IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 26, 2020, accepted January 18, 2021, date of publication January 22, 2021, date of current version February 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053778

Instance-Based Classification Through Hypothesis Testing

ZENGYOU HE[1,2], CHAOHUA SHENG[2], YAN LIU[2], AND QUAN ZOU[3], (Senior Member, IEEE)

[1]Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian University of Technology, Dalian 116620, China
[2]School of Software, Dalian University of Technology, Dalian 116620, China
[3]Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu 610054, China

Corresponding author: Zengyou He (zyhe@dlut.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61972066 and Grant 61572094, and in part by the Fundamental Research Funds for the Central Universities under Grant DUT20YG106.

ABSTRACT Classification is a fundamental problem in machine learning and data mining. During the past decades, numerous classification methods have been presented based on different principles. However, most existing classifiers cast the classification problem as an optimization problem and do not address the issue of statistical significance. In this paper, we formulate the binary classification problem as a two-sample testing problem. More precisely, our classification model is a generic framework that is composed of two steps. In the first step, the distance between the test instance and each training instance is calculated to derive two distance sets. In the second step, the two-sample test is performed under the null hypothesis that the two sets of distances are drawn from the same cumulative distribution. After these two steps, we have two p-values for each test instance and the test instance is assigned to the class associated with the smaller p-value. Essentially, the presented classification method can be regarded as an instance-based classifier based on hypothesis testing. The experimental results on 38 real data sets show that our method is able to achieve the same level performance as several classic classifiers and has significantly better performance than existing testing-based classifiers. Furthermore, we can handle outlying instances and control the false discovery rate of test instances assigned to each class under the same framework.

INDEX TERMS Classification, hypothesis testing, two-sample testing, machine learning.

I. INTRODUCTION

Classification is a fundamental data analysis procedure, which is ubiquitously used across different fields. Thousands of classification algorithms (classifiers) have been developed during the past decades [1]. These classifiers range from simple models such as k-nearest neighbor (k-NN) [2] to more sophisticated models such as support vector machine (SVM) [3] and random forests [4].

Despite the advances in the development of new classifiers, no single classification algorithm can always achieve the best performance on all data sets [1]. This indicates that different classifiers are complementary to each other in different contexts. Therefore, it is still necessary to develop new and alternative classifiers based on some principles that remain unexplored.

The motivation behind this research is based on the following observations. First, existing non-lazy classifiers typically formulate the classification problem as an optimization problem. Such optimization-based learning strategies can always generate the target classifiers, regardless of the statistical significance of learned models. Second, classifiers such as logistic regression are able to provide probability values for categorizing an unknown test instance. However, it is not an easy task to determine a universal probability threshold to ensure that the classification of the test instance into the corresponding class is statistically significant. Last but not least, existing classifiers cannot control the number of misclassified test instances in terms of metrics such as the false discovery rate (FDR) [5]. Such capability is quite important in the scenario of biological data analysis, in which the prediction results will be further validated by wet-lab experiments that can be costly and time-consuming [6]. Thus, we need to add some notion of statistical significance to classifiers.

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

In fact, the classification problem has already been formulated as a hypothesis testing issue in [7]. More recently, several research efforts [8], [9] further extend the initial formulation in [7] from different aspects. However, the following observations motivate this research. First of all, existing testing-based classification methods deserve certain theoretical drawbacks, as discussed and summarized in Section II. Second, only simulation data sets and several small real data sets have been empirically tested, making it difficult to convince people on the practical usage of such testing-based formulation. Third, the connection between this new formulation and existing classification methods has never been discussed. Finally, the potential benefit of the testing-based classification model remains unexplored.

Based on the above observations, we present a new testing-based classification formulation, in which the null hypothesis is that, informally, the test instance does not belong to any class. To precisely define the null hypothesis, we focus on the classification problem in a two-class setting. First, we can calculate the distance between the test instance and each training instance in the training data set. In this way, we will generate two sets of distances for one test instance to be classified. Then, the hypothesis testing issue can be casted as a two-sample testing problem [10], in which each sample corresponds to a set of distances. In this formulation, the null hypothesis is that two sets of distances are drawn from the same cumulative distribution.

Two-sample testing is a fundamental problem in statistics. We employ the classical Wilcoxon-Mann-Whitney (WMW) test for quantifying the statistical significance in terms of p-values. To alleviate the effect of outlying and irrelevant training instances, we further apply the WMW test to two distance sets that are generated from k-nearest neighbors (k-NNs) of the test instance.

The testing-based classification formulation has several salient features. First of all, it can provide p-values for each test instance to quantify the statistical significance of classifying this instance to certain classes. Accordingly, we can detect outlying test instances that do not belong to any class if the p-values with respect to all classes are larger than the significance level threshold. Second, we can control the FDR of test instances that are assigned to each class based on their p-values.

We evaluate our method on 38 data sets from the UCI [11] repository and the KEEL-dataset repository [12] with respect to the standard classification task. The experimental results show that our method is able to achieve the same level performance as the state-of-the-art classifiers. Meanwhile, it can handle outlying test instances and control the FDR of test instances assigned to each class in a natural manner.

The main contributions of this paper can be summarized as follows.

(1) The binary classification issue is formulated as a two-sample testing problem. Since two-sample testing is a fundamental problem in statistics and many well-known tests are available in the literature, it can be expected that we may introduce many effective testing-based classifiers in the near future.

(2) The classification model that integrates hypothesis testing and the k-NN method is presented. This formulation can alleviate the effect of outlying and irrelevant training instances to improve the classification accuracy significantly.

(3) A comprehensive performance comparison over 38 real data sets is conducted. The experimental results demonstrate the fact that the testing-based classifier is able to achieve the same level performance as standard classifiers such as SVM and decision tree.

(4) Some interesting connections between our testing-based classifiers and existing classification methods are presented.

(5) The advantage of the testing-based classification model on handling outliers and controlling the type I error rate in terms of FDR is empirically investigated.

The rest of this paper is organized as follows. Section II discusses some previous works that are related to our method. Section III presents the details of our method, followed by experimental results on 38 real data sets in Section IV. Section V discusses the relationship between our method and other approaches. Finally, Section VI concludes this paper.

II. RELATED WORK

A. INSTANCE-BASED LEARNING

Instance-based learning is a lazy learning scheme in which the training instances are simply stored. When a new instance is encountered, a set of similar training instances are retrieved to classify the unknown testing instance. The most basic instance-based method is the k-nearest neighbor algorithm (k-NN) [2], [13], which assigns a new instance to the most common class among its k-NNs in training instances.

Essentially, our method can be considered as an instance-based learning approach since the two-sample test is conducted on the distance sets generated from all training instances or k-NNs. This indicates that it is feasible to apply techniques developed for instance-based learning during the past decades [14]–[16] to further improve our method.

B. CLASSIFICATION BASED ON HYPOTHESIS TESTING

Liao and Akritas [7] introduce a classification method based on hypothesis testing, which is abbreviated to TBC. Suppose there are two classes (positive vs. negative) in the training set, i.e., a binary classification problem, the issue is to allocate a new instance t^* to one of the two classes. The basic idea of TBC is that if t^* is placed into the wrong class, then the difference of two samples will be blurred. To implement this idea, two tests with respect to the equality of the means of two samples are conducted, in which t^* is placed into the set of positive instances and the set of negative instances, respectively. Accordingly, we will obtain two p-values p_+ and p_-, where p_+ (p_-) is generated from the test in which t^* is assumed to belong to the positive (negative) class. If $p_+ < p_-$, then t^* is classified as a positive instance.

Otherwise, t^* will be classified as a negative instance. This method works well when the theoretical p-values can be computed and compared. However, TBC has two deficiencies. First, when the number of features of data set is larger than the sample size of one class, the p-values cannot be computed at all because of the singularity of the sample covariance matrix. Second, when the instances from two classes are well separated, the two p-values will equal to zero so that one test instance cannot be classified.

