IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 9, 2021, accepted April 23, 2021, date of publication May 3, 2021, date of current version June 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077125

A Chi-MIC Based Adaptive Multi-Branch Decision Tree

JIAHAO YE[1], JINGJING YANG[1], JIANG YU[1], SIQIAO TAN[2], FENG LUO[3], (Senior Member, IEEE), ZHEMING YUAN[1], AND YUAN CHEN[1,4]

[1]Hunan Engineering and Technology Research Center for Agricultural Big Data Analysis and Decision-Making, Hunan Agricultural University, Changsha 410128, China
[2]Department of Information Intelligence, Hunan Agricultural University, Changsha 410128, China
[3]Department of Computing, Clemson University, Clemson, SC 29634, USA
[4]Hunan Provincial Key Laboratory of Crop Germplasm Innovation and Utilization, Hunan Agricultural University, Changsha 410128, China

Corresponding authors: Zheming Yuan (zhmyuan@sina.com) and Yuan Chen (chenyuan0510@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61701177, in part by the Hunan Provincial Natural Science Foundation under Grant 2018JJ3225, in part by the Science Foundation Open Project of Hunan Provincial Key Laboratory of Crop Germplasm Innovation and Utilization under Grant 18KFXM08, and in part by the Hunan Agricultural University Natural Science Foundation under Grant 18QN08.

ABSTRACT Since the decision trees (DTs) have an advantage over "black-box" models, such as neural nets or support vector machines, in terms of comprehensibility, such that it might merit improvement for further optimization. The node splitting measures and pruning methods are primary among the techniques that can improve the generalization abilities of DTs. Here, we introduced the unequal interval optimization for node splitting, as well as the local chi-square test for tree pruning. This new method was named an adaptive multi-branch decision tree (CMDT). 11 benchmark data sets with different scales were chosen from UCI Machine Learning Repository and coupled with 12 classifiers to evaluate the CMDT algorithm. The results showed that CMDT can be more reliable than the twelve comparative approaches, especially for imbalanced datasets. We also discussed the performance metrics and the weighted decision-making table in unbalanced data sets. The CMDT algorithm can be found here: https://github.com/chenyuan0510/CMDT.

INDEX TERMS Decision tree, node splitting, Chi-MIC, CMDT, pruning methods.

I. INTRODUCTION

With the advancement of science and technology, machine learning has been widely employed for classification and recognition tasks in many domains [1]. As it is one type of the many important techniques in data mining, decision trees (DTs) have been very popular classification tools for decades due to their higher performance fewer parameters, and better comprehensibility [2]. Furthermore, DTs can naturally result in attribute selection and work with both categorical and numerical data directly. The Iterative Dichotomiser 3 algorithm (ID3) is the well-known and most widely used DT algorithm. [3]. As seen, many efforts are also underway to improve the performance of DT, such as Improvement on ID3 Algorithm, Re optimization of ID3, C4.5, and C5.0 algorithm [4], [5]. G. Ke *et al.* proposed an improved Gradient Boosting Decision Tree (GBDT) called LightGBM that could speed up the training process of conventional GBDT by up to over 20 times while achieving almost the same accuracy [6]. P. Tzirakis *et al.* proposed the T3C algorithm to offer better splits continuous attributes and results in high accuracy whilst keeping the size of the tree [7]. However, they are not always robust [8]. The larger decision trees tend to produce poorer generalization performance. Node splitting measures and pruning methods are primary among the techniques that can improve the generalization abilities of DTs.

Overall, we proposed a Chi-MIC-based adaptive multi-branch decision tree (CMDT), which can also handle mixed-type attributes. For discrete attributes, the constructing CMDT is composed of two major phases. Firstly, cutting the number of splitting points for each attribute by backtracking method based on local chi-square test. Secondly, growing the tree and simplify each node by backtracking method based on local chi-square test, the growing phase continues until no attributes can be introduced on the backtracking criterion. The DT induction process for numerical attributes is consistent with the former, except that the numerical attribute should first be discretized by the Chi-MIC method.

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wang.

To evaluate the proposed CMDT method, 12 reference classifiers were compared on 11 benchmark datasets with different scales from UCI Machine Learning Repository [9].

The results showed that CMDT can be more reliable than the comparative approaches, especially for imbalanced data sets. The remainder of this paper is organized as follows: Section II discusses the two major phases of the DT and the background of MIC. The algorithm of CMDT and data sets are included in Section III. Section IV shows the result of CMDT compared with the other 12 classifiers in balanced and imbalanced data sets. Some issues related to performance metrics and splitting criteria in unbalanced classification are discussed in Section V. And the conclusion reveals the limitation of the algorithm in Section VI.

II. BACKGROUND

There are two major phases of the DT induction process: the growth phase and the pruning phase. In the growth process, the choice of splitting measurement is particularly important. The splitting measure provides a means for evaluating the discrimination power and significance of each attribute in the classification process, as well as the generation of child nodes when growing the tree. The central choice in tree algorithms is finding the best split point [2]. Numerous types of algorithms have been employed to calculate suitable splitting points, especially for numerical data. Most of them tend to construct binary decision trees by some splitting criteria [10], [11], such as information gain, gain ratio, Gini value [12], Kolmogorov-Smirnov distance [13] and histogram-based method [14], etc. The minimization problem for decision trees is known to be NP-hard [2], the binarization of data can simplify the growing of trees. However, there are multiple complicated nonlinear relations in real-world data, not just a simple linear relation, binarization might lead to loss of information [15], [16]. Reshef *et al.* presented a novel estimator for two variables called maximal information coefficient (MIC) [17]. This algorithm could search the approximate optimal split by unequal interval optimizing and can capture a wide range of associations, both linear and non-linear. In this paper, we employed the improved MIC algorithm, Chi-MIC [18], [19], to find suitable splitting points for the numeric attributes.

The pruning phase aims to control DT complexity and generalize the DT. The actions of the pruning phase are often referred to as post-pruning (error-based pruning, pessimistic error pruning and, minimum error pruning) in contrast to the pre-pruning (cost-complexity pruning, reduced error pruning). The pruning phase aims to delete unnecessary leaves and avoid overfitting. This procedure is the same as the maximal grid size $B(n)$ of MIC [17] while setting $B(n)$ too high (too many splitting points), such as $B(n)$ equal to n (n is the number of samples), can lead to MIC score equal to "1" even for random data because each data point gets its own cell. In previous works, we used a local chi-square test to determine whether each splitting point is useful, and remove the splitting points that are not significant. The local chi-square test method can significantly control the grid size and improve the performance of MIC [18]. Here we use the backtracking method based on the local chi-square test to determine the branching of each node, and as the stopping criteria for tree-growing.

