A Hybrid Classification Framework Based on Clustering

Jin Xiao, *Member, IEEE*, Yuhang Tian, *Student Member, IEEE*, Ling Xie, Xiaoyi Jiang, *Senior Member, IEEE*, and Jing Huang

***Abstract*—The traditional supervised classification algorithms tend to focus on uncovering the relationship between sample attributes and the class labels; they seldom consider the potential structural characteristics of the sample space, often leading to unsatisfactory classification results. To improve the performance of classification models, many scholars have sought to construct hybrid models by combining both supervised and unsupervised learning. Although the existing hybrid models have shown significant potential in industrial applications, our experiments indicate that some shortcomings remain. With the aim of overcoming such shortcomings of the existing hybrid models, this article proposes a hybrid classification framework based on clustering (HCFC). First, it applies a clustering algorithm to partition the training samples into *K* clusters. It then constructs a clustering-based attribute selection measure—namely, the hybrid information gain ratio, based upon which it then trains a C4.5 decision tree. Depending on the differences in the clustering algorithms used, this article constructs two different versions of the HCFC (HCFC-K and HCFC-D) and tests them on eight benchmark datasets in the healthcare and disease diagnosis industries and on 15 datasets from other fields. The results indicate that both versions of the HCFC achieve a comparable or even better classification performance than the other three hybrid and six single models considered. In addition, the HCFC-D has a stronger ability to resist class noise compared with the HCFC-K.**

Manuscript received November 26, 2018; revised April 6, 2019 and July 8, 2019; accepted July 29, 2019. Date of publication August 7, 2019; date of current version January 17, 2020. This work was supported in part by the Major Project of the National Social Science Foundation of China under Grant 18VZL006; in part by the EU Horizon 2020 RISE Project ULTRACEPT under Grant 778062; in part by the National Natural Science Foundation of China under Grant 71471124; in part by the Tianfu Ten-Thousand Talents Program of Sichuan Province; in part by the Excellent Youth Fund of Sichuan University under Grant skqx201607, Grant sksyl201709, and Grant skzx2016-rcrw14; and in part by the Leading Cultivation Talents Program of Sichuan University. Paper no. TII-18-3079. *(Corresponding authors: Jin Xiao; Jing Huang.)*

J. Xiao and Y. Tian are with the Business School, Sichuan University, Chengdu 610064, China (e-mail: xiaojin@scu.edu.cn; tyh70537@outlook.com).

L. Xie is with the School of Medical Information Engineering, Zunyi Medical University, Zunyi 563006, China (e-mail: xie_ling0101@126.com).

X. Jiang is with the Department of Mathematics and Computer Science, University of Münster, D-48149 Münster, Germany (e-mail: xjiang@uni-muenster.de).

J. Huang is with the School of Public Administration, Sichuan University, Chengdu 610064, China (e-mail: totojh@scu.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2019.2933675

***Index Terms*—Classification, clustering, decision tree, hybrid model, hybrid information gain ratio, industrial application.**

I. Introduction

CLASSIFICATION is an active research problem in the areas of data mining and machine learning [1], and has a wide range of applications in industry, including fault diagnosis [2], image recognition [3], network anomaly detection [4], disease diagnosis [5], and power system security [6]. To date, most classification algorithms are based on supervised learning. In supervised learning, the class labels of the training samples are known. The goal of a supervised classification algorithm is to determine the relationship between the sample attributes and the class labels; from there, it can be used to construct a classification model that can accurately predict the class labels of new samples. However, few of the current supervised classification algorithms consider the potential structure of the sample space when conducting the modeling, which often leads to poor classification results. Unlike supervised learning, unsupervised learning can group similar samples together based on their distance or similarity: it does not rely on the class labels of the samples. Although unsupervised learning does not directly generate label predictions, it can reveal the underlying structure of the sample space and uncover the intrinsic relationship between the samples; in this way, it provides useful information for solving classification problems [7], [8]. An increasing number of researchers have, therefore, recently started combining supervised and unsupervised learning.

A. Literature Review

The existing supervised classification algorithms can be roughly divided into two categories, namely, symbolic and statistical learning algorithms. Symbolic learning algorithms are the most common, and include decision tree algorithms, such as ID3 [9] and C4.5 [10]; rule-based algorithms, such as CN2 [11] and first-order inductive learners (FOIL) [12]; and example-based algorithms, such as *k*-nearest neighbor (KNN) [13] and a parallel exemplar-based learning system [14]. Statistical learning algorithms, by contrast, mainly include the classification and regression tree [15], support vector machine (SVM) [16], naive Bayes (NB) [17], logistic regression (LR) [18], and neural network algorithms. In recent years, deep neural networks (DNNs) have achieved a significant level of success in various applications, particularly, in tasks involving visual

and speech information. Nevertheless, DNNs also have clear deficiencies. For example, DNNs usually require a significant amount of training data and powerful computational facilities, and a lot of time is required to tune their parameters [19].