Ghimire and Wang [8] improve the TBC method by introducing a minimum distance into the method and come up with a new classifier for image pixels. Their new method works well in the context of image pixel classification.

Modarres [17]–[19] studies the properties of squared Euclidean interpoint distances (IPDs) between different samples which are taken from multivariate Bernoulli, multivariate Poisson and multinomial distributions. And he also discusses some applications based on IPDs within one sample and across two samples in different distributions.

Afterwards, Guo and Modarres [9] employ interpoint distances to measure the closeness of the samples and develop a new testing-based classifier for the classification of high dimensional discrete observations, which can be abbreviated to IDC. IDC is capable of classifying high dimensional instances by computing the IPDs between different instances. Several different test statistics based on IPDs have been discussed in [9] and we will take the Baringhaus and Franz (BF) statistic as the example. Given two sets of training instances, i.e., one positive set D^+ and one negative set D^-, IDC first computes the average IPDs within D^+, within D^- and between D^+ and D^-, which are denoted by $\bar{d}_{D^+}$, $\bar{d}_{D^-}$ and $\bar{d}_{D^+D^-}$ respectively. Then, it calculates $BF_0 = 2\bar{d}_{D^+D^-} - \bar{d}_{D^+} - \bar{d}_{D^-}$. Similarly, $BF_1 = 2\bar{d}_{(D^+\cup\{t^*\})D^-} - \bar{d}_{D^+\cup\{t^*\}} - \bar{d}_{D^-}$ and $BF_2 = 2\bar{d}_{D^+(D^-\cup\{t^*\})} - \bar{d}_{D^+} - \bar{d}_{D^-\cup\{t^*\}}$ can be obtained by placing t^* into D^+ and D^-, respectively. Note that $|BF_1 - BF_0|$ ($|BF_2 - BF_0|$) can be used to measure the change in the value of BF when t^* is assigned to D^+ (D^-). Therefore, if $|BF_1 - BF_0| < |BF_2 - BF_0|$, t^* is classified as a positive instance; otherwise, t^* will be labeled as a negative instance.

C. ASYMMETRIC CLASSIFICATION ERROR CONTROL

In binary classification, most classifiers are constructed to minimize the overall classification error, which is a weighted sum of type I error (misclassifying a negative instance as a positive one) and type II error (misclassifying a positive instance as a negative one). However, in many realistic applications, different types of errors are often asymmetric, which have different costs and need to be treated with different weights.

The cost-sensitive classification (CSC) method [20], [21] can solve this problem to some extent. It takes the misclassification costs into consideration and aims to minimize the total cost of both errors. Another method is the Neyman-Pearson (NP) classification [22], which is inspired by classical NP hypothesis testing. It is a novel statistical framework for handling asymmetric type I/II error priorities and can seek a classifier that minimizes the type II error while maintaining the type I error below a user-specified level α [23], [24]. CSC and NP classification are fundamentally different approaches that have their own pros and cons [22]. A main advantage of the NP classification is that it is a general framework that allows users to control type I classification error under α with a high probability.

It is very easy to control the type I error in terms of FDR in our formulation since the p-values of each test instance with respect to different classes will be generated in the classification phase. In other words, such testing-based classification formulation provides a unified framework to control the asymmetric classification error in a natural way.

III. METHOD

A. TWO-SAMPLE TESTING

Given two independent random samples G_X and G_Y, where $G_X = \{x_1, x_2, \ldots, x_m\}$ is drawn from the X population and $G_Y = \{y_1, y_2, \ldots, y_n\}$ is drawn from the Y population, the general two-sample testing problem is concerned with the null hypothesis that the two samples are drawn from identical populations [10]:

$$H_0 : F_X(t) = F_Y(t) \text{ for all } t,$$

where F_X and F_Y are the cumulative distribution functions for the X population and the Y population, respectively.

B. PROBLEM FORMULATION

We consider the binary classification problem, in which the training set D is composed of two disjoint sets D^+ and D^-. $D^+ = \{t_1^+, t_2^+, \ldots, t_m^+\}$ and $D^- = \{t_1^-, t_2^-, \ldots, t_n^-\}$ are called the positive training set and the negative training set, respectively. Given a test instance $(t^*, \hat{y})$ whose class label $\hat{y}$ is unknown, the classification task is to decide its class label (positive vs. negative).

We formulate the binary classification problem as a two-sample testing problem. In this formulation, the first sample G_X is a set of m observations, where the i-th observation is the distance between the test instance t^* and the i-th training instance t_i^+ in D^+, i.e. $G_X = \{x_i | x_i = d(t^*, t_i^+), 1 \le i \le m\}$. Similarly, each observation in the second sample G_Y is the distance between the test instance and each training instance in D^-, i.e. $G_Y = \{y_j | y_j = d(t^*, t_j^-), 1 \le j \le n\}$.

To conduct the standard classification task, we may test the null hypothesis against two alternative hypotheses ($F_X(t) < F_Y(t)$ and $F_Y(t) < F_X(t)$) to obtain two one-sided p-values (p_X and p_Y). If $p_X < p_Y$, we will label t^* as a positive instance. Otherwise, we will classify t^* as a negative instance.

To handle the multi-classification problem with K classes ($K > 2$), we can explore the one-vs-all strategy like existing testing-based classification methods (TBC [7] and IDC [9]). Specifically, we regard the set of instances from one class as the positive training set and the set of instances from the remaining classes as the negative training set. For each of K binary classification problems, we first generate a one-sided

p-value for the corresponding class by conducting the two-sample test where the alternative hypothesis is $F_X(t) < F_Y(t)$. Then, we can assign the test instance to the class that has the smallest p-value.

C. K-NN VARIANTS

In the above problem formulation, the distances to all training instances are utilized in the hypothesis testing. However, the existence of outlying and irrelevant training instances may decrease the classification accuracy. To alleviate this issue, we can conduct the hypothesis testing on two samples that are derived from the k-NNs of the test instance in the training sets.

Under H_0, two natural k-NN variants can be formulated. Similar to the k-NN classifier, the first variant is to directly take the k-NNs of the test instance to generate two samples. The distances from the test instance to these k nearest training instances are divided into two groups according to the class label, where each group corresponds to one sample in our scenario. The second variant is to take k_1 nearest instances from D^+ and retrieve k_2 nearest instances from D^- to generate two distance sets, where $\frac{k_1}{k_2} = \frac{m}{n}$. The rationale behind the second variant is that, if the null hypothesis is true, then the number of k-NNs from each class is proportional to the number of training instances in that class. Since $k_1 = k_2$ when $n = m$, we can take the same number of k-NNs from each class in this case.

D. THE CHOICE OF TESTING METHODS

The testing method for two-sample differences has been extensively investigated in the literature. One widely used test for this issue is the WMW test, which is also called the Mann-Whitney U test or Wilcoxon rank-sum test [25]. To obtain the test statistic in WMW test, G_X and G_Y are merged to form a combined sample $G_Z = \{z_1, z_2, \ldots, z_{m+n}\}$. Then, the observations in G_Z are ordered:

$$z_{(1)} \leq z_{(2)} \leq \ldots \leq z_{(m+n)}.$$

According to the ordered list, R_{i1} is defined as the rank of x_i in G_Z and $R_1 = \sum_{i=1}^{m} R_{i1}$. Then we can get $U_1 = R_1 - \frac{m(m+1)}{2}$. If the null hypothesis H_0 is true, then

$$Z = \frac{U_1 - E(U_1|H_0)}{\sqrt{Var(U_1|H_0)}} \sim N(0, 1),$$

where

$$E(U_1|H_0) = \frac{mn}{2}, Var(U_1|H_0) = \frac{mn(m+n+1)}{12}.$$

Based on the above normal approximation, we can calculate the one-sided p-value to test H_0 against $H_1(F_X(t) < F_Y(t))$ for some t.