III. METHODS AND MATERIALS

A. DISCRETIZATION FOR NUMERICAL ATTRIBUTES

Binarization of numerical attributes is not the best discrete method to evaluate the discrimination power of each attribute or discover the complicated association. The MIC method explores various combinations of splitting points by unequal interval optimizing and can capture a wide range of associations. Theoretically, MIC should find the best splitting points for each numerical attribute, however, the approximation algorithm proposed by Reshef *et al.* tend to provide too many splitting points. In our previous work, the Chi-MIC method, based on the local chi-square test, was proposed to resolve this problem by removing splitting points that are not significant [18]. Here, Chi-MIC was used to find splitting points for numerical attributes, and the numerical attribute was discretized.

B. SIMPLIFICATION FOR SPLITTING POINTS

Given a finite set $D = \{Y{:}X\}$, Y is labeled as positive and negative (represented by the values "+" and "−"), attribute X taking a set of discrete, mutually exclusive values (x_1, x_2, x_3, and x_4). It is unnecessary that a discrete attribute is split on all splitting points into many child nodes since it could result in a larger tree. For the set D we can construct a 2×4 contingency table depicted in Figure 1A. While one splitting point is pruned, there are 6 alternative 2×3 contingency tables depicted in Figure 1B, and then the table with maximum χ^2-value is selected as the candidate. For the candidate table (Figure 1C), it transformed form to the 2×3 contingency table by merge the x_2 and x_4. We can use a chi-square test to determine whether the merging is useful. The sample points distributed in the red area (Figure 1C) are used to perform a chi-square test on a 2×2 contingency table. If the p-value of the chi-square test is greater than a given threshold (0.05/0.01), the merging is useful and the algorithm continues pruning for the next splitting point. On the other hand, if the p-value of the chi-square test is lower than the given threshold, the merging is quite unnecessary and the process of simplifying splitting points is terminated. Additionally, if all splitting points of one attribute are merged, the attribute also is pruned.

C. GROWING OF THE TREE

While the splitting points are confirmed for each attribute, the growth phase of the tree includes the following steps: attributes evaluation, attributes introduction and stopping criteria.

Step 1: Attributes evaluation: Let $S = \{X_j\}_{j=1}^{m}$ be a set of attributes, where X_j is the j-th attribute and $|S|$ (= m)

TABLE 1. The detailed information of the data sets.

Datasets	No. of classes	No. of samples in training dataset	No. of features	No. of samples in test dataset
Cardiotocography*	3	1595(1231,224,140)	21	531(424,71,36)
Waveform+noise	3	3750(1254,1258,1238)	40	1250(438,395,417)
Wine	3	134(45,52,37)	13	44(14,19,11)
Forest Types	4	198(54,48,37,59)	27	325(105,38,46,136)
Vehicle	4	635(173,155,158,149)	18	211(45,57,59,50)
Page Blocks*	5	4105(3691,243,22,58,91)	10	1368(1222,86,6,30,24)
First-order Theorem*	6	4588(816,362,560,449,471,1930)	51	1530(273,124,188,168,153,624)
Gas Sensor	6	10433(1917,2203,1249,1431,2242,1391)	128	3477(648,723,392,505,767,442)
Sat Image	6	4435(1072,479,961,415,470,1038)	36	2000(461,224,397,211,237,47)
Glass*	6	161(54,57,14,11,8,17)	9	53(16,19,3,2,1,12)
Segment	7	210(30,30,30,30,30,30,30)	18	2100(300,300,300,300,300,300,300)

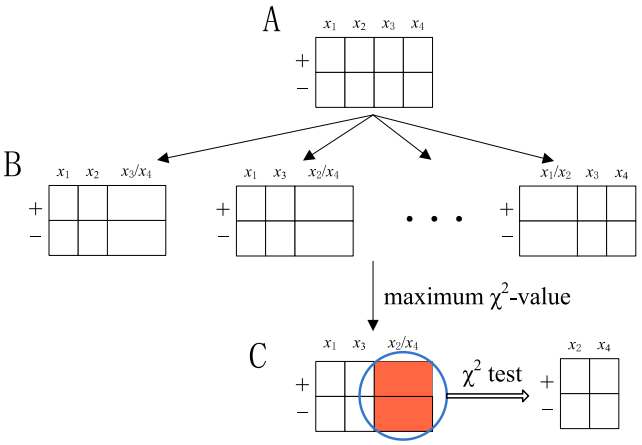

FIGURE 1. Backtracking method for simplifying splitting points based on local chi-square test.

is the number of features in S. The Information Gain Ratio (Y, X_j) [20] is computed for evaluating the discrimination power of each attribute. The attribute with maximum Gain Ratio (having Gain greater than Average Gain) is selected as the first introduced attribute.

Step 2: Attributes introduction: Suppose the current tree including q nodes, we will select additional attributes from the rest attributes set Ω_S in an incremental way: earlier selected attributes remain in the attribute set [21]. Suppose attribute $X_j \in \Omega_S$ with $k-1$ splitting points (X_j containing k values), then X_j splits the current of each node into k child nodes. For these child nodes, the backtracking method based on the local chi-square test (the same as above in part 2.2 Simplification for splitting points) is used to determine the branching of each node. Let U_{hj} $(1 \leq U_{hj} \leq \mathrm{k})$ denote the numbers of branches in the h-th node, and we can get $\sum_{h=1}^{q} U_{hj}$ child nodes and compute the Gain Ratio. At last, the attribute with maximum Gain Ratio (having Gain greater than Average Gain) is selected as the nested attribute.

Step 3: Stopping Criteria: For the q nodes, if $\sum_{j=1}^{|\Omega_s|}\sum_{h=1}^{q} U_{hj} = |\Omega_s| \bullet q$, there will not attributes be introduced and the growth phase is terminated.

D. WEIGHTED DECISION-MAKING FOR IMBALANCED DATA SETS

For the two-class case, if n training samples reach a node, of which $n+$ and $n-$ are positive (Minority Class) and negative instances (Majority class), respectively. Generally, DT classifiers implicitly make decisions by comparing the number of the instances, it risks introducing bias for imbalanced data sets [22]. Here, we use the corrective sample size based on prior probability to make the decision. Let N_+ and N_- are the total number of positive class and negative class samples, the number of corrective class samples are denoted as $n_+^{'} = n_+ \cdot \frac{N_+ + N_-}{2N_+}$ and $n_-^{'} = n_- \cdot \frac{N_+ + N_-}{2N_-}$. For the test instance that reaching this node, if $n_+^{'} > n_-^{'}$, it would be classified as being positive, else as negative.

For multi-class data sets, One-vs-One (OVO) strategy [23] was used to convert multi-class classification into binary classification. In the OVO strategy, an m-class classification problem is transformed into m(m−1)/2 binary classification problems. Then m(m−1)/2 binary CMDT model were trained and used to respectively predict the test sample. The class with the largest number in the prediction results was assigned to the test sample.

