9699	0.9679	1,189.8887
BPNN	Hiddensize = 3	0.9534	0.7702	0.7362	0.6915	1325.6219

As it can be observed from Table 17, the accuracy, macro-F1, kappa coefficient, CBA, and total misclassification cost of optimized BT-MTS are better than those of BT-MTS. In addition, the DRR of the method proposed in this paper is also higher, which also indicates that the optimized BT-MTS not only improves the classification accuracy but also enhances the classification efficiency, and the ability of MTS to solve the imbalanced multi-classification problem is enhanced. Compared with the FCBF-SVM method, the optimized BT-MTS did not have a clear advantage regarding the average DRR. However, in terms of imbalanced multi-classification performance, the optimized BT-MTS was substantially superior to the FCBF-SVM. This may be due to the fact that the FCBF-SVM method loses some important information in the process of feature selection. Compared with the methods of RF, SMOTE-KNN, and BPNN, the classification performance of the optimized BT-MTS and RF is relatively better from the comprehensive view of the five evaluation indexes. Although the accuracy, macro-F1, kappa coefficient, and CBA of the SMOTE-KNN are

higher, its misclassification leads to a larger loss, which will lead to serious consequences in real life.

5. Conclusions

The imbalanced multi-classification problem has broad research significance and application value, and is important in the field of machine learning. The MTS has become one of the most robust methods to classify unbalanced data because it has no requirement on the distribution of data. However, the MTS was initially proposed for binary classification, and cannot be directly used to solve multi-classification problems, and the MS of the MTS consists of normal samples, which is the only population. Therefore, we first combine the MTS with the binary tree approach to decompose the multi-classification problem into multiple binary classification problems through the OVA architecture, and thus realize its application to multi-classification. On this basis, we propose an optimized BT-MTS for multi-classification. It incorporates the construction of the MS, feature selection, and threshold determination into a unified framework to simultaneously determine the classification order of samples, feature variables involved in classification, and classification threshold by updating the information on class separability, feature subsets, and classifiers, including their threshold, robustness, and performance, through multiple iterations. The experimental results on four imbalanced multi-class datasets show that this method has better classification performance for imbalanced multi-class data than BT-MTS. In addition, compared with RF, SMOTE-KNN, FCBF-SVM, and BPNN, the macro-F1, kappa coefficient, and balanced accuracy of the proposed method are higher, at a significantly lower total misclassification cost. This has important theoretical and practical significance, as the generalization ability of the MTS is improved, and the possibility of misclassification of high-cost samples is greatly reduced.

Author Contributions: The lead author of this paper is Y.S. Additional support was provided by Y.Z. and J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Social Science Foundation of China (Project Grant No. 18BJY033).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The calculation software used in this paper is MATLAB 2019a. The Kaggle database is available online at https://sci2s.ugr.es/keel/imbalanced.php#sub40, accessed on 8 June 2022.

Acknowledgments: The authors would like to express sincere gratitude to School of Economics and Management form China Jiliang University.

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

References

1. Teng, A.; Peng, L.; Xie, Y.; Zhang, H.; Chen, Z. Gradient descent evolved imbalanced data gravitation classification with an application on Internet video traffic identification. *Inf. Sci.* **2020**, *539*, 447–460. [CrossRef]
2. Zhang, T.; Chi, G. A heterogeneous ensemble credit scoring model based on adaptive classifier selection: An application on imbalanced data. *Int. J. Financ. Econ.* **2021**, *26*, 4372–4385. [CrossRef]
3. Zhang, J.; Chen, L.; Tian, J.-X.; Abid, F.; Yang, W.; Tang, X.-F. Breast Cancer Diagnosis Using Cluster-based Undersampling and Boosted C5.0 Algorithm. *Int. J. Control Autom. Syst.* **2021**, *19*, 1998–2008. [CrossRef]
4. Liu, Y.; Wang, Y.; Ren, X.; Zhou, H.; Diao, X. A Classification Method Based on Feature Selection for Imbalanced Data. *IEEE Access* **2019**, *7*, 81794–81807. [CrossRef]
5. Xu, Z.; Shen, D.; Nie, T.; Kou, Y. A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data. *J. Biomed. Inform.* **2020**, *107*, 103465. [CrossRef]
6. Wu, Z.; Zhou, C.; Xu, F.; Lou, W. A CS-AdaBoost-BP model for product quality inspection. *Ann. Oper. Res.* **2022**, *308*, 685–701. [CrossRef]