In our classification model, the choice of testing method is very flexible since the samples to be tested are unidimensional. That is, we can use any univariate two-sample testing method in our classifier. Therefore, we can also employ the testing methods such as pooled t-test, two-sample Kolmogorov-Smirnov test [26] and precedence test instead of the WMW test. In Section V, we will further show that the use of different testing methods will establish the connection between our formulation and existing classification models.

E. HANDLING OUTLIERS AND FDR CONTROL

As we have argued, the testing-based classification model has the advantage of controlling the FDR of classified test instances and handling outlying instances under the same framework. In general, we will assign the test instance to the class that has the smallest p-value among K p-values, where K is the number of classes. However, it is inappropriate to do so when all K p-values are not significant. Luckily, we can use FDR to tackle this problem. We can obtain K sets of p-values from all test instances because our method returns K p-values to classify every test instance. Every p-value set is firstly sorted in a non-descending order: $p_1 \leq p_2 \leq \ldots \leq p_u$, where u is the number of all test instances. Given a significance level α, let i_{max} be the largest index for which

$$p_i \leq \frac{i \times \alpha}{u}.$$

If $i \leq i_{max}$, then the corresponding test instance will be assigned to the current class. After conducting FDR control on all K p-value sets, we can label the test instances that are not classified to any class as outliers.

IV. EXPERIMENTS

A. DATA SETS AND EXPERIMENTAL SETTINGS

We have conducted experiments on 38 data sets from the UCI [11] repository and the KEEL-dataset repository [12]. Among these data sets, the number of instances ranges from 80 to 10092 and the number of features varies from 2 to 90. All the features in these data sets are numeric and the Euclidean distance is used for measuring the dissimilarity between two instances. Most data sets have less than 10 classes and only six of them have more than 10 classes. The detailed characteristics of these data sets are given in Table 1. Moreover, the instances with missing values are discarded and all the numeric feature values are normalized into the interval [0, 1] in the pre-processing process.

In the experiment, we perform 10-fold stratified cross-validation (CV) and compute an average classification accuracy value for ten folds. For every data set, we repeat the 10-fold CV experiment 10 times and record the average and standard deviation of 10 accuracy values as the final results. For the classifiers to be compared, we also compute their average accuracies and ranks over 38 data sets.

B. ALL INSTANCES VS. K-NNs

In the first experiment, we compare several variants of our formulation to check which one is better in practice. Since our method is a classifier that combines instance-based learning and hypothesis testing, we will use the abbreviation IBT to denote such a classification model. To distinguish different

TABLE 1. The main characteristics of the data sets used in the experiment. If there are some missing values in one data set, the number of instances with missing values is provided inside the parentheses in the second column. The number of features and classes are listed in the third and fourth column, respectively.

Names	#Instances	#Features	#Classes	Links
Appendicitis	106	7	2	KEEL
Balance	625	4	3	UCI, KEEL
Banana	5300	2	2	KEEL
Bands	365(539)	19	2	UCI, KEEL
Bupa	345	6	2	UCI, KEEL
Cleveland	297(303)	13	5	UCI, KEEL
Dermatology	358(366)	34	6	UCI, KEEL
Haberman	306	3	2	UCI, KEEL
Hayes-roth	160	4	3	UCI, KEEL
Heart	270	13	2	UCI, KEEL
Hepatitis	80(155)	19	2	UCI, KEEL
Ionosphere	351	34	2	UCI, KEEL
Iris	150	4	3	UCI, KEEL
Led7digit	500	7	10	UCI, KEEL
Marketing	6876(8993)	13	9	KEEL
Monks-2	432	6	2	UCI, KEEL
Movement_libras	360	90	15	UCI, KEEL
Newthyroid	215	5	3	UCI, KEEL
Page-blocks	5473	10	5	UCI, KEEL
Penbased	10992	16	10	UCI, KEEL
Phoneme	5404	5	2	UCL, KEEL
Pima	768	8	2	UCI, KEEL
Ring	7400	20	2	KEEL
Satimage	6435	36	7	UCI, KEEL
Segment	2310	19	7	UCI, KEEL
Sonar	208	60	2	UCI, KEEL
Spambase	4601	57	2	UCI, KEEL
Spectfheart	267	44	2	UCI, KEEL
Texture	5500	40	11	UCL, KEEL
Thyroid	7200	21	3	UCI, KEEL
Titanic	2201	3	2	KEEL
Twonorm	7400	20	2	KEEL
Vehicle	846	18	4	UCI, KEEL
Vowel	990	13	11	UCI, KEEL
Wdbc	569	30	2	UCI, KEEL
Wine	178	13	3	UCI, KEEL
Winequality-red	1599	11	6	UCI, KEEL
Wisconsin	683(699)	9	2	UCI, KEEL

TABLE 2. The average accuracy and rank over 38 data sets for IBT-U and two k-NN variants (k=3).

Methods	Avg accuracy	Avg rank
IBT-U	0.6841	2.45
IBT-U-K-D	0.8106	1.68
IBT-U-K-S	0.7978	1.87

variants, IBT-U is used to denote the classification model when the Mann-Whitney U test is applied to the distance sets derived from all training instances. Similarly, IBT-U-K is used to denote the classification model in which the distance sets are generated according to k-NNs of the test instance. Furthermore, two k-NN variants are denoted by IBT-U-K-D (k-NNs are obtained **D**irectly without considering the class label) and IBT-U-K-S (k-NNs are obtained **S**eparately from different classes), respectively.

Additionally, the parameter k for two k-NN variants is specified as 3, 5, 7 and 9, respectively. The detailed experimental results on IBT-U and two k-NN variants are given in Appendix Table 6, Appendix Table 7 and Appendix Table 8 respectively. According to these results, we also compute the average accuracy and the average rank for each method over 38 data sets, which are summarized in Table 2 and Table 3.

TABLE 3. The average accuracy and rank over 38 data sets for two k-NN variants when k varies.

k	Avg accuracy		Avg rank	
	IBT-U-K-D	IBT-U-K-S	IBT-U-K-D	IBT-U-K-S
k=3	0.8106	0.7978	1.45	1.55
k=5	0.7927	0.7891	1.50	1.50
k=7	0.7767	0.7801	1.68	1.32
k=9	0.7638	0.7762	1.74	1.26

As shown in Table 2, the performance of IBT-U is much worse than that of two k-NN variants. This indicates that it is plausible to explore the k-NN strategy in the testing-based classification model. As shown in Table 3, the average classification accuracies and ranks of two k-NN variants are quite similar when k varies from 3 to 9. In the forthcoming sections, we will use IBT-U-K-D (k=3) as a representative of our classification method in the performance comparison.

C. OUR METHOD VS. OTHER TESTING-BASED CLASSIFIERS

In the second experiment, we compare our method with two previous methods, TBC [7] and IDC [9], which also use hypothesis testing to solve a classification problem. Their detailed experimental results are given in Appendix Table 9. According to these results, we further record the average accuracy and the average rank for each method over 38 data sets in Table 4.

In the implementation of TBC, we employ the Hotelling's T^2 test as the testing method, which has been utilized in [7]. And we use the Hotelling's T^2 statistics instead of p-values in the classification since the generated p-values are often zeros. In the implementation of IDC, we use the Baringhaus and Franz (BF) statistic as the test statistic and assume equal prior probabilities in spite of unequal sample sizes.

For TBC, the classification accuracies on five data sets (*Cleveland*, *Dermatology*, *Hepatitis*, *Movement_libras* and *Winequality-red*) are N/A because the number of features of these data sets is larger than the sample size of one class so that the Hotelling's T^2 statistics cannot be calculated. As a result, we only use the rest 33 data sets to compute the average classification accuracy. For IDC, it can be applied to all data sets, so we simply compute the average of 38 accuracy values.

From Table 4, we can see that our method can achieve the best performance among these three methods. The reasons are as follows. First, our method only consider the k-NNs of test instance while TBC and IDC utilize all training instances without considering the existence of outlying and irrelevant ones. Second, our method employs a hypothesis testing strategy that is easy to understand and totally different from that used in TBC and IDC.