E. DATA SETS

11 benchmark data sets with different scales were chosen from UCI Machine Learning Repository [9] to evaluate the CMDT algorithm. The data sets are involved in life sciences, social sciences, and medical sciences. According to the number of samples, the data sets were roughly divided into two groups: 7 balanced data sets and 4 imbalanced data sets. (Table 1) (Asterisked items are imbalanced data sets, others are balanced data sets. The final reserved attributes

TABLE 2. Comparison of testing performance among different models on the seven balanced data sets (%).

Models	Metric	Waveform +noise	Wine	Forest Types	Vehicle	Gas Sensor	Sat Image	Segment	Average
CMDT	*WA*	79.18	100.00	81.55	72.09	97.49	85.61	91.24	**86.74**
	AC	78.96	100.00	82.15	69.67	97.61	86.25	91.24	86.55
CART	*WA*	72.82	86.92	77.59	70.74	82.29	72.35	87.14	78.55
	AC	72.56	86.36	80.00	68.25	83.38	78.30	87.14	79.43
CHAID	*WA*	74.00	85.17	76.33	69.18	95.10	81.45	85.76	81.00
	AC	73.76	84.09	78.77	66.82	95.48	84.60	85.76	81.33
SVM	*WA*	85.71	98.25	78.02	57.21	69.14	78.33	82.33	78.43
	AC	85.44	97.73	80.00	54.50	69.23	83.30	82.33	78.93
QUADRC	*WA*	80.75	100.00	61.46	84.04	98.41	76.10	86.33	83.87
	AC	80.64	100.00	55.08	82.46	98.56	80.50	86.33	83.37
FISHERC	*WA*	85.07	100.00	74.82	74.32	67.54	64.81	87.76	79.19
	AC	84.96	100.00	76.92	72.04	73.74	64.15	87.76	79.94
LOGLC	*WA*	85.75	100.00	79.13	75.89	41.00	78.81	92.38	78.99
	AC	85.60	100.00	81.54	73.93	43.08	83.05	92.38	79.94
NMSC	*WA*	79.40	96.49	79.49	42.38	57.08	78.00	83.90	73.82
	AC	78.64	95.45	80.00	41.23	57.52	78.60	83.90	73.62
QDC	*WA*	81.67	98.25	67.84	83.18	96.67	81.77	76.86	83.75
	AC	81.52	97.73	65.54	81.52	96.58	85.70	76.86	83.63
KNNC	*WA*	85.64	67.95	82.96	62.79	98.69	88.72	87.67	82.06
	AC	85.28	65.91	86.46	60.19	98.68	90.35	87.67	82.08
PARZENC	*WA*	81.95	74.01	82.26	59.24	98.69	88.29	83.38	81.12
	AC	81.76	70.45	84.00	56.40	98.68	90.35	83.38	80.72
NAIVEBC	*WA*	78.67	98.25	79.37	58.47	57.70	80.07	83.67	76.60
	AC	77.92	97.73	79.38	55.92	57.98	80.50	83.67	76.16
BPXNC	*WA*	85.68	98.25	79.45	80.77	33.20	73.42	76.10	75.27
	AC	85.52	97.73	82.46	79.15	41.01	80.85	76.10	77.54

for each dataset by CMDT were listed in supplement table (Table S1: https://github.com/chenyuan0510/CMDT/blob/master/tableS1.xlsx)).

IV. RESULT

A. COMPARISON OF INDEPENDENT PREDICTION ACCURACY AMONG DIFFERENT MODELS

We used 12 reference models, Classification and Regression Trees (CART) by package of *rpart* for *R* [24], Chi-square Automatic Interaction Detection (CHAID) by package of *party* for *R* [25], Support Vector Machine (SVM) by LIBSVM tools [26], Quadratic classifier (QUADRC) [27], Fisher's least square linear classifier (FISHERC) [28], [29], [30], Logistic linear classifier (LOGLC) [29], [31], Nearest mean scaled classifier (NMSC) [32], Quadratic bayes normal classifier (QDC) [33], [29], k-nearest neighbor classifier (KNNC) [34], Parzen classifier (PARZENC) [35], Naive bayes classifier (NAIVEBC) [36], Back-propagation trained feed-forward neural network classifier by back-propagation (BPXNC) [37], to evaluate the performance of CMDT, the latter nine methods were performed based on the *PRTools* [38]. The Weighted Accuracy (*WA*) [39] was used as the performance metric. The WA defined as follows:

$$WA = \frac{\sum_{i=1}^{C} AC_i}{C} \tag{1}$$

Here, C is the number of classes. AC_i denotes the accuracy of the i-th class.

The test accuracy (*AC*) and *WA* for the seven balanced datasets and four imbalanced datasets are listed in Table 2 and Table 3 respectively. From Table 2, we can find that the best models based on average *WA* are CMDT (86.74%), QUADRC (83.87%), QDC (83.75%), KNNC (82.06%), PARZENC (81.12%), and CHAID (81.00%). On the balanced data sets, the *WA* of CMDT is slightly better than that of QUADRC. However, in the comparison of three decision tree models (CMDT, CART, and CHAID), the proposed CMDT method is superior to the CART and CHAID, as the average *WA* of CMDT is 8.19% and 5.74% higher than that of CART and neural CHAID (The p values of paired t-test of CMDT-CART and CMDT-CHAID are 0.008 and