In the last ten years, scholars have successively proposed a number of hybrid models that combine unsupervised and supervised learning, to improve the classification performance of models. The most commonly used method for constructing a hybrid model is to combine clustering with a decision tree algorithm. For example, Gaddam *et al.* first proposed a hybrid model called K-means+ID3 [20]. With this method, K-means clustering is applied to partition the training samples into K disjoint clusters; an ID3 decision tree is then trained on each cluster. Finally, to obtain a final classification decision for each test sample, the following two rules are used: 1) the nearest-neighbor rule and 2) the nearest-consensus rule. Experiments on three network anomaly detection datasets showed that the hybrid model achieves a better performance than a single ID3 model. Bose *et al.* proposed a two-stage hybrid model (KM-Boosted C5.0) [7] consisting of an unsupervised clustering technique and a boosted C5.0 decision tree to predict customer churn. In the first stage, a clustering algorithm is used to cluster the samples; from there, the clustering labels are added to the original dataset as a new attribute. During the second stage, the newly obtained dataset is used to train the boosted C5.0 decision tree model. Their experiment results indicate that the use of clustering information leads to an improved top-decile lift for the hybrid model, as compared with a benchmark case in which no clustering information is applied. Kaewchinporn *et al.* combined a decision tree with a clustering algorithm, and proposed a hybrid model called tree bagging and weighted clustering (TBWC) [21]. First, important attributes and their weights are selected by applying decision tree bagging; the weighted attributes are then used to generate clusters through which new objects are classified. The experiment results showed that, based on five experiment datasets applied, the TBWC model achieves a higher accuracy than the C4.5 decision tree.

In addition, there are some studies that use unsupervised and supervised learning to improve the classification performance of neural networks. Wang *et al.* proposed a hybrid intrusion detection approach called FC-ANN [22], which is based on a back propagation neural network (BPNN) and fuzzy c-means. The training set is first divided into several subsets by fuzzy c-means clustering. Then, based on different training subsets, different BPNN models are trained as different base models. Finally, a metalearner, namely, a fuzzy aggregation module, is employed to aggregate the results. Experiment results on the KDD CUP 1999 dataset showed that the proposed approach outperforms a BPNN and other well-known methods, such as decision tree and NB in terms of the detection precision and detection stability. Eslamloueyan proposed a duty-oriented hybrid model for isolating the faults of the Tennessee–Eastman process (TEP) [23]. First, it applies a fuzzy c-means clustering algorithm to partition the fault pattern space into a few subspaces. Then, for each subspace, a special multilayer perceptron (MLP) neural network is trained to diagnose the faults of that subspace. Finally, a supervisor MLP neural network is developed to determine which special MLP neural network should be triggered. Experiments on the simulation datasets showed that the proposed hybrid method is considerably better than a single MLP neural network. Ma *et al.* proposed a hybrid approach called spectral clustering and deep neural network (SCDNN) [24], which combines spectral clustering and a DNN. First, the training set is divided into K subsets and K cluster centers are obtained. Second, a DNN model is trained on each training subset. Finally, initialized using the K cluster centers, the test set is divided into K subsets, and each subset is then fed into the most appropriate DNN model to evaluate the performance of the hybrid approach. Experiments on six intrusion detection datasets showed that the SCDNN model performs better than a BPNN, SVM, random forest, and Bayes tree models in terms of the detection accuracy.

Finally, some other supervised learning techniques have also been used to construct hybrid models. For example, Huang *et al.* proposed a learning system for predicting customer behaviors, which combines a weighted K-means clustering algorithm and FOIL (WK+FOIL) [8]. First, to obtain satisfactory clustering results on high-dimensional datasets, a weighted K-means algorithm is used to cluster the samples. Then, to obtain a highly interpretive classification model, the system uses a rule-inductive technique on each cluster to obtain a set of rules that can accurately identify churn customers. The experiment results showed that the WK+FOIL model performed better than other hybrid models on the experiment datasets. Furthermore, Rajamohamed *et al.* consecutively combined rough K-means clustering with five supervised classification models, to construct different versions of hybrid models [25]. The final experiment results showed that, when the rough K-means algorithm was combined with a support vector machine (RK+SVM), the hybrid model achieved the best performance.

B. Our Motivation

The aforementioned propositions have made important contributions to the development of hybrid models. However, we found experimentally that the existing studies, and their outcomes, still have a number of shortcomings.