D. OUR METHOD VS. CLASSIC CLASSIFIERS

In the third experiment, we compare our method with four classic classifiers: k-NN, support vector machine (SVM), decision tree (DT) and nearest centroid classifiers (NC).

TABLE 4. The average accuracy and rank over 38 data sets for IBT-U-K-D (k=3) and two existing testing-based classification methods: TBC, IDC.

Methods	Avg accuracy	Avg rank
TBC	0.5947	2.26
IDC	0.6859	2.21
IBT-U-K-D	0.8106	1.54

TABLE 5. The average accuracy and rank over 38 data sets for IBT-U-K-D (k=3) and four classic classifiers: k-NN (k=3), SVM, decision tree (DT), nearest centroid classifier (NC).

Methods	Avg accuracy	Avg rank
k-NN	0.8146	2.47
SVM	0.7826	2.63
DT	0.8071	2.87
NC	0.7172	4.03
IBT-U-K-D	0.8106	3.00

The detailed experimental results are given in Appendix Table 10 and Appendix Table 11. Meanwhile, the average accuracies and the average ranks of these classifiers over 38 data sets are presented in Table 5.

For k-NN, SVM and DT, we use the functions *fitcknn*, *fitcecoc* and *fitctree* in MATLAB 2018b, respectively. More specifically, *fitcecoc* adopts linear kernel as its kernel function and uses the one-versus-all strategy like our method when the number of classes is larger than 2. *fitctree* is an implementation of the standard CART algorithm [27]. In the experiment, we use the default parameter settings of *fitctree*.

From Table 5, we can see that our method is able to achieve the same level performance as these classic classifiers (SVM, k-NN and DT) and nearest centroid classifier performs worst among these five methods. Concretely, there are 13, 19 and 18 data sets on which our method can produce higher classification accuracies than k-NN, SVM and DT among the 38 data sets, respectively. In a word, our method is competitive to these classic classifiers with respect to the overall performance.

E. HANDLING OUTLIERS THROUGH FDR CONTROL

In the last experiment, we investigate the potential of our method on outlier detection and FDR control. The *Balance* data set from UCI is used as an example, which has 625 instances and three classes (L, B and R). There are 288, 49 and 288 instances in the three classes respectively. If we take a subset of the 576 (288+288) instances from the class L and R as training instances and use the 49 instances from the class B as test instances, then it is obvious that all test instances should be considered as outliers.

We randomly take 80 percent of instances from the class L and R to compose the training set. In order to obtain the average performance, 10 different random training sets are generated. We use IBT-U as the classifier and the significance level for FDR is set to be 0.05. The experimental results show that 48 of 49 test instances can be labeled as outliers on average. Specifically, there are at most two test instances which cannot be labeled as outliers and they are usually different when the training set is different. Therefore, our method is able to recognize outliers and control the FDR of classification results in the same time.

V. RELATIONSHIP TO OTHER APPROACHES

Our classification method is a two-phase approach: two distance sets are first generated and then the two-sample test is conducted. As we have discussed, we may use different significance testing methods in the second phase. In this section, we will show that the use of different testing methods will lead to different classifiers that have close relationship with existing classification models.

A. CONNECTION TO NEAREST CENTROID CLASSIFIER

The nearest centroid (mean) classifier is one of the most widely used instance-based classification models [28]. In the training phase, only the centroid for each class is calculated and stored. In the classification phase, the distance between one unknown instance and each centroid is calculated to find the nearest centroid. Then, this new test instance is assigned to the class of its nearest centroid.

If the pooled t-test is employed as the significance testing procedure in our model, then we can reveal some interesting connections between our method and the nearest centroid classifier. To simplify the analysis, we first consider the scenario of univariate data set and then discuss the case of multivariate data set.

Given two one-dimensional sets $D^+ = \{t_1^+, t_2^+, \ldots, t_m^+\}$ and $D^- = \{t_1^-, t_2^-, \ldots, t_n^-\}$, their centroids (means) can be easily computed by $C_{D^+} = \frac{1}{m}\sum_{i=1}^{m} t_i^+$ and $C_{D^-} = \frac{1}{n}\sum_{j=1}^{n} t_j^-$. Given an unknown instance t^*, the distances between t^* and these two centroids can be measured by $d^+ = |t^* - C_{D^+}|$ and $d^- = |t^* - C_{D^-}|$. The nearest centroid classification method will assign t^* to the positive or the negative class according to whether $d^+ < d^-$.

In our method, two samples $G_X = \{|t^* - t_i^+|, 1 \le i \le m\}$ and $G_Y = \{|t^* - t_j^+|, 1 \le j \le n\}$ are obtained and their means are denoted by $\bar{d}_X = \frac{1}{m}\sum_{i=1}^{m} |t^* - t_i^+|$ and $\bar{d}_Y = \frac{1}{n}\sum_{j=1}^{n} |t^* - t_j^-|$. Then, we test the null hypothesis against two alternative hypotheses ($F_X(t) < F_Y(t)$ and $F_Y(t) > F_X(t)$) on the two samples to obtain two one-sided p-values (p_X and p_Y). At last, our method will assign t^* to the positive (negative) class if $p_X < p_Y$ ($p_X > p_Y$).

Note that when the pooled t-test is employed in our method, we will obtain two t statistics (t_X and t_Y). We can get

$$\begin{aligned} p_X < p_Y &\Leftrightarrow t_X < t_Y \\ &\Leftrightarrow \bar{d}_X - \bar{d}_Y < \bar{d}_Y - \bar{d}_X \\ &\Leftrightarrow \bar{d}_X < \bar{d}_Y. \end{aligned}$$

Similarly, we can also get $p_X > p_Y \Leftrightarrow \bar{d}_X > \bar{d}_Y$. Therefore, our method will assign t^* to the positive class if $\bar{d}_X < \bar{d}_Y$. Otherwise, we will label t^* as a negative instance.

TABLE 6. The detailed experimental results of IBT-U.

Data sets	IBT-U	
	Avg	Std
Appendicitis	0.8557	0.0046
Balance	0.8800	0.0039
Banana	0.5998	0.0017
Bands	0.6405	0.0128
Bupa	0.5574	0.0170
Cleveland	0.5505	0.0048
Dermatology	0.8944	0.0041
Haberman	0.7144	0.0166
Hayes-roth	0.5581	0.0221
Heart	0.8241	0.0047
Hepatitis	0.8088	0.0084
Ionosphere	0.6638	0.0033
Iris	0.9567	0.0047
Led7digit	0.7206	0.0076
Marketing	0.2995	0.0015
Monks-2	0.5185	0.0149
Movement_libras	0.3883	0.0146
Newthyroid	0.8581	0.0025
Page-blocks	0.9043	0.0005
Penbased	0.5566	0.0005
Phoneme	0.7172	0.0008
Pima	0.7233	0.0032
Ring	0.5049	0.0000
Satimage	0.7262	0.0005
Segment	0.7923	0.0013
Sonar	0.6861	0.0204
Spambase	0.8241	0.0008
Spectfheart	0.4097	0.0054
Texture	0.7414	0.0009
Thyroid	0.3158	0.0015
Titanic	0.7760	0.0000
Twonorm	0.9770	0.0003
Vehicle	0.4375	0.0086
Vowel	0.2748	0.0060
Wdbc	0.9404	0.0010
Wine	0.9416	0.0039
Winequality-red	0.5131	0.0035
Wisconsin	0.9458	0.0000
Avg	0.6841	0.0055

According to the triangle inequality, we can get

$$\begin{aligned} d^+ &= |t^* - C_{D^+}| \\ &= |t^* - \frac{1}{m}\sum_{i=1}^{m} t_i^+| \\ &= \frac{1}{m}|mt^* - \sum_{i=1}^{m} t_i^+| \\ &\leq \frac{1}{m}\sum_{i=1}^{m} |t^* - t_i^+| \\ &= \bar{d}_X \end{aligned}$$

in which the equality holds if and only if $t^* \geq \max\limits_{1\leq i\leq m} t_i^+$ or $t^* \leq \min\limits_{1\leq i\leq m} t_i^+$. Similarly, we can get $d^- \leq \bar{d}_Y$ in which the equality holds if and only if $t^* \geq \max\limits_{1\leq i\leq m} t_i^-$ or $t^* \leq \min\limits_{1\leq i\leq m} t_i^-$.