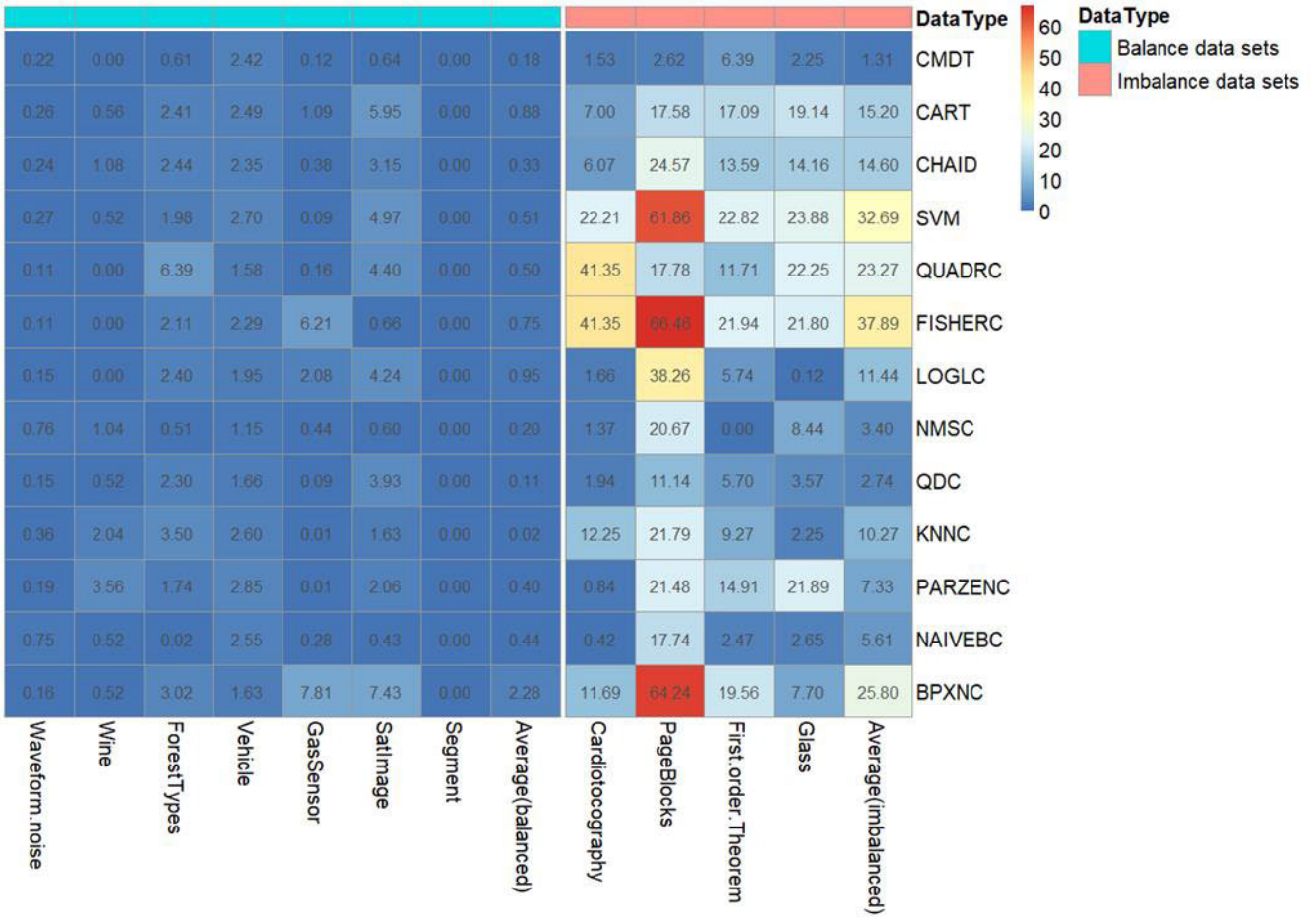

FIGURE 2. The difference (%) between AC and WA for two groups data sets.

0.01 respectively). For paired t-test of CMDT and CART about the WA on the 7 balanced data sets, there are seven paired WA values for the 7 balanced data sets by the CMDT and CART algorithms. The paired t-test is to test whether there is a difference between the two algorithms on the performance of classification. For the paired t-test of AC and WA on 7 balanced data sets, we first calculated the average of the WA scores and AC scores of all models on one data set, then we could get 7 paired WA vs. AC scores. This paired t-test is to test whether there is a difference between the two performances of metrics.

As described in Table 3, on imbalanced data sets, the CMDT model outperforms other reference models drastically, as the average *WA* is 77.14%, this value is 8.28% and 14.44% higher than that of KNNC (the second-highest model) and CHAID (the third-highest model). However, the *WA* of CMDT is only 2.87% and 2.99% higher than that of the second-highest and third-highest model on balanced data sets. Furthermore, the standard deviations of average *WA* among all models on imbalanced and balanced data sets are 11.95% and 3.91% respectively. We can find that the CMDT should perform better on imbalanced data sets compare to balanced data sets.

As shown in Fig. 2, We further compare the difference of *WA* and *AC* on the two group data sets. For the balanced data sets, there is a slight variation between *AC* and *WA* of all models. The average difference between *AC* and *WA* of all models on balanced data sets is only 1.40±1.7%. The *p*-value of paired t-test between average *AC* and average *WA* is 0.1810. This indicates that there is no significant difference between the two metrics on balanced data sets. Such as the ''Segment'' dataset (each class with 30 samples in the training dataset), the *AC* is the same score as the *WA* for all models. In contrast, there is a distinct variation between *AC* and *WA* for the whole models on imbalanced data sets except the CMDT model, and in most cases, there are *AC* score greater than the *WA* score. The average difference between *AC* and WA for these 12 models on imbalanced data sets is up to 16.27±15.78%, the *p*-value of paired t-test between average *AC* and average *WA* is 0.0028. This indicates that the two metrics are significantly different for these 12 models on imbalanced data. Such as the *AC* of the FISHERC model

TABLE 3. Comparison of testing performance among different models on the four imbalanced data sets (%).

Models	Metric	Cardiotocography	PageBlocks	First-order Theorem	Glass	Average
CMDT	*WA*	92.87	92.92	45.05	77.72	**77.14**
	AC	91.34	95.54	51.44	75.47	78.45
CART	*WA*	87.35	79.28	30.50	52.56	62.42
	AC	94.35	96.86	47.58	71.70	77.62
CHAID	*WA*	87.91	72.06	37.06	53.76	62.70
	AC	93.97	96.64	50.65	67.92	77.30
SVM	*WA*	64.80	29.36	18.95	34.61	36.93
	AC	87.01	91.23	41.76	58.49	69.62
QUADRC	*WA*	39.81	76.59	22.15	49.45	47.00
	AC	81.17	94.37	33.86	71.70	70.27
FISHERC	*WA*	39.81	24.19	20.94	46.13	32.77
	AC	81.17	90.64	42.88	67.92	70.65
LOGLC	*WA*	84.41	48.51	34.46	56.49	55.97
	AC	86.06	86.77	40.20	56.60	67.41
NMSC	*WA*	72.45	63.25	26.80	57.49	55.00
	AC	73.82	83.92	26.80	49.06	58.40
QDC	*WA*	72.64	82.43	34.06	54.92	61.01
	AC	74.58	93.57	28.37	58.49	63.75
KNNC	*WA*	77.58	73.75	46.35	77.72	68.85
	AC	89.83	95.54	55.62	75.47	79.12
PARZENC	*WA*	83.52	42.10	36.92	76.61	59.78
	AC	82.67	20.61	51.83	54.72	52.46
NAIVEBC	*WA*	79.90	64.35	34.32	59.61	59.55
	AC	79.47	82.09	36.80	62.26	65.16
BPXNC	*WA*	78.51	26.33	25.74	64.00	48.65
	AC	90.21	90.57	45.29	71.70	74.44

on the 'PageBlocks' dataset is 66.45% greater than the *WA*. However, unlike these 12 reference models, there is a slight difference between average *AC* (78.45%) and average *WA* (77.14%) for CMDT on imbalanced data sets, the *p*-value of paired t-test between *AC* and *WA* on the four imbalanced data sets is 0.5609.