7. Kwon, H.; Kim, Y.; Park, K.-W.; Yoon, H.; Choi, D. Advanced Ensemble Adversarial Example on Unknown Deep Neural Network Classifiers. *IEICE Trans. Inf. Syst.* **2018**, *E101-D*, 2485–2500. [CrossRef]
8. Kwon, H. Detecting Backdoor Attacks via Class Difference in Deep Neural Networks. *IEEE Access* **2020**, *8*, 191049–191056. [CrossRef]
9. Du, H.; Zhang, Y.; Gang, K.; Zhang, L.; Chen, Y.-C. Online ensemble learning algorithm for imbalanced data stream. *Appl. Soft Comput.* **2021**, *107*, 107378. [CrossRef]
10. Hoyos-Osorio, J.; Alvarez-Meza, A.; Daza-Santacoloma, G.; Orozco-Gutierrez, A.; Castellanos-Dominguez, G. Relevant information undersampling to support imbalanced data classification. *Neurocomputing* **2021**, *436*, 136–146. [CrossRef]
11. Guo, S.; Wei, M.; Wang, S.; Chen, R.; Guo, C.; Li, H.; Li, T. Identify High-Impact Bug Reports by Combining the Data Reduction and Imbalanced Learning Strategies. *Appl. Sci.* **2019**, *9*, 3663. [CrossRef]
12. Wan, J.; Chen, Y.; Bai, B. Joint feature extraction and classification in a unified framework for cost-sensitive face recognition. *Pattern Recognit.* **2021**, *115*, 107927. [CrossRef]
13. Chang, Z.P.; Li, Y.W.; Fatima, N. A theoretical survey on Mahalanobis-Taguchi system. *Measurement* **2019**, *136*, 501–510. [CrossRef]
14. Su, C.-T.; Wang, P.-C.; Chen, Y.-C.; Chen, L.-F. Data Mining Techniques for Assisting the Diagnosis of Pressure Ulcer Development in Surgical Patients. *J. Med. Syst.* **2012**, *36*, 2387–2399. [CrossRef] [PubMed]
15. Kim, S.-G.; Park, D.; Jung, J.-Y. Evaluation of One-Class Classifiers for Fault Detection: Mahalanobis Classifiers and the Mahalanobis–Taguchi System. *Processes* **2021**, *9*, 1450. [CrossRef]
16. Woodall, W.H.; Koudelik, R.; Tsui, K.-L.; Kim, S.B.; Stoumbos, Z.G.; Carvounis, C.P. A Review and Analysis of the Mahalanobis—Taguchi System. *Technometrics* **2003**, *45*, 1–15. [CrossRef]
17. El-Banna, M. Modified Mahalanobis Taguchi System for Imbalance Data Classification. *Comput. Intell. Neurosci.* **2017**, *2017*, 5874896. [CrossRef]
18. Wang, N.; Zhang, Z.; Zhao, J.; Hu, D. Recognition method of equipment state with the FLDA based Mahalanobis–Taguchi system. *Ann. Oper. Res.* **2022**, *311*, 417–435. [CrossRef]
19. Hsiao, Y.-H.; Su, C.-T.; Fu, P.-C. Integrating MTS with bagging strategy for class imbalance problems. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 1217–1230. [CrossRef]
20. Mao, T.; Zhou, L.; Zhang, Y.; Sun, Y. Classification algorithm for class imbalanced data based on optimized Mahalanobis-Taguchi system. *Appl. Intell.* **2022**, *52*, 10674–10691. [CrossRef]
21. Peng, Z.; Cheng, L.; Yao, Q. Multi-feature Extraction for Bearing Fault Diagnosis Using Binary-tree Mahalanobis-Taguchi System. In Proceedings of the 31st Chinese Control and Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 3303–3308.
22. Peng, C.-F.; Ho, L.-H.; Tsai, S.-B.; Hsiao, Y.-C.; Zhai, Y.; Chen, Q.; Chang, L.-C.; Shang, Z. Applying the Mahalanobis–Taguchi System to Improve Tablet PC Production Processes. *Sustainability* **2017**, *9*, 1557. [CrossRef]