When $d^+ = \bar{d}_X$ and $d^- = \bar{d}_Y$, our method will assign the test instance to the same class label as the nearest centroid classification method. Obviously, the above analysis establish the equivalence between our method and the nearest centroid classifier under very strict constraints: (1) one-dimensional data set, (2) the test instance is no less (more) than all training instances in each class.

TABLE 7. The detailed experimental results of IBT-U-K-D.

Data sets	k=3		k=5		k=7		k=9	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Appendicitis	0.8283	0.0116	0.7764	0.0141	0.7642	0.0252	0.7170	0.0209
Balance	0.7782	0.0030	0.7528	0.0039	0.7184	0.0065	0.6834	0.0078
Banana	0.8642	0.0016	0.8500	0.0020	0.8338	0.0024	0.8238	0.0013
Bands	0.6978	0.0132	0.6726	0.0147	0.6564	0.0121	0.6452	0.0226
Bupa	0.5986	0.0087	0.5948	0.0116	0.5797	0.0196	0.5713	0.0119
Cleveland	0.5380	0.0074	0.5091	0.0191	0.4707	0.0122	0.4609	0.0150
Dermatology	0.9402	0.0046	0.9349	0.0076	0.9179	0.0106	0.9101	0.0072
Haberman	0.6585	0.0118	0.6585	0.0180	0.6261	0.0135	0.5971	0.0096
Hayes-roth	0.7500	0.0189	0.7256	0.0163	0.7038	0.0232	0.6969	0.0221
Heart	0.7552	0.0088	0.6856	0.0126	0.6722	0.0142	0.6652	0.0160
Hepatitis	0.8150	0.0115	0.7850	0.0287	0.7425	0.0251	0.7363	0.0161
Ionosphere	0.8556	0.0052	0.8575	0.0059	0.8558	0.0043	0.8541	0.0078
Iris	0.9600	0.0054	0.9420	0.0077	0.9053	0.0129	0.9127	0.0097
Led7digit	0.5770	0.0091	0.5230	0.0162	0.4604	0.0089	0.4286	0.0072
Marketing	0.2573	0.0016	0.2567	0.0025	0.2553	0.0024	0.2480	0.0027
Monks-2	0.7704	0.0124	0.7683	0.0174	0.7745	0.0182	0.7745	0.0170
Movement_libras	0.8181	0.0086	0.8036	0.0113	0.7978	0.0084	0.7875	0.0155
Newthyroid	0.9614	0.0058	0.9581	0.0062	0.9470	0.0070	0.9474	0.0054
Page-blocks	0.9534	0.0013	0.9466	0.0015	0.9405	0.0013	0.9361	0.0016
Penbased	0.9931	0.0002	0.9915	0.0005	0.9896	0.0004	0.9876	0.0005
Phoneme	0.8900	0.0014	0.8675	0.0022	0.8516	0.0020	0.8415	0.0033
Pima	0.6915	0.0089	0.6634	0.0096	0.6406	0.0135	0.6319	0.0134
Ring	0.7894	0.0013	0.7948	0.0016	0.8003	0.0020	0.8041	0.0018
Satimage	0.8949	0.0012	0.8827	0.0027	0.8706	0.0022	0.8634	0.0022
Segment	0.9640	0.0017	0.9572	0.0017	0.9513	0.0017	0.9396	0.0027
Sonar	0.8630	0.0089	0.8452	0.0115	0.8260	0.0084	0.7957	0.0109
Spambase	0.8978	0.0017	0.8704	0.0021	0.8458	0.0026	0.8306	0.0017
Spectfheart	0.6835	0.0149	0.6408	0.0129	0.6431	0.0188	0.6015	0.0131
Texture	0.9889	0.0005	0.9845	0.0010	0.9814	0.0009	0.9766	0.0008
Thyroid	0.9038	0.0012	0.8834	0.0020	0.8663	0.0016	0.8457	0.0019
Titanic	0.7897	0.0009	0.7899	0.0013	0.7717	0.0049	0.7564	0.0010
Twonorm	0.9381	0.0014	0.9194	0.0019	0.9006	0.0018	0.8880	0.0018
Vehicle	0.6833	0.0060	0.6619	0.0102	0.6426	0.0078	0.6344	0.0093
Vowel	0.9862	0.0027	0.9767	0.0024	0.9743	0.0023	0.9618	0.0028
Wdbc	0.9499	0.0037	0.9387	0.0062	0.9250	0.0060	0.9178	0.0057
Wine	0.9506	0.0069	0.9365	0.0046	0.9298	0.0130	0.9022	0.0113
Winequality-red	0.6196	0.0052	0.5790	0.0080	0.5444	0.0032	0.5225	0.0061
Wisconsin	0.9492	0.0022	0.9384	0.0031	0.9388	0.0041	0.9290	0.0053
Avg	0.8106	0.0058	0.7927	0.0080	0.7767	0.0086	0.7638	0.0082

For the multivariate case, it is very difficult to analyze their relationship in a quantitative manner. One naive connection is that if $(d_X - d_Y)(d^+ - d^-) > 0$, then our method and the nearest centroid classification method will produce the same classification result.

B. CONNECTION TO K-NN CLASSIFIER

The k-NN classifier is one of the most popular classification methods in the literature [29]. In our formulation, if the precedence test [10] is employed as the significance testing method, then we may uncover some interesting connections between our method and the k-NN classifier.

We still consider the binary classification problem in which the training data is composed of m positive instances from D^+ and n negative instances from D^-. Given an unknown instance t^*, the k-NN classification method finds its k nearest neighbors (k-NNs) to conduct the classification. These k-NNs can be divided into two groups: k^+ positive instances from D^+ and k^- instances from D^-, where $k = k^+ + k^-$. If $k^+ > k^-$, then t^* will be classified as a positive instance. Otherwise, t^* is assigned to the negative class.

The precedence test is a two-sample test based on the order of early failures [30]. Given two independent samples, $G_X = \{x_1, x_2, \ldots, x_m\}$ and $G_Y = \{y_1, y_2, \ldots, y_n\}$, let $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(m)}$ and $y_{(1)} \leq y_{(2)} \leq \ldots \leq y_{(n)}$ denote their order statistics. The precedence test is based on the number of observations from one sample which exceed (precede) some threshold specified by the other sample. More precisely, the

TABLE 8. The detailed experimental results of IBT-U-K-S.