To sum up, on imbalanced data sets, these 12 reference models have the higher *AC* but lower *WA*, the performance bias occurs in the imbalance data. However, the CMDT method performs well on both two metrics

B. CONVERTING BALANCED DATA INTO IMBALANCED DATA

Compare to balanced data sets, we can find that the CMDT indeed performs better on imbalanced data sets. Here we further convert the balanced data sets into imbalanced data set to verify this conclusion. For the seven balanced data sets, the ''Wine'', ''Vehicle'' and ''Forest Types'' have a lesser number of samples. Therefore, ''Waveform+noise (WN)'', ''Gas Sensor (GS)'', ''Sat Image (SI)'' and ''Segment (SE)'' are selected for further evaluation. For each data set, we first merge the training and test samples, and then randomly sample some classes as the majority class, the other classes as the minority class. We use three imbalance ratios in the training set (2:1, 5:1, and 10:1). The training samples and test samples in a 3:1 ratio.

Each experiment is repeated 10 times, and the average results are shown in Table 4. For the average *WA* of each method for three ratios on four datasets, the CMDT method performs best, as it is 88.21%. As illustrated in Figure 3, there is a slight variation between *AC* and *WA* of the CMDT method on all three ratios. However, most methods have high *AC*, but low *WA*, especially in the case of a serious imbalance (10:1). Although the *WA* of NAIVEBC, PARZENC, and NMSC are higher than the *AC* of them, these methods worse than the CMDT method, as they have lower average *WA* scores (76.47%, 76.16%, and 73.87%). Figure 4 illustrates the differences between *AC* and *WA* of all methods for a ratio of 10:1 on four data sets. Except for NAIVEBC, PARZENC, and NMSC methods, the CMDT method has the lowest difference (the average difference is 3.1%). The average differences for the CART, CHAID, SVM, QUADRC, FISHERC, QDC, KNNC,

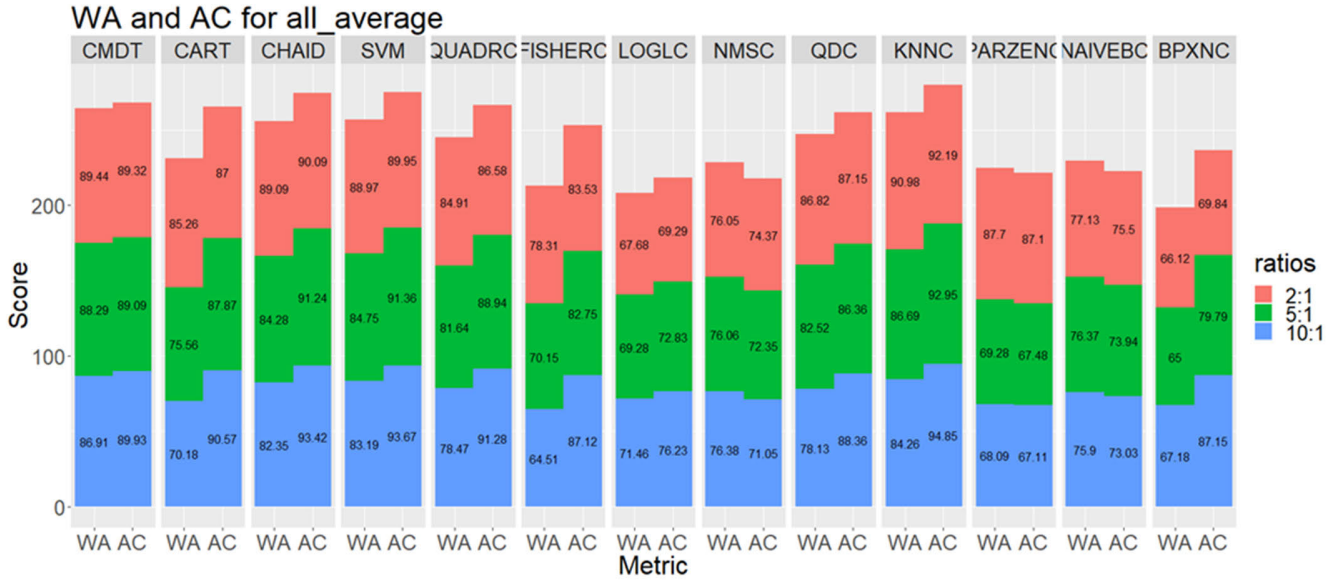

FIGURE 3. The average *WA* and *AC* of each models for three ratios.

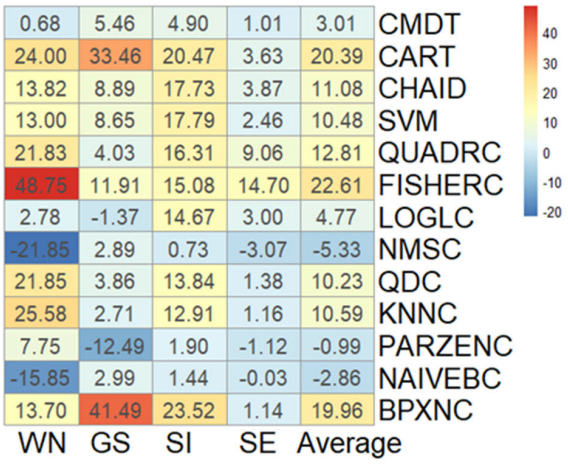

FIGURE 4. The differences between *AC* and *WA* (%) on ratio of 10:1.

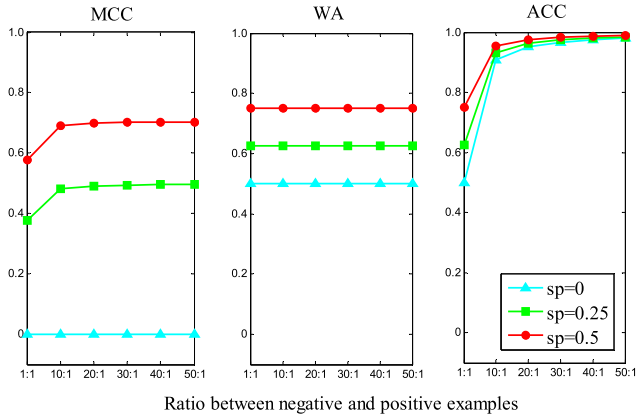

FIGURE 5. The performance of *MCC*, *WA* and *AC* on ratio of 10:1.

BPXNC are all higher than 10%. For example, the difference of CMDT on "WN" is only 0.68%, but differences of FISHERC, KNNC, CART, QDC, and QUADRC on this dataset are 48.75%, 25.58%, 24.00%, 21.85%, and 21.83%, respectively. The LOGLC method has a slight variation between *AC* and *WA* in a ratio of 10:1 but has a lower *WA* score (average *WA* is 71.46%).