23. Cheng, L.; Yaghoubi, V.; Van Paepegem, W.; Kersemans, M. Mahalanobis classification system (MCS) integrated with binary particle swarm optimization for robust quality classification of complex metallic turbine blades. *Mech. Syst. Signal Process.* **2021**, *146*, 107060. [CrossRef]
24. Asakura, T.; Yashima, W.; Suzuki, K.; Shimotou, M. Anomaly Detection in a Logistic Operating System Using the Mahalanobis–Taguchi Method. *Appl. Sci.* **2020**, *10*, 4376. [CrossRef]
25. Huh, D.-A.; Lim, H.L.; Sohn, J.-R.; Byeon, S.-H.; Jung, S.; Lee, W.-K.; Moon, K.W. Development of a Screening Method for Health Hazard Ranking and Scoring of Chemicals Using the Mahalanobis–Taguchi System. *Int. J. Environ. Res. Public Health* **2018**, *15*, 2208. [CrossRef]
26. Ramlie, F.; Muhamad, W.; Harudin, N.; Abu, M.; Yahaya, H.; Jamaludin, K.; Talib, H.A. Classification Performance of Thresholding Methods in the Mahalanobis–Taguchi System. *Appl. Sci.* **2021**, *11*, 3906. [CrossRef]
27. Su, C.-T.; Hsiao, Y.-H. An Evaluation of the Robustness of MTS for Imbalanced Data. *IEEE Trans. Knowl. Data Eng.* **2007**, *19*, 1321–1332. [CrossRef]
28. Niu, L.; Wan, J.; Wang, H.; Zhou, K. Cost-sensitive Dictionary Learning for Software Defect Prediction. *Neural Process. Lett.* **2020**, *52*, 2415–2449. [CrossRef]
29. Gan, D.; Shen, J.; An, B.; Xu, M.; Liu, N. Integrating TANBN with cost sensitive classification algorithm for imbalanced data in medical diagnosis. *Comput. Ind. Eng.* **2020**, *140*, 106266. [CrossRef]
30. Höppner, S.; Baesens, B.; Verbeke, W.; Verdonck, T. Instance-dependent cost-sensitive learning for detecting transfer fraud. *Eur. J. Oper. Res.* **2022**, *297*, 291–300. [CrossRef]
31. Xue, A.; Wang, X. Cost-sensitive design of error correcting output codes. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2018**, *232*, 1871–1881. [CrossRef]
32. Su, Y.; Han, L.; Wang, J.; Wang, H. Quantum-behaved RS-PSO-LSSVM method for quality prediction in parts production processes. *Concurr. Comput. Pract. Exp.* **2022**, *34*, 5522. [CrossRef]
33. Sibalija, T.V. Particle swarm optimisation in designing parameters of manufacturing processes: A review (2008–2018). *Appl. Soft Comput.* **2019**, *84*, 105743. [CrossRef]
34. Nouiri, M.; Bekrar, A.; Jemai, A.; Niar, S.; Ammari, A.C. An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem. *J. Intell. Manuf.* **2015**, *29*, 603–615. [CrossRef]
35. Wang, J.; He, F. Binary Tree Support Vector Machine Based on Number of Inter-Class Similarity Direction. *Stat. Decis.* **2018**, *4*, 15–19. [CrossRef]

36. Niu, J.L.; Cheng, L.S. Classification using improved Mahanobis-Taguchi system based on omni-optimizer. *Syst. Eng. Theory Pract.* **2012**, *32*, 1324–1336.
37. Ooi, B.P.; Rahim, N.A.; Zakaria, A.; Masnan, M.J.; Shukor, S.A.A. Random subspace oracle (RSO) ensemble to solve small sample-sized classification problems. *J. Intell. Fuzzy Syst.* **2019**, *36*, 3225–3234. [CrossRef]
38. Kwon, H.; Kim, Y. BlindNet backdoor: Attack on deep neural network using blind watermark. *Multimed. Tools Appl.* **2022**, *81*, 6217–6234. [CrossRef]