Data sets	k=3		k=5		k=7		k=9	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Appendicitis	0.7585	0.0119	0.7594	0.0156	0.7896	0.0214	0.8047	0.0100
Balance	0.7282	0.0047	0.7490	0.0070	0.7494	0.0069	0.7878	0.0083
Banana	0.8826	0.0011	0.8885	0.0014	0.8941	0.0015	0.8965	0.0010
Bands	0.6915	0.0097	0.6734	0.0129	0.6770	0.0100	0.6575	0.0129
Bupa	0.6232	0.0189	0.6188	0.0140	0.6101	0.0101	0.6168	0.0118
Cleveland	0.4879	0.0131	0.4845	0.0092	0.4916	0.0091	0.4889	0.0081
Dermatology	0.9567	0.0020	0.9536	0.0024	0.9489	0.0042	0.9464	0.0018
Haberman	0.6010	0.0104	0.6173	0.0117	0.6281	0.0111	0.6212	0.0147
Hayes-roth	0.7325	0.0218	0.6038	0.0341	0.4988	0.0206	0.4850	0.0236
Heart	0.7833	0.0066	0.7985	0.0063	0.8037	0.0086	0.8026	0.0065
Hepatitis	0.7950	0.0087	0.8150	0.0053	0.8000	0.0118	0.8063	0.0106
Ionosphere	0.8698	0.0030	0.8695	0.0042	0.8678	0.0047	0.8667	0.0032
Iris	0.9587	0.0042	0.9600	0.0054	0.9593	0.0073	0.9587	0.0061
Led7digit	0.7088	0.0049	0.7242	0.0075	0.7336	0.0075	0.7324	0.0065
Marketing	0.2922	0.0027	0.2996	0.0028	0.3052	0.0018	0.3084	0.0027
Monks-2	0.7752	0.0084	0.7426	0.0109	0.7384	0.0096	0.7153	0.0095
Movement_libras	0.7839	0.0083	0.7106	0.0095	0.6264	0.0079	0.5964	0.0113
Newthyroid	0.9577	0.0056	0.9507	0.0050	0.9577	0.0056	0.9535	0.0066
Page-blocks	0.8574	0.0012	0.8441	0.0013	0.8377	0.0016	0.8427	0.0010
Penbased	0.9934	0.0002	0.9919	0.0003	0.9902	0.0002	0.9889	0.0003
Phoneme	0.8736	0.0018	0.8656	0.0010	0.8568	0.0016	0.8506	0.0017
Pima	0.7250	0.0045	0.7354	0.0069	0.7316	0.0052	0.7311	0.0067
Ring	0.7155	0.0021	0.6885	0.0012	0.6687	0.0013	0.6539	0.0016
Satimage	0.9024	0.0016	0.9021	0.0013	0.8988	0.0013	0.8959	0.0009
Segment	0.9601	0.0014	0.9515	0.0015	0.9506	0.0018	0.9487	0.0016
Sonar	0.8375	0.0101	0.8341	0.0129	0.7947	0.0072	0.7683	0.0136
Spambase	0.9047	0.0018	0.9031	0.0011	0.9048	0.0013	0.9026	0.0017
Spectfheart	0.6296	0.0101	0.5906	0.0092	0.5918	0.0092	0.5809	0.0076
Texture	0.9868	0.0004	0.9835	0.0007	0.9811	0.0005	0.9785	0.0007
Thyroid	0.7707	0.0016	0.7826	0.0024	0.7568	0.0016	0.7504	0.0021
Titanic	0.7601	0.0000	0.7607	0.0013	0.7883	0.0008	0.7892	0.0001
Twonorm	0.9667	0.0008	0.9710	0.0005	0.9726	0.0005	0.9732	0.0007
Vehicle	0.7116	0.0067	0.7047	0.0119	0.6974	0.0067	0.6918	0.0070
Vowel	0.9606	0.0048	0.8629	0.0095	0.7551	0.0113	0.6969	0.0093
Wdbc	0.9645	0.0026	0.9664	0.0021	0.9680	0.0031	0.9685	0.0029
Wine	0.9534	0.0075	0.9528	0.0054	0.9517	0.0060	0.9573	0.0039
Winequality-red	0.4826	0.0070	0.4994	0.0057	0.4946	0.0066	0.5063	0.0078
Wisconsin	0.9750	0.0034	0.9755	0.0029	0.9739	0.0013	0.9735	0.0016
Avg	0.7978	0.0057	0.7891	0.0064	0.7801	0.0060	0.7762	0.0060

test statistic W_r is the number of observations in G_X that precede the r-th order statistic $y_{(r)}$ from G_Y. Alternatively, one can use the number of observations in G_Y that exceed the s-th order statistic $x_{(s)}$ from G_x as the test statistic W_s. Large values of these two test statistics will lead to the rejection of the null hypothesis that two distributions are equal.

In our problem formulation, G_X (G_Y) is the distance set between t^* and the instances in D^+ (D^-). Then, $x_{(1)}, x_{(2)}, \ldots, x_{(k^+)}, y_{(1)}, y_{(2)}, \ldots, y_{(k^-)}$ will be the k distance values between t^* and its k-NNs. If we use the precedence test as the significance testing method and suppose that $x_{(k^+)} \leq y_{(k^-+1)} \leq x_{(k^++1)}$, we can set $r = k^- + 1$ to obtain the corresponding test statistic $W_r = k^+$ for testing the null hypothesis against the alternative hypothesis ($F_X < F_Y$). Alternatively, if we let $s = k^+ + 1$, we can obtain another test statistic $W_s = k^-$ for testing the null hypothesis against the alternative hypothesis ($F_X > F_Y$). And we can also get two p-values, p_X and p_Y. At last, t^* will be assigned to the positive (negative) class if the former (latter) is smaller.

If we further assume that the positive training set and the negative training set have the same size, i.e., $m = n$, then the two p-values will be totally determined by the two test statistics: $p_X < p_Y \Leftrightarrow k^+ > k^-$ or $p_X > p_Y \Leftrightarrow k^+ < k^-$. Therefore, our method and the k-NN classifier will generate the same classification result under the above assumptions. From this aspect, we may regard our method equipped with the precedence test as a generalized ''statistical'' k-NN classifier.

VI. CONCLUSION

Due to the importance of the classification problem, many effective classification algorithms have been proposed from different societies. However, most work on classification does not address the issue of statistical significance. Towards this direction, several initial research efforts have investigated the feasibility of constructing a classifier through significance testing. Unfortunately, this interesting idea has not received much attention during the past 10 years. This is mainly because of the following reasons: (1) there are still no such testing-based classifiers that can achieve the same level performance as the state-of-the-art methods on real data sets; (2) the potential benefit of deploying such testing-based classifiers is still not clear.

Based on the above observations, this paper takes one step further towards this direction by formulating the classification problem as a two-sample testing problem. This new formulation enables us to generate several testing-based classifiers that have comparable performance with standard classifiers such as SVM. In addition, we show that it is quite easy to handle outlying test instances and control the FDR of classification results based on the p-values associated with each test instance.

We believe this paper will significantly contribute to the development of testing-based classification model, which

TABLE 9. The detailed experimental results of TBC and IDC.

Data sets	TBC		IDC	
	Avg	Std	Avg	Std
Appendicitis	0.8613	0.0064	0.8075	0.0101
Balance	0.8654	0.0050	0.7618	0.0065
Banana	0.5568	0.0013	0.7313	0.0019
Bands	0.6088	0.0115	0.5841	0.0141
Bupa	0.6275	0.0088	0.5803	0.0086
Cleveland	N/A	N/A	0.4892	0.0126
Dermatology	N/A	N/A	0.8746	0.0066
Haberman	0.7310	0.0064	0.6876	0.0222
Hayes-roth	0.5288	0.0053	0.4744	0.0238
Heart	0.8396	0.0072	0.8170	0.0040
Hepatitis	N/A	N/A	0.8475	0.0211
Ionosphere	0.8695	0.0057	0.7513	0.0043
Iris	0.6667	0.0000	0.9060	0.0021
Led7digit	0.2622	0.0109	0.4736	0.0080
Marketing	0.8088	0.0016	0.1284	0.0018
Monks-2	0.2652	0.0029	0.6391	0.0140
Movement_libras	N/A	N/A	0.2642	0.0169
Newthyroid	0.3023	0.0000	0.8377	0.0056
Page-blocks	0.0750	0.0006	0.8892	0.0007
Penbased	0.1998	0.0000	0.6636	0.0004
Phoneme	0.7595	0.0005	0.7684	0.0010
Pima	0.7615	0.0041	0.7177	0.0028
Ring	0.7621	0.0008	0.9603	0.0004
Satimage	0.3448	0.0002	0.6317	0.0010
Segment	0.2857	0.0000	0.6624	0.0022
Sonar	0.7447	0.0159	0.7226	0.0116
Spambase	0.9064	0.0011	0.8376	0.0006
Spectfheart	0.6105	0.0122	0.7528	0.0138
Texture	0.3163	0.0013	0.5477	0.0022
Thyroid	0.0713	0.0001	0.9239	0.0005
Titanic	0.7807	0.0008	0.6900	0.0015
Twonorm	0.9781	0.0003	0.9771	0.0001
Vehicle	0.5324	0.0196	0.3018	0.0041
Vowel	0.1818	0.0000	0.2899	0.0060
Wdbc	0.9617	0.0023	0.9374	0.0017
Wine	0.6011	0.0000	0.9472	0.0060
Winequality-red	N/A	N/A	0.4021	0.0026
Wisconsin	0.9581	0.0012	0.9555	0.0016
Avg	0.5947	0.0041	0.6859	0.0066

will become a new promising classifier family. As the study on the testing-based classification model is still in its infancy stage, many research issues remain unexplored and should be further investigated in the future work. For example, since all the existing testing-based classifiers are based on the idea of instance-based learning, how to build a non-lazy testing-based classifier will be an interesting and challenging issue.