V. DISCUSSIONS

A. COMPARISON FOR PERFORMANCE METRICS

AC as a common measure for determining the performance of prediction methods is sensitive to the class distribution of the data set. It will prefer to the majority class, and we should determine with caution whether higher *AC* means a better global accuracy [40]. Matthews Correlation Coefficient (*MCC*) [41] and G-mean [42] are independent of the class distribution in the data set, but the *MCC* only for binary classification, and the G-mean tend to be zero while all samples of one minority class are misclassified. Here, we compared the performance of *AC* and *WA* on simulated binary datasets by referring to *MCC*. For the simulated binary datasets, the sensitivity (accuracy of positive class) is fixed to 1.0 (100%), the specificity (accuracy of negative class) is set to three levels (sp = 0, 0.25, and 0.5).

As the showed in Figure 5, there is a slight increase in *MCC* values as the degree of imbalance increases. As expected, the *AC* values increase rapidly and converge to 100% with an increasing negative sample size. On the contrary, the *WA* values can hold constant as imbalance is growing. So, in this paper, the *WA* is used as the main evaluation measure for each model. The results regarding the classification accuracy also confirm that higher *AC* does not mean better global accuracy in the case of the imbalance.

TABLE 4. The average *WA* and *AC* of each models for three ratios on the four datasets (%).

Models	Metric	2:1				5:1				10:1				Average
		WN	GS	SI	SE	WN	GS	SI	SE	WN	GS	SI	SE	
CMDT	*WA*	78.32	96.61	86.31	96.52	79.23	94.00	85.40	94.54	78.20	90.58	84.77	94.10	**88.21**
	AC	77.20	97.00	86.94	96.14	77.36	96.12	87.53	95.36	78.88	96.04	89.67	95.11	89.45
CART	*WA*	83.13	82.25	83.58	92.09	73.25	65.54	74.52	88.92	68.00	54.94	72.04	85.75	77.00
	AC	85.58	84.49	86.57	91.35	88.11	84.96	89.33	89.08	92.00	88.40	92.51	89.38	88.48
CHAID	*WA*	84.12	93.64	85.48	93.10	80.97	89.60	77.64	88.90	80.22	86.82	75.37	86.98	85.24
	AC	85.70	94.76	87.62	92.28	90.07	94.43	90.03	90.45	94.04	95.71	93.10	90.85	91.59
SVM	*WA*	84.22	93.53	85.00	93.14	81.10	89.40	77.99	90.51	81.17	87.26	75.45	88.90	85.64
	AC	85.68	94.62	87.24	92.28	89.93	94.39	90.02	91.08	94.17	95.91	93.24	91.36	91.66
QUADRC	*WA*	80.32	98.25	76.13	84.95	74.25	97.36	75.89	79.07	68.05	94.10	74.04	77.70	81.67
	AC	82.28	98.48	80.07	85.50	84.68	98.47	86.98	85.65	89.88	98.13	90.35	86.76	88.93
FISHERC	*WA*	68.90	95.46	65.92	82.96	42.17	94.85	66.33	77.26	35.00	83.47	66.27	73.31	70.99
	AC	76.63	96.24	75.72	85.54	75.21	97.18	73.05	85.54	83.75	95.38	81.35	88.01	84.47
LOGLC	*WA*	85.70	25.90	80.32	78.82	85.87	47.84	77.30	66.09	85.85	42.07	76.48	81.43	69.47
	AC	85.78	29.50	84.05	77.83	87.43	52.68	87.82	63.38	88.63	40.70	91.15	84.43	72.78
NMSC	*WA*	80.27	58.08	77.74	88.12	81.52	58.10	77.86	86.76	81.18	58.05	78.67	87.62	76.16
	AC	73.25	59.09	78.38	86.74	66.39	59.06	79.35	84.59	59.33	60.94	79.40	84.55	72.59
QDC	*WA*	81.58	96.76	82.16	86.76	75.33	95.92	80.33	78.51	67.57	91.90	76.69	76.35	82.49
	AC	82.08	96.38	85.47	84.66	83.43	95.89	88.17	77.94	89.42	95.76	90.53	77.73	87.29
KNNC	*WA*	80.50	98.59	88.93	95.89	72.03	97.54	84.40	92.77	65.80	95.69	81.21	94.35	87.31
	AC	84.30	98.63	89.99	95.83	87.50	98.44	91.77	94.09	91.38	98.40	94.12	95.51	93.33
PARZENC	*WA*	81.55	nan	88.80	92.75	81.17	16.67	87.59	91.68	74.92	16.67	87.57	93.22	73.87
	AC	80.53	nan	89.01	91.75	82.07	7.13	89.47	91.24	82.67	4.18	89.47	92.10	72.69
NAIVEBC	*WA*	80.27	58.82	80.44	88.99	81.73	58.52	80.17	85.08	81.18	56.69	80.75	84.97	76.47
	AC	73.73	59.38	80.81	88.07	68.14	59.66	81.67	86.28	65.33	59.68	82.19	84.94	74.16
BPXNC	*WA*	84.67	19.32	65.53	94.98	81.60	31.39	53.25	93.75	78.63	27.70	68.10	94.31	66.10
	AC	85.80	26.35	72.27	94.94	89.71	57.59	77.08	94.78	92.33	69.19	91.62	95.45	78.93

B. SPLITTING CRITERIA BASED ON MIC

For the DT algorithm to be successful, accurate characterization of the attribute is necessary. Binarization of numerical data simplifies the growing of trees and provides simple logical functions explaining the role of attributes. However, binarization might lead to the loss of information for multiple complicated nonlinear associations in real-world data. As shown in Figure 6, the first feature of ''Forest Types'' is a typical example for binarization (Figure 6A), on the contrary, quadripartition of the tenth feature of ''Segment'' is necessary to distinguish Class 1 from Class 7 (Figure 6B). CMDT algorithm uses MIC algorithm as the splitting criteria, can capture both linear and non-linear associations illustrated in Figure 6.

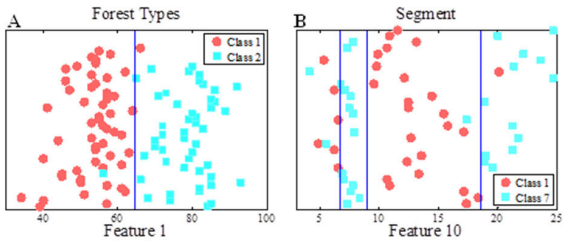

FIGURE 6. Examples of scatter plots of discretization for numerical data.