1) Most of the existing hybrid models first partition the training set into multiple clusters, and then, train a single classifier on each cluster. Finally, for each test sample, a classifier is selected according to certain classification criteria [8], [20], [25]. Intuitively, the advantage of this approach is that it breaks down a complex classification problem into many simpler problems such that each classifier is more focused on the classification of samples in a specific region. However, a potential problem caused by this type of approach is that many samples may be located near the boundaries of the clusters when given a training set that is not-well-separated [26], and such samples are often difficult to classify correctly using nearby classifiers because they are far from any cluster center.
2) Real-world datasets may contain a large amount of noise; therefore, it is important to compare the performance of

the classification models in noisy environments. To the best of our knowledge, the existing research on hybrid models does not consider the effects of noise.

C. Our Contributions

To compensate for the deficiencies inherent in the existing hybrid models, we propose a new hybrid classification framework based on clustering (HCFC). Unlike the existing hybrid models, which train classifiers separately on many clusters, the HCFC integrates the clustering information into the training of classifiers from the perspective of the information theory. The HCFC can be roughly divided into two steps: 1) clustering and 2) classification. In the first step, any clustering algorithm can be used to cluster the training set. In the second step, we construct a clustering-based attribute selection measure, namely, the hybrid information gain ratio, to train a single C4.5 decision tree on the entire training set. Because the training of the final classifier of the HCFC is conducted on the whole training set, it has a significant advantage in solving classification problems where the training sets are not-well-separated compared with the existing hybrid models. Moreover, because the hybrid information gain ratio proposed in this article considers both the class labels of the samples and the clusters to which they belong, the HCFC is expected to achieve a better performance for classification problems with a large amount of class noise.

In this article, we constructed two versions of the HCFC (HCFC-K and HCFC-D), using an improved version of K-means [27] and a density-based spatial clustering of applications with noise (DBSCAN) [28], respectively. We used these two clustering algorithms because they are widely applied in industry and have their own unique advantages. In addition, we choose C4.5 because it remains one of the most popular classification models in the industry, and not only does it achieve a good classification performance, it also has strong interpretability [29]. The final experiment results on the 23 UCI (University of California at Irvine) datasets show that the proposed models achieve a better classification performance than three hybrid and six single classification models considered.

The novelty of this article can be summarized as follows.

1) We propose a new hybrid classification framework that does not restrict the selection of clustering algorithms and is extremely flexible in industrial applications.
2) We propose a clustering-based attribute selection method called the hybrid information gain ratio that effectively integrates the clustering information into the training of the decision tree and improves the classification performance.
3) The proposed framework takes into account the influence of noise while effectively utilizing the clustering information, and effectively avoids the shortcomings of the traditional hybrid models.

D. Organization of this Article

This article is structured as follows. Section II introduces related theories. Section III details the HCFC, including the basic concept, a calculation of the hybrid information gain ratio, a complexity analysis, and the modeling steps. Section IV presents the experiment results and the corresponding analysis. Section V concludes this article.

Algorithm 1: Improved K-means Algorithm(H).

Input: Dataset H
Output: Clustering result C

$CM = \emptyset$
for $j = 1$ to J **do**
  let H_j be a small random subset of H
  H_j is clustered via the classical K-means algorithm producing cluster centers CM_j
  $CM = CM \cup CM_j$
end for
for $j = 1$ to J **do**
  let CM_j be the initial cluster centers
  CM is clustered via the classical K-means algorithm producing a solution FM_j
end for
let $FM_{\text{refined}} = \arg\min_{FM_j} \{\text{distortion}(FM_j, CM)\}$
FM_{refined} is used as the refined initial cluster centers in the classical K-means clustering
return clustering result C

II. Related Theories

A. Improved K-means Algorithm

K-means is one of the most widely used clustering algorithms. However, given the gradient descent nature of the K-means algorithm, it is highly sensitive to the initial placement of the cluster centers. To address this issue, numerous initialization methods for the cluster centers have been proposed. Celebi *et al.* compared eight different initialization methods in detail using five effectiveness and two efficiency criteria [30]. Their experiment results show that, in terms of most of the criteria considered, the method proposed by Bradley and Fayyad [27] is superior to other methods. We, therefore, used this method to derive the refined initial cluster centers for the K-means algorithm.

The general steps of Bradley and Fayyad's method are as follows. First, J subsets are randomly selected from the dataset H, which is denoted as $H_j (j = 1, 2, ..., J)$. Second, the classical K-means algorithm is used to cluster the J subsets consecutively, and we let $CM_j (j = 1, 2, ..., J)$ denote the K cluster centers obtained from the subset H_j. Next, we set $CM = \{CM_1, CM_2, ..., CM_j, ..., CM_J\}$, which contains $J \times K$ cluster centers. Third, CM is clustered J times using the classical K-means algorithm, each time initialized using the cluster centers CM_j; here, FM_j denotes the final cluster centers obtained by the jth clustering. Finally, the refined initial cluster centers FM_{refined} are chosen as FM_j, which achieves minimal distortion over the set CM. Algorithm 1 briefly summarizes the method, and further details can be found in [27].