APPENDIX. DETAILED EXPERIMENTAL RESULTS

A. THE DETAILED EXPERIMENTAL RESULTS OF IBT-U

The detailed experimental results of IBT-U are given in Table 6.

B. THE DETAILED EXPERIMENTAL RESULTS OF IBT-U-K-D

The detailed experimental results of IBT-U-K-D are given in Table 7.

C. THE DETAILED EXPERIMENTAL RESULTS OF IBT-U-K-S

The detailed experimental results of IBT-U-K-S are given in Table 8.

D. THE DETAILED EXPERIMENTAL RESULTS OF TBC AND IDC

The detailed experimental results of TBC and IDC are given in Table 9.

E. THE DETAILED EXPERIMENTAL RESULTS OF k-NN

The detailed experimental results of k-NN are given in Table 10.

TABLE 10. The detailed experimental results of k-NN.

Data sets	k=3		k=5		k=7		k=9	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Appendicitis	0.8406	0.0094	0.8642	0.0119	0.8764	0.0030	0.8708	0.0100
Balance	0.8485	0.0065	0.8661	0.0059	0.8813	0.0048	0.8928	0.0048
Banana	0.8841	0.0014	0.8896	0.0012	0.8942	0.0021	0.8978	0.0012
Bands	0.7093	0.0122	0.6942	0.0122	0.6797	0.0083	0.6712	0.0098
Bupa	0.6371	0.0113	0.6078	0.0130	0.6238	0.0121	0.6293	0.0134
Cleveland	0.5545	0.0152	0.5545	0.0057	0.5663	0.0115	0.5626	0.0117
Dermatology	0.9623	0.0033	0.9592	0.0027	0.9575	0.0039	0.9517	0.0040
Haberman	0.6954	0.0109	0.6944	0.0082	0.7111	0.0054	0.7186	0.0070
Hayes-roth	0.6350	0.0187	0.5575	0.0255	0.4344	0.0215	0.3581	0.0228
Heart	0.7778	0.0089	0.8033	0.0066	0.8126	0.0068	0.8115	0.0069
Hepatitis	0.8288	0.0145	0.8525	0.0255	0.8800	0.0134	0.8563	0.0169
Ionosphere	0.8570	0.0044	0.8501	0.0054	0.8393	0.0041	0.8425	0.0043
Iris	0.9507	0.0034	0.9560	0.0034	0.9673	0.0066	0.9527	0.0049
Led7digit	0.6598	0.0077	0.7116	0.0047	0.7090	0.0058	0.7234	0.0041
Marketing	0.2872	0.0030	0.2942	0.0015	0.2990	0.0025	0.3050	0.0020
Monks-2	0.7972	0.0072	0.8000	0.0054	0.7914	0.0127	0.7644	0.0074
Movement_libras	0.8075	0.0049	0.7417	0.0103	0.7181	0.0090	0.6739	0.0218
Newthyroid	0.9409	0.0044	0.9381	0.0058	0.9316	0.0054	0.9237	0.0050
Page-blocks	0.9596	0.0012	0.9583	0.0009	0.9545	0.0009	0.9536	0.0006
Penbased	0.9935	0.0004	0.9926	0.0004	0.9919	0.0003	0.9905	0.0003
Phoneme	0.8878	0.0021	0.8808	0.0028	0.8752	0.0017	0.8701	0.0023
Pima	0.7396	0.0055	0.7367	0.0072	0.7449	0.0055	0.7357	0.0046
Ring	0.7186	0.0014	0.6922	0.0010	0.6747	0.0012	0.6608	0.0017
Satimage	0.9096	0.0012	0.9078	0.0011	0.9065	0.0015	0.9049	0.0019
Segment	0.9613	0.0020	0.9532	0.0014	0.9502	0.0015	0.9481	0.0015
Sonar	0.8303	0.0072	0.8135	0.0115	0.7880	0.0135	0.7457	0.0175
Spambase	0.9019	0.0021	0.9030	0.0015	0.8995	0.0013	0.8959	0.0023
Spectfheart	0.7150	0.0134	0.7390	0.0149	0.7629	0.0142	0.7547	0.0124
Texture	0.9878	0.0005	0.9853	0.0005	0.9828	0.0007	0.9809	0.0007
Thyroid	0.9391	0.0008	0.9407	0.0005	0.9401	0.0005	0.9400	0.0002
Titanic	0.6109	0.0107	0.7796	0.0118	0.7819	0.0013	0.7816	0.0034
Twonorm	0.9650	0.0010	0.9697	0.0007	0.9705	0.0008	0.9714	0.0006
Vehicle	0.7033	0.0051	0.7025	0.0054	0.7039	0.0055	0.6941	0.0096
Vowel	0.9706	0.0025	0.9387	0.0057	0.8871	0.0071	0.7972	0.0108
Wdbc	0.9692	0.0017	0.9678	0.0024	0.9705	0.0027	0.9692	0.0028
Wine	0.9640	0.0039	0.9573	0.0089	0.9596	0.0052	0.9567	0.0088
Winequality-red	0.5839	0.0062	0.5902	0.0069	0.5797	0.0040	0.5803	0.0042
Wisconsin	0.9691	0.0022	0.9742	0.0024	0.9728	0.0019	0.9706	0.0021
Avg	0.8146	0.0058	0.8163	0.0064	0.8124	0.0055	0.8028	0.0065

F. THE DETAILED EXPERIMENTAL RESULTS OF SVM, DECISION TREE AND NEAREST CENTROID CLASSIFIER

The detailed experimental results of SVM, decision tree (DT) and nearest centroid classifier (NC) are given in Table 11.

TABLE 11. The detailed experimental results of SVM, DT and NC.