C. WEIGHTED DECISION-MAKING TABLE

In real-world applications, a class imbalance problem usually occurs. The classifier is usually biased toward the majority class. However, the minority class usually is the class of interest and more important. Existing methods such as data-level [43], algorithmic-level [44],. and cost-sensitive learning [45], [46] have been proposed to address this problem. In the CMDT method, the weighted decision-making table, a cost-sensitive-like method, is used to reduce the biased performance toward majority class examples. For the decision-making table that comes from the child nodes of DT, we generally assume an equal weight for training samples of minority and majority class. This should take some responsibility for the bias in classification. Here, we weighted the sample as described in section 2.4, and making a prediction based on the weighted decision-making table. Taking the

TABLE 5. Comparison of *AC* among raw decision-making table and weighted decision-making table.

Class	No. of samples in training dataset	AC (%)	
		Raw decision-making table	Weighted decision-making table
class1	3691	98.61	96.24
class2	243	82.56	88.37
class3	22	83.33	100.00
class4	58	93.33	96.67
class5	91	70.83	83.33

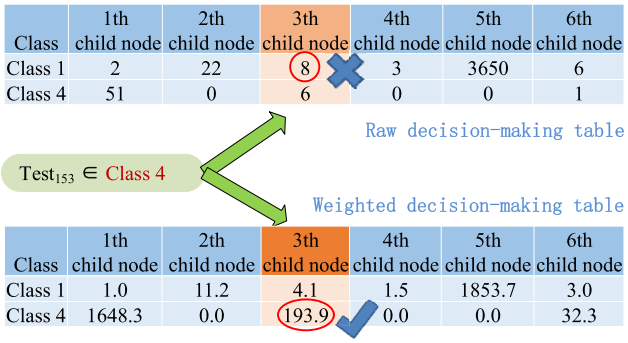

FIGURE 7. Test153 weighted before and after comparison.

classification of Class 1 and Class 4 in PageBlocks dataset as an example, the $Test_{153}$ should be misclassified belong to Class 1 by the raw decision-making table (Figure 7). After these training samples were weighted, the $Test_{153}$ could be classified belong to Class 4 correctly (Figure 7). For the entire dataset of PageBlocks, while the predictor is based on the raw decision-making table, the *WA* and *AC* are 85.73% and 96.93%, respectively, while the predictor is based on the weighted decision-making table the *WA* and *AC* are 92.92% and 95.95%, respectively. The prediction accuracy for each category is listed in Table 5. Obviously, the predictor suffers from performance bias toward the majority class (Class 1) when using the raw decision-making table, it has a high *AC* (96.93%), but low *WA* (85.73%). The weighted decision-making table can avoid performance bias (*WA* = 92.92%). It's worth mentioning that the CMDT can also reduce bias better in classification compare to the reference models, even if based on the raw decision-making table. As shown in Table 3, for the 12 reference models, the highest *WA* is only 82.43% (QDC), but the lowest *WA* is as low as 24.19% (FISHERC).

VI. CONCLUSION

Machine learning technology has been evolving over time from so-called conventional methods to the recent deep learning methods, and many studies have shown that deep learning (DL) can be more accurate and robust than the conventional machine learning approaches in many applications. However, the black box-like prediction from deep learning makes it weak in comprehensibility. Sometimes, the stronger comprehensibility of machine learning is critical to interpret the output and correlate it with the attributes. Decision trees are comprehensible, but at the cost of relatively lower prediction accuracy compared to DL methods. In this study, we proposed a Chi-MIC-based adaptive multi-branch decision tree to improve their classification performance, which can also handle mixed-type attributes by introducing the unequal interval optimization for node splitting, as well as the local chi-square test for tree pruning. The outstanding simulation results show that CMDT can be more reliable than the twelve comparative approaches, especially for imbalanced datasets.

It is important to note that, the CMDT algorithm evaluates attributes by optimizing the splitting points for each attribute based on the MIC approach. However, the MIC is a typical computationally intensive method. It will lead to the limited practicality of the algorithm in the case of big-sample and multi-attribute situations. Therefore, developing parallel algorithms, or faster MIC approximation algorithms may be helpful for handling a wider range of domain problems. In addition, in most applications, random forests integrated by decision trees can achieve better performance than decision tree itself. It is a significant and urgent problem that developing a random forest algorithm with higher prediction accuracy and strong interpretability based on the CMDT algorithm.

REFERENCES

[1] Q. Zou and Q. Liu, "Advanced machine learning techniques for bioinformatics," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 4, pp. 1182–1183, Jul. 2019.
[2] S. B. Kotsiantis, "Decision trees: A recent overview," *Artif. Intell. Rev.*, vol. 39, no. 4, pp. 261–283, Apr. 2013.
[3] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986.
[4] X. J. Chen, Z. G. Zhang, and Y. Tong, "An improved ID3 decision tree algorithm," *Adv. Mater. Res.*, vols. 962–965, pp. 2842–2847, Jun. 2014.
[5] D. Thakur, N. Markandaiah, and D. S. Raj, "Re optimization of ID3 and C4.5 decision tree," in *Proc. Int. Conf. Comput. Commun. Technol.*, Sep. 2010, pp. 448–450.
[6] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, San Mateo, CA, USA, 2017, pp. 3146–3154.
[7] P. Tzirakis and C. Tjortjis, "T3C: Improving a decision tree classification algorithm's interval splits on continuous attributes," *Adv. Data Anal. Classification*, vol. 11, no. 2, pp. 1–18, Jun. 2017.
[8] F. Hammann and J. Drewe, "Decision tree models for data mining in hit discovery," *Expert Opinion Drug Discovery*, vol. 7, no. 4, pp. 341–352, Apr. 2012.