B. DBSCAN

DBSCAN is a classical density-based clustering algorithm that has been applied in many fields of science [28]. Compared with the K-means algorithm, DBSCAN can find clusters with an arbitrary shape. In DBSCAN, the density of a sample can be measured by counting the number of samples within a specified radius (ϵ) around the sample. The samples with a density above a specified threshold (*MinPts*) are constructed as clusters. Refer to [28] for a more detailed description of this algorithm.

C. C4.5 Decision Tree

C4.5 is a well-known algorithm for generating decision trees. As an extension of ID3, this algorithm can handle continuous attributes and uses the information gain ratio as a criterion by which to select splitting attributes. Compared with the information gain used in ID3, the use of the information gain ratio can preclude bias toward multivalued attributes. In addition, C4.5 uses a pruning method called "error-based pruning" [10], which has the advantage of not requiring an additional validation set.

The general steps required for building a decision tree are as follows. Given a training set *D* that includes *M* different classes, $L = \{l_m \mid m = 1, 2, ..., M\}$, its attribute set is *S*. For any attribute $A \in S$, the information gain ratio $GainRatio(D, A, L)$ is calculated and the best splitting attribute A_{best} with the highest information gain ratio is chosen. Once the splitting attribute is determined, the training set *D* is divided into multiple subsets according to the values of the splitting attribute. In each subset, the algorithm terminates if any one of the following three situations occurs:

1) all training samples are of the same class;
2) the current attribute set is empty;
3) no training samples exist.

Otherwise, the partitioning process will be conducted recursively. Algorithm 2 shows the pseudocode of the C4.5 decision tree, and further details can be found in [10].

Algorithm 2: C4.5(D, S, L).

Input: Training set D; attributes set S; class set L
Output: Decision tree *Tree*

Tree $= \{\}$
if all samples in *D* are of the same class $l_m \in L$ **then**
  return a single node with the class l_m
end if
if S is empty **then**
  return a single node with most frequent class in *D*
end if
for $A \in S$ **do**
  compute GainRatio(D, A, L)
end for
$A_{\text{best}} = \arg\max_A(\text{GainRatio}(D, A, L))$
Tree = create a decision node that tests A_{best} in the root
D_v = induced subsets from *D* based on A_{best}
for all D_v **do**
  Tree$_v$ $= C4.5(D_v, S\backslash\{A_{\text{best}}\}, L)$
  attach Tree$_v$ to the corresponding branch of *Tree*
end for
return *Tree*

III. HCFC Framework

A. Basic Underlying Ideas

Clustering results can provide useful information for constructing classification models, and thus, combining unsupervised with supervised learning to construct a hybrid classification model may achieve good classification results. Most existing hybrid models train classifiers on the clusters obtained through clustering, which may make it easier to misclassify the samples near the boundaries of the clusters. Unlike the existing hybrid models, the HCFC proposed in this article does not train classifiers separately on numerous clusters; rather, it builds a clustering-based attribute selection measure, namely, the hybrid information gain ratio, to train a single decision tree classifier. Because the training of the final classifier in the HCFC is applied to the entire training set, it can effectively avoid the shortcomings of the existing hybrid models. Additionally, the hybrid information gain ratio considers both the class labels of the samples and the clusters to which they belong, which may reduce the effect of the class noise on the model to a certain extent.

Let a classification problem include *M* different classes, $L = \{l_m \mid m = 1, 2, ..., M\}$, and *D* and *T* represent the training and test sets, respectively. The HCFC can be roughly divided into the following two steps: 1) clustering, in which any clustering algorithm can be applied to partition the training set into *K* clusters (the class labels of the training samples are not used), where $C = \{c_k \mid k = 1, 2, ..., K\}$, and 2) classification, in which a decision tree is trained on the training set *D* according to the hybrid information gain ratio, and the resulting decision tree is simply the final classifier of the HCFC framework. We can then verify the generalization ability of the final classifier on the test set *T*. As mentioned earlier, we constructed two versions of the HCFC (HCFC-K and HCFC-D), which use an improved K-means model and DBSCAN during the clustering step, respectively. Fig. 1 shows a flowchart of the HCFC framework. In the next subsection, we describe in detail the specific calculation steps for the hybrid information gain ratio.

B. Calculation of Hybrid Information Gain Ratio

In general, a decision tree consists of internal nodes and leaf nodes. Each internal node denotes a test on a splitting attribute that is chosen according to the attribute selection measure applied. In this subsection, one of the internal nodes is taken as an example to illustrate the calculation steps of the proposed hybrid information gain ratio. The notation used herein is as follows. Let D' denotes the sample set of the current node, $|D'|$ represents the number of samples in D', $p(m)$ represents the proportion of samples belonging to class l_m in D', and $q(k)$ represents the proportion of samples belonging to cluster c_k in D'. Furthermore, suppose S is the set of all attributes in D',

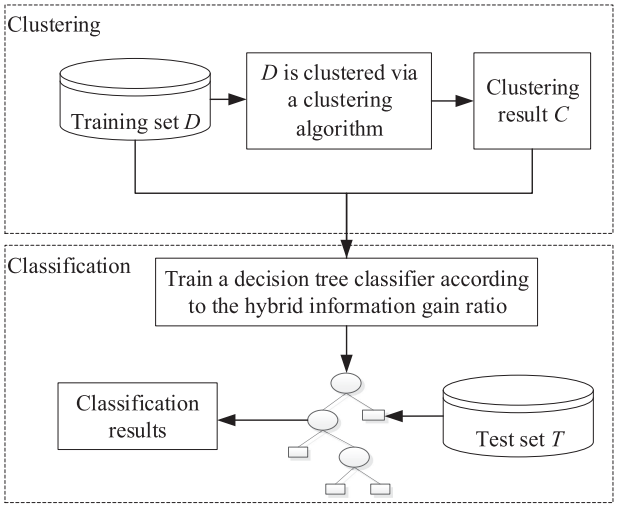

Fig. 1. Flowchart of the HCFC framework.

and that we partition the samples in $D^{'}$ on an attribute $A \in S$ having V distinct values, $\{a_1, a_2, ..., a_v, ..., a_V\}$, as observed from the training set. As a result, $D^{'}$ is divided into V subsets, $\{D_1^{'}, D_2^{'}, ..., D_v^{'}, ..., D_V^{'}\}$, where $D_v^{'}$ contains those samples in $D^{'}$, that have an outcome a_v of A. For class set L and cluster set C, the information entropy of $D^{'}$ can be expressed as follows:

$$\text{Ent}(D^{'}, L) = -\sum_{m=1}^{M} p(m) \times \log p(m) \tag{1}$$

$$\text{Ent}(D^{'}, C) = -\sum_{k=1}^{K} q(k) \times \log q(k). \tag{2}$$

Next, we define the information gain on attribute A separately as follows:

$$\text{Gain}(D^{'}, A, L) = \text{Ent}(D^{'}, L) - \sum_{v=1}^{V} \frac{|D_v^{'}|}{D^{'}} \text{Ent}(D_v^{'}, L) \tag{3}$$

$$\text{Gain}(D^{'}, A, C) = \text{Ent}(D^{'}, C) - \sum_{v=1}^{V} \frac{|D_v^{'}|}{D^{'}} \text{Ent}(D_v^{'}, C). \tag{4}$$

Similarly, according to $\text{Gain}(D^{'}, A, L)$ and $\text{Gain}(D^{'}, A, C)$, we calculate the corresponding information gain ratio as follows:

$$\text{GainRatio}(D^{'}, A, L) = \frac{\text{Gain}(D^{'}, A, L)}{\text{SplitInformation}(D^{'}, A)} \tag{5}$$

$$\text{GainRatio}(D^{'}, A, C) = \frac{\text{Gain}(D^{'}, A, C)}{\text{SplitInformation}(D^{'}, A)} \tag{6}$$

where

$$\text{SplitInformation}(D^{'}, A) = -\sum_{v=1}^{V} \frac{|D_v^{'}|}{|D^{'}|} \log \frac{|D_v^{'}|}{|D^{'}|}. \tag{7}$$

Finally, we define the hybrid information gain ratio as

$$\text{HybridGainRatio}(D^{'}, A, C, L) = \lambda \times \text{GainRatio}(D^{'}, A, C) + \text{GainRatio}(D^{'}, A, L) \tag{8}$$

where the coefficient λ controls the relative importance of $\text{GainRatio}(D^{'}, A, C)$ and $\text{GainRatio}(D^{'}, A, L)$.

The attribute that achieves the highest hybrid information gain ratio will be used to split the current dataset into multiple subsets. In addition, unlike the traditional information gain ratio, the hybrid information gain ratio considers both the class labels of the samples and the clusters to which they belong; therefore, the samples in each subset are more consistent in terms of both their class labels and attributes.

C. Complexity Analysis

Because the HCFC is a framework for building hybrid models, its time and space complexity depend on two aspects, namely, clustering and classification. In this subsection, we take HCFC-K as an example for analysis because the complexity of hybrid models using different clustering algorithms varies.

Assume that dataset *D* contains *N* samples with *W* attributes. During the clustering step, the refined initial cluster centers need to be determined first, and thus, *J* small subsets are selected from the dataset *D* and clustered using the classical K-means algorithm. The time complexity for determining the refined initial cluster centers is $O(N^{'}T^{'}KWJ)$, where $N^{'}$ represents the number of samples in the subset, $T^{'}$ represents the number of iterations required for convergence, and *K* indicates the number of clusters. Then, *D* is clustered according to the obtained initial cluster centers; the time complexity of this part is $O(NTKW)$, where *T* represents the number of iterations required for convergence. Because $N^{'} \ll N, T^{'} < T$, and *J* is generally quite small [27], the time needed to determine the initial cluster centers is negligible. Therefore, the time complexity of the clustering step is $O(NTKW)$. During the classification step, because the calculation of the hybrid information gain ratio does not change the overall time complexity, the time complexity of this step is identical to that of the C4.5 decision tree, which is $O(NW^2)$ [31]. Finally, the time complexity of the two parts is added to obtain the time complexity of HCFC-K as $O(NW(TK + W))$. Because the running time of the HCFC-K increases linearly with the number of samples N and quadratically with the number of attributes W, it is not very efficient in processing high-dimensional datasets. The main reason being the C4.5 decision tree, which is used in the hybrid model, requires a large amount of time to process continuous attributes. In fact, numerous studies on how to reduce the time complexity of C4.5 have been conducted [31]–[33]. If a more effective C4.5 decision tree is adopted, the time complexity of the hybrid model is expected to reduce to $O(NWTK)$.

The space complexities of the classical K-means approach and the method for determining the initial cluster centers are $O((N + K)W)$ and $O((N^{'} + K)WJ)$, respectively. In addition, the space complexity of the C4.5 decision tree is $O(NW)$ [32]. Because $N^{'}$, $K \ll N$, and J is generally

Algorithm 3: HCFC(D, K, S, L).

Input: Training set D; number of clusters K; attributes set S; class set L
Output: Decision tree *Tree*

procedure $Clustering(D)$
  A clustering algorithm is applied to partition the training set D into K clusters, $C = \{c_k \mid k = 1, 2, \ldots, K\}$
  Return cluster set C
end procedure
procedure $Classification(D, S, C, L)$
  $Tree = \{\}$
  if all samples in D are of the same class $l_m \in L$ **then**
    return a single node with the class l_m
  end if
  if S is empty **then**
    return a single node with most frequent class in D
  end if
  for $A \in S$ **do**
    compute HybridGainRatio(D, A, C, L)
  end for
  $A_{\text{best}} = \arg\max_A(\text{HybridGainRatio}(D, A, C, L))$
  $Tree =$ create a decision node that tests A_{best} in the root
  $D_v =$ induced subsets from D based on A_{best}
  for all D_v **do**
    $\text{Tree}_v = \text{Classification}(D_v, S\backslash\{A_{\text{best}}\}, C, L)$
    attach Tree_v to the corresponding branch of *Tree*
  end for
  return *Tree*
end procedure

extremely small, the space complexity of the HCFC-K can be regarded as $O(NW)$.

D. Modeling Process

Algorithm 3 introduces the two main steps of the HCFC. The clustering results returned in the first step will be used as input in the second step. In the second step, the hybrid information gain ratio is used to determine the optimal splitting attributes of the decision tree. After these two steps, we obtain the final decision tree classifier.

IV. Experiments

In this section, we first introduce the experiment datasets and the parameter settings for each model. Second, through experiments, we investigate the effects of some of the important parameters on the HCFC-K performance. Third, we analyze the effects of clustering and the hybrid information gain ratio on the classification performance of the proposed framework. Fourth, we compare the classification performances of the HCFC-K and HCFC-D with that of the six single and three hybrid models. Finally, we compare the classification performance of HCFC-K, HCFC-D, C4.5, and the three hybrid models on datasets containing class noise.

TABLE I
UCI Datasets for Classification Experiments

Dataset	Instance	Attribute	Class	Industry
Abalone	4177	8	28	biology
Auto	205	26	6	business
Blood-Transfusion(BT)	748	5	2	healthcare
Connectionist	208	60	2	others
Dermatology	366	33	6	disease diagnosis
E.coli	336	8	8	healthcare
Flags	194	30	8	others
Flare	1389	10	2	astronomy
Frogs-MFCCs-Family(FMF)	7196	22	4	biology
Glass	214	10	6	materials
Magic	19020	11	2	astronomy
Monk2	432	7	2	others
Parkinsons	197	23	2	disease diagnosis
Seeds	210	7	3	biology
Soybean	307	35	15	disease diagnosis
Teaching-Assistant-Evaluation (TAE)	151	5	3	others
User-Knowledge-Modeling(UKM)	403	5	4	others
Vertebral-column-2C(VC2)	310	6	2	disease diagnosis
Vertebral-column-3C(VC3)	310	6	3	disease diagnosis
Wholesale-Customers(WC)	44	8	2	business
Wilt	4889	6	2	disease diagnosis
Yeast	1484	8	9	biology
Zoo	101	17	7	biology

A. Experimental Setup

To analyze the performance of the HCFC proposed in this article, we conducted experiments on eight benchmark datasets from the healthcare and disease diagnosis industry, as well as 15 datasets from other industries; all were extracted from the University of California at Irvine (UCI) repository [34]. Table I summarizes the properties of all datasets applied. For convenience, the names of some of the datasets are abbreviated. This article compares the HCFC-K and HCFC-D to six commonly used single models, namely, C4.5 [10], KNN [13], SVM [16], NB [17], LR [18], and MLP [35]. Furthermore, it compares HCFC-K and HCFC-D to the three hybrid models, namely, K-means+ID3, proposed by Gaddam *et al.* [20]; RK+SVM, proposed by Rajamohamed *et al.* [25]; and TBWC, proposed by Kaewchinporn *et al.* [21]. It is worth mentioning that the hybrid model WK+FOIL [8] deals only with binary classification problems, and because most of the datasets in this article contain multiple classes, is not considered in this article.

In order to compare the performance of 11 classification models on the 23 datasets considered fairly, we used a fivefold cross-validation method. First, each dataset was divided equally into five subsets at random. During each experiment, one subset was selected as the test set, one subset was randomly selected as the validation set from the other four subsets, and the remaining three subsets were selected as the training set. Next, the training set was used to estimate the model parameters, and the validation set was applied to select the optimal parameters for each model. Finally, the test set was used to evaluate the classification performance of each model with the optimal parameters. The aforementioned process was repeated five times to ensure that each of the five subsets was used as the test set one time, and the entire process underwent a fivefold cross validation.

The parameters of HCFC-K include the coefficient λ in (8), the number of clusters K, and the number of subsets J. During the

experiment, λ was set to within the range of [0.1, 1], with a step size of 0.1, and K was set to within a range of [2, 8], also with the step size of 1. Because we found during the experiment that the parameter J has little effect on the performance of the model, we set $J = 10$, as per [27]. The parameters of the HCFC-D include the radius of the cluster (ϵ) and the minimum samples required inside the cluster (*MinPts*), where ϵ was set to within of the range [0.1, 1], with a step size of 0.1, and *MinPts* was set within the range of [2, 10], with a step size of 1. Both K-means+ID3 and RK+SVM need to determine the number of clusters K. According to [20] and [25], K in K-means+ID3 is set to within a range of [2, 20], with a step size of 1, and K in RK+SVM is set to within a range of [2, 8], also with a step size of 1. The parameters of TBWC include the number of decision trees num_trees and the number of clusters K. According to [21], we let both num_trees and K have a range of [5, 40] with a step size of 5. For the KNN, the parameter k is set within the range of [1, 9], with a step size of 2. In this article, we set up two hidden layers for MLP and use ReLU as the activation function. The optimal number of neurons in the hidden layer is varied from 1 to 20 with a step size of 1. We chose the library for support vector machines (LIBSVM) [36] to achieve the SVM classifier and used the radial basis function kernel. In LIBSVM, the kernel parameter γ is set to $1/W$ by default, where W represents the number of attributes of the dataset. In addition, the penalty parameter P was selected from (1, 10, 100, 1000) [37]. For C4.5, it is necessary to select the confidence factor to determine the degree of pruning. Here, we allowed it to be 25%, as suggested by Quinlan [10]. Because NB and K-means+ID3 can handle only discrete attributes, we used the method proposed by Fayyad and Irani [38] to discretize the continuous attributes prior to the modeling.

Finally, in order to further enhance the reliability of the experimental results, the aforementioned fivefold cross validation was repeated ten times and the average was applied as the final result. All models were implemented in Python 2.7.

B. Effects of the Parameters on HCFC-K Performance

In this subsection, we take HCFC-K as an example to investigate the effects of the parameters on the classification performance of the proposed framework. Specifically, we considered two parameters, namely, the number of clusters K and the coefficient λ. Given a lack of prior knowledge, when studying the effects of parameter K on the model, we temporarily set $\lambda = 1$. After determining the optimal K for each dataset, we investigated the effects of parameter λ on the model performance.

1) Effects of Parameter K on the Performance of HCFC-K: Because the distribution of samples in each dataset differs, it is clearly unreasonable to use the same parameter K with all datasets. Therefore, we selected the optimal parameter K for each dataset on the validation set mentioned earlier. Table II shows the classification accuracy of the HCFC-K on the validation set when K takes different values. According to the results in Table II, we can determine the corresponding parameter K for each dataset.

TABLE II
CLASSIFICATION ACCURACY (%) OF HCFC-K ON ALL THE DATASETS WHEN K TAKES DIFFERENT VALUES

Dataset	K						
	2	3	4	5	6	7	8
Abalone	24.86	**25.13**	23.50	23.36	23.37	23.13	23.13
Auto	73.17	72.68	76.59	77.07	76.10	75.12	**77.07**
BT	75.67	74.87	74.74	75.68	75.94	**76.35**	75.54
Connectionist	78.82	**78.89**	76.47	76.49	76.45	74.58	72.58
Dermatology	92.64	**94.54**	92.89	92.62	92.08	91.27	90.99
Ecoli	**82.46**	80.97	81.87	81.86	81.87	80.97	81.86
Flags	**61.81**	60.82	58.76	58.70	60.80	58.79	57.73
Flare	84.70	**85.98**	85.43	85.48	85.25	85.56	85.87
FMF	91.70	92.63	90.96	90.83	91.34	**94.17**	90.45
Glass	67.97	69.03	**69.32**	67.14	68.15	67.52	67.52
Magic	81.09	**81.72**	81.48	78.82	79.90	81.01	79.31
Monk2	86.80	85.32	85.76	84.76	86.41	**87.86**	84.76
Parkinsons	88.21	87.69	85.13	83.08	86.15	**89.23**	85.64
Seeds	93.81	91.43	91.43	92.86	93.33	93.81	**95.71**
Soybean	88.25	88.79	88.97	88.97	88.25	**89.15**	88.07
TAE	**62.95**	61.61	60.90	61.57	60.22	62.28	62.92
UKM	91.31	92.29	**93.79**	91.82	92.05	92.81	92.05
VC2	81.61	81.61	81.61	82.26	81.94	81.94	**82.90**
VC3	82.90	82.26	82.26	82.26	81.94	81.29	**83.55**
WC	89.32	89.32	89.09	89.77	**90.23**	89.55	89.89
Wilt	**95.36**	93.87	92.13	88.23	92.08	93.57	93.60
Yeast	50.20	50.07	50.05	**51.64**	50.35	50.24	50.13
Zoo	**98.05**	96.10	96.10	97.05	96.14	97.00	97.10

TABLE III
CLASSIFICATION ACCURACY (%) OF HCFC-K ON ALL THE DATASETS WHEN λ TAKES DIFFERENT VALUES

Dataset	λ									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Abalone	23.25	**24.85**	23.31	23.71	23.23	23.18	23.05	24.19	23.17	24.84
Auto	75.12	73.17	73.17	74.15	74.15	72.20	72.20	71.71	76.10	**76.59**
BT	**76.88**	76.34	75.67	75.40	75.40	75.01	75.13	75.01	75.00	74.60
Connectionist	74.96	74.97	74.58	**79.30**	77.86	76.40	75.92	76.90	78.35	78.35
Dermatology	92.34	92.34	92.89	92.34	92.34	91.80	91.80	91.80	92.63	**94.53**
Ecoli	81.26	82.15	81.28	81.55	**82.72**	80.97	81.55	81.26	80.65	80.65
Flags	63.93	**65.99**	65.48	65.45	65.98	65.96	65.96	65.96	63.39	61.84
Flare	85.84	85.15	86.19	**86.87**	82.98	85.61	86.55	85.78	86.41	85.75
FMF	93.46	95.03	94.58	94.94	**96.32**	92.80	91.28	92.11	92.41	90.65
Glass	65.84	68.83	71.37	70.86	**72.94**	70.40	69.23	71.90	72.08	68.99
Magic	81.81	82.67	**83.23**	81.96	80.97	81.74	81.14	82.32	82.23	81.44
Monk2	86.36	87.44	**88.75**	87.63	84.39	85.25	86.19	88.09	87.61	87.96
Parkinsons	88.72	88.72	87.18	88.72	88.21	84.10	85.13	85.13	88.72	**89.23**
Seeds	92.86	92.86	93.33	91.90	91.90	**93.81**	92.48	92.48	92.86	92.38
Soybean	88.61	88.79	**89.86**	86.83	86.65	87.90	88.26	87.72	87.54	87.18
TAE	60.92	60.92	**62.92**	61.66	61.66	60.32	60.32	60.32	60.24	60.24
UKM	91.31	91.31	91.31	92.54	92.54	**92.55**	91.81	91.81	91.31	91.31
VC2	83.87	**84.52**	82.90	81.94	81.94	81.29	81.29	79.68	81.61	81.61
VC3	**83.55**	82.90	81.29	80.65	80.65	80.97	80.97	81.94	82.26	82.58
WC	89.32	89.32	89.32	89.55	89.55	**90.23**	89.32	89.32	89.32	89.32
Wilt	97.25	97.76	97.78	**98.50**	98.39	98.33	97.81	96.79	96.97	96.34
Yeast	47.36	51.58	52.52	52.31	49.75	49.68	48.49	**53.02**	52.36	52.35
Zoo	98.00	98.00	98.00	**98.05**	97.10	97.10	97.10	97.10	97.00	97.05

2) Effects of Parameter λ on the Performance of HCFC-K: Table III shows the classification accuracy of the HCFC-K on the validation set when $\lambda = 0.1, 0.2, ..., 1$. The maximum value in each row is shown in bold. According to the results in Table III, we can determine the corresponding parameter λ for each dataset.

C. Influence of Clustering and Hybrid Information Gain Ratio on the Performance of HCFC

To analyze the influence of the clustering and