Data sets	SVM		DT		NC	
	Avg	Std	Avg	Std	Avg	Std
Appendicitis	0.8702	0.0051	0.8358	0.0135	0.8315	0.0067
Balance	0.8787	0.0038	0.7894	0.0080	0.7403	0.0068
Banana	0.5517	0.0000	0.8799	0.0027	0.5611	0.0008
Bands	0.6893	0.0054	0.6285	0.0272	0.6267	0.0082
Bupa	0.5801	0.0032	0.6571	0.0183	0.5769	0.0098
Cleveland	0.5823	0.0089	0.5091	0.0079	0.5312	0.0110
Dermatology	0.9809	0.0035	0.9374	0.0058	0.9634	0.0034
Haberman	0.7334	0.0032	0.6935	0.0139	0.6590	0.0313
Hayes-roth	0.5341	0.0070	0.8181	0.0192	0.5368	0.0225
Heart	0.8400	0.0065	0.7581	0.0196	0.8085	0.0058
Hepatitis	0.8530	0.0212	0.8350	0.0269	0.8235	0.0150
Ionosphere	0.8769	0.0076	0.8806	0.0101	0.7426	0.0045
Iris	0.8547	0.0076	0.9487	0.0045	0.9253	0.0082
Led7digit	0.7241	0.0075	0.7114	0.0075	0.7365	0.0055
Marketing	0.2412	0.0073	0.2970	0.0032	0.2896	0.0015
Monks-2	0.8056	0.0004	0.9067	0.0130	0.8051	0.0022
Movement_libras	0.6799	0.0101	0.6572	0.0265	0.5605	0.0076
Newthyroid	0.8406	0.0042	0.9298	0.0060	0.9344	0.0036
Page-blocks	0.9235	0.0002	0.9649	0.0010	0.7669	0.0013
Penbased	0.9284	0.0004	0.9582	0.0010	0.8133	0.0006
Phoneme	0.7734	0.0006	0.8650	0.0032	0.7394	0.0007
Pima	0.7699	0.0033	0.7078	0.0105	0.7314	0.0023
Ring	0.7654	0.0012	0.8858	0.0028	0.7613	0.0010
Satimage	0.8201	0.0008	0.8608	0.0039	0.7805	0.0007
Segment	0.8986	0.0011	0.9568	0.0039	0.8405	0.0011
Sonar	0.7781	0.0157	0.7221	0.0185	0.6891	0.0095
Spambase	0.9031	0.0006	0.9190	0.0028	0.8359	0.0006
Spectfheart	0.7969	0.0018	0.7401	0.0155	0.6717	0.0045
Texture	0.9794	0.0003	0.9220	0.0030	0.7643	0.0006
Thyroid	0.9372	0.0001	0.9960	0.0004	0.4434	0.0007
Titanic	0.7760	0.0000	0.7898	0.0013	0.7506	0.0000
Twonorm	0.9783	0.0004	0.8431	0.0048	0.9769	0.0002
Vehicle	0.6857	0.0054	0.7139	0.0115	0.4428	0.0041
Vowel	0.4513	0.0081	0.7666	0.0111	0.3842	0.0253
Wdbc	0.9776	0.0017	0.9185	0.0040	0.9385	0.0022
Wine	0.9785	0.0036	0.9096	0.0107	0.9651	0.0042
Winequality-red	0.5295	0.0020	0.6077	0.0102	0.3393	0.0032
Wisconsin	0.9705	0.0014	0.9492	0.0042	0.9649	0.0001
Avg	0.7826	0.0042	0.8071	0.0094	0.7172	0.0057

REFERENCES

[1] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3133–3181, 2014.

[2] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[3] C. Cortes and V. N. Vapnik, "Support vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1997.

[4] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[5] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: A practical and powerful approach to multiple testing," *J. Roy. Stat. Soc. B, Methodol.*, vol. 57, no. 1, pp. 289–300, Jan. 1995.

[6] O. Wagih, J. Reimand, and G. D. Bader, "MIMP: Predicting the impact of mutations on kinase-substrate phosphorylation," *Nature Methods*, vol. 12, no. 6, pp. 531–533, Jun. 2015.

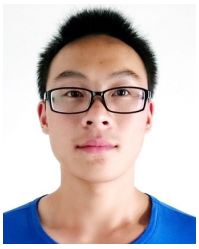
[7] S.-M. Liao and M. Akritas, "Test-based classification: A linkage between classification and statistical testing," *Statist. Probab. Lett.*, vol. 77, no. 12, pp. 1269–1281, Jul. 2007.

[8] S. Ghimire and H. Wang, "Classification of image pixels based on minimum distance and hypothesis testing," *Comput. Statist. Data Anal.*, vol. 56, no. 7, pp. 2273–2287, Jul. 2012.

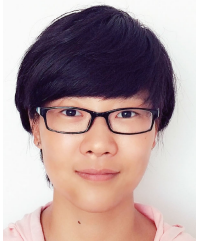
[9] L. Guo and R. Modarres, "Interpoint distance classification of high dimensional discrete observations," *Int. Stat. Rev.*, vol. 87, no. 2, pp. 191–206, Aug. 2019.

[10] J. D. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference*, 5th ed. Boca Raton, FL, USA: CRC Press, 2011.
[11] D. Dheeru and E. K. Taniskidou. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml
[12] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, and S. García, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Multiple-Valued Logic Soft Comput.*, vol. 17, nos. 2–3, pp. 255–287, 2011.
[13] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, 1997.
[14] D. R. Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithms," *Mach. Learn.*, vol. 38, no. 3, pp. 257–286, 2000.
[15] S. Garcia, J. Derrac, J. R. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: Taxonomy and empirical study," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 417–435, Mar. 2012.
[16] J. Derrac, S. García, and F. Herrera, "Fuzzy nearest neighbor algorithms: Taxonomy, experimental analysis and prospects," *Inf. Sci.*, vol. 260, pp. 98–119, Mar. 2014.
[17] R. Modarres, "On the interpoint distances of Bernoulli vectors," *Statist. Probab. Lett.*, vol. 84, pp. 215–222, Jan. 2014.
[18] R. Modarres, "Multivariate Poisson interpoint distances," *Statist. Probab. Lett.*, vol. 112, pp. 113–123, May 2016.
[19] R. Modarres, "Multinomial interpoint distances," *Stat. Papers*, vol. 59, no. 1, pp. 341–360, Mar. 2018.
[20] C. Elkan, "The foundations of cost-sensitive learning," in *Proc. 17th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2001, pp. 973–978.
[21] B. Zadrozny, J. Langford, and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting," in *Proc. 3rd IEEE Int. Conf. Data Mining*, Nov. 2003, pp. 435–442.
[22] C. Scott and R. Nowak, "A Neyman–Pearson approach to statistical learning," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3806–3819, Nov. 2005.
[23] X. Tong, Y. Feng, and A. Zhao, "A survey on Neyman-Pearson classification and suggestions for future research," *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 8, no. 2, pp. 64–81, Mar. 2016.
[24] X. Tong, Y. Feng, and J. J. Li, "Neyman-Pearson classification algorithms and NP receiver operating characteristics," *Sci. Adv.*, vol. 4, no. 2, Feb. 2018, Art. no. eaao1659.
[25] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *Ann. Math. Statist.*, vol. 18, no. 1, pp. 50–60, Mar. 1947.
[26] G. Marsaglia, W. Tsang, and J. Wang, "Evaluating Kolmogorov's distribution," *J. Statist. Softw.*, vol. 8, no. 18, pp. 1–4, 2003.
[27] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen, *Classification Regression Trees*. Boca Raton, FL, USA: CRC Press, 1984.
[28] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*. New York, NY, USA: Springer, 2001.
[29] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.
[30] N. Balakrishnan and H. T. Ng, *Precedence-Type Tests and Applications*. Hoboken, NJ, USA: Wiley, 2006.

ZENGYOU HE received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, China, in 2000, 2002, and 2006, respectively, all in computer science. From February 2007 to February 2010, he was a Research Associate with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology. He is currently a Professor with the School of Software, Dalian University of Technology. His research interests include data mining and bioinformatics.



CHAOHUA SHENG received the B.S. degree in software engineering from the Dalian University of Technology, China, in 2018, where he is currently pursuing the M.S. degree with the School of Software. His research interests include data mining and its applications.



YAN LIU received the M.S. degree in applied statistics from the Dongbei University of Finance and Economics, in 2018. She is currently pursuing the Ph.D. degree with the School of Software, Dalian University of Technology. Her research interests include data mining and its applications.

QUAN ZOU (Senior Member, IEEE) received the Ph.D. degree from the Harbin Institute of Technology, China, in 2009. From 2009 to 2015, he was an Assistant and an Associate Professor with Xiamen University, China. He is currently a Professor of computer science with Tianjin University. Several related works have been published in *Briefings in Bioinformatics*, *Bioinformatics*, IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, and so on. Google Scholar showed that his more than 100 articles have been cited more than 1,300 times. His research interests include bioinformatics, machine learning, and parallel computing. In February 2005, he received the Outstanding Reviewers for Computers in Biology and Medicine Award. He is also a reviewer for many impacted journals, including *Bioinformatics*, *Briefings in Bioinformatics*, IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, and *BMC Bioinformatics*.