[9] *UCI Machine Learning Repository*. Accessed: Feb. 14, 2018. [Online]. Available: http://archive.ics.uci.edu/ml/
[10] F. Wang, Q. Wang, F. Nie, W. Yu, and R. Wang, "Efficient tree classifiers for large scale datasets," *Neurocomputing*, vol. 284, pp. 70–79, Apr. 2018.
[11] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 3, pp. 660–674, May/Jun. 1991.
[12] M. Sandri and P. Zuccolotto, "A bias correction algorithm for the Gini variable importance measure in classification trees," *J. Comput. Graph. Statist.*, vol. 17, no. 3, pp. 611–628, Sep. 2008.
[13] D. A. Darling, "On the theorems of Kolmogorov-Smirnov," *Theory Probab. Appl.*, vol. 5, no. 4, pp. 356–361, Jan. 1960.
[14] L. M. Sanchez-Brea, J. A. Quiroga, A. Garcia-Botella, and E. Bernabeu, "Histogram-based method for contrast measurement," *Appl. Opt.*, vol. 39, no. 23, pp. 98–106, Aug. 2000.
[15] A. Dimitris, "Computational analysis of the synergy among multiple interacting genes," *Mol. Syst. Biol.*, vol. 3, pp. 83–91, Jul. 2007.
[16] T. M. Ignac, A. Skupin, N. A. Sakhanenko, and D. J. Galas, "Discovering pair-wise genetic interactions: An information theory-based approach," *PLoS ONE*, vol. 9, no. 3, Mar. 2014, Art. no. e92310.
[17] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti, "Detecting novel associations in large data sets," *Science*, vol. 334, no. 6062, pp. 1518–1524, Dec. 2011.
[18] Y. Chen, Y. Zeng, F. Luo, and Z. Yuan, "A new algorithm to optimize maximal information coefficient," *PLoS ONE*, vol. 11, no. 6, Jun. 2016, Art. no. e0157567.
[19] Y. Zeng, H. Yuan, Z. Yuan, and Y. Chen, "A high-performance approach for predicting donor splice sites based on short window size and imbalanced large samples," *Biol. Direct*, vol. 14, no. 1, pp. 1–5, Dec. 2019.
[20] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986.
[21] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
[22] C. Elkan, "The foundations of cost-sensitive learning," in *Proc. 17th Int. Joint Conf. Artif. Intell.*, 2001, pp. 973–978.
[23] R. Debnath, N. Takahide, and H. Takahashi, "A decision based one-against-one method for multi-class support vector machine," *Pattern Anal. Appl.*, vol. 7, no. 2, pp. 164–175, Jul. 2004.
[24] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Wadsworth, OH, USA: Chapman & Hall, 1984.
[25] *CHAID*. Accessed: Jun. 16, 2018. [Online]. Available: https://r-forge.rproject.org/R/?group_id=343
[26] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.
[27] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning," *J. Roy. Stat. Soc.*, vol. 167, no. 1, p. 192, Feb. 2004.
[28] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, 2nd ed. New York, NY, USA: Wiley, 2001.
[29] A. Webb, *Statistical Pattern Recognition*. New York, NY, USA: Wiley, 2002.
[30] S. Raudys and R. P. W. Duin, "On expected classification error of the Fisher linear classifier with pseudo-inverse covariance matrix," *Pattern Recognit. Lett.*, vol. 19, nos. 5–6, pp. 385–392, Apr. 1998.
[31] P. R. Krishnaiah and L. N. Kanal, *Classification, Pattern Recognition and Reduction of Dimensionality*. Amsterdam, North Holland: Oxford, 1982, pp. 169–191.
[32] *Nearest Mean Scaled Classifier*. Accessed: Jun. 20, 2018. [Online]. Available: http://prtools.tudelft.nl/prhtml/prtools/nmsc.html
[33] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley, 2001.
[34] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 6, pp. 607–616, Jun. 1996.
[35] T. Lissack and K.-S. Fu, "Error estimation in pattern recognition via L_α-distance between posterior density functions," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 1, pp. 34–45, Jan. 1976.
[36] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Mach. Learn.*, vol. 29, no. 2, pp. 131–163, Nov. 1997.
[37] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.
[38] R. P. W. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D. M. J. Tax, and S. Verzakov, "PRTools4.1, A MATLAB toolbox for pattern recognition," Delft Univ. Technol., Delft, The Netherlands, Tech. Rep., 2007.
[39] P. K. Meher, T. K. Sahu, and A. R. Rao, "Prediction of donor splice sites using random forest with a new sequence encoding approach," *BioData Mining*, vol. 9, no. 1, pp. 1–25, Dec. 2016.
[40] I. Chaabane, R. Guermazi, and M. Hammami, "Enhancing techniques for learning decision trees from imbalanced data," *Adv. Data Anal. Classification*, vol. 14, no. 3, pp. 677–745, Sep. 2020.
[41] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using Matthews correlation coefficient metric," *PLoS ONE*, vol. 12, no. 6, Jun. 2017, Art. no. e0177678.
[42] J. Ri and H. Kim, "G-mean based extreme learning machine for imbalance learning," *Digit. Signal Process.*, vol. 98, Mar. 2020, Art. no. 102637.
[43] K. Ahlawat, A. Chug, and A. P. Singh, "Benchmarking framework for class imbalance problem using novel sampling approach for big data," *Int. J. Syst. Assurance Eng. Manage.*, vol. 10, no. 4, pp. 824–835, Aug. 2019.
[44] X. Zhang, R. Li, B. Zhang, Y. Yang, J. Guo, and X. Ji, "An instance-based learning recommendation algorithm of imbalance handling methods," *Appl. Math. Comput.*, vol. 351, pp. 204–218, Jun. 2019.
[45] B. S. Raghuwanshi and S. Shukla, "Class-specific cost-sensitive boosting weighted ELM for class imbalance learning," *Memetic Comput.*, vol. 11, no. 3, pp. 263–283, Sep. 2019.
[46] A. Iranmehr, H. Masnadi-Shirazi, and N. Vasconcelos, "Cost-sensitive support vector machines," *Neurocomputing*, vol. 343, pp. 50–64, May 2019.

JIAHAO YE received the B.S. degree in bioinformatics from Hunan Agricultural University, Changsha, China, in 2019, where he is currently pursuing the M.S. degree in bioinformatics with the College of Plant Protection. His research interest includes pattern recognition.

JINGJING YANG received the B.S. and M.S. degrees in bioinformatics from Hunan Agricultural University, Changsha, China, in 2009 and 2020, respectively. Her research interest includes machine learning.

JIANG YU received the B.S. and M.S. degrees in bioinformatics from Hunan Agricultural University, Changsha, China, in 2017 and 2020, respectively. His research interest includes pattern recognition.

SIQIAO TAN received the Ph.D. degree in plant protection from Hunan Agricultural University, in 2008. He is currently a Professor with the College of Information Intelligence, Hunan Agricultural University. His research interest includes deep learning and application.

FENG LUO (Senior Member, IEEE) received the Ph.D. degree in computer science from The University of Texas at Dallas, in 2004. He is currently a Full Professor with the School of Computing, Clemson University. His research interests include deep learning and application, high throughput biological data analysis, data-intensive bioinformatics, network biology, and computational genomics and genetics.

ZHEMING YUAN received the B.S. degree in biology from Hunan Normal University, in 1992, the M.S. degree in plant protection from Hunan Agricultural University, in 1995, and the Ph.D. degree in agricultural entomology and pest control from Zhejiang University, in 2000. He is currently a Professor with the College of Plant Protection, Hunan Agricultural University. His research interests include pattern recognition and machine learning.

YUAN CHEN received the B.S., M.S., and Ph.D. degrees in bioinformatics from Hunan Agricultural University, Changsha, China, in 2009, 2012, and 2016, respectively. He is currently an Associate Professor with the College of Plant Protection, Hunan Agricultural University. His research interests include pattern recognition, deep learning and application, and computational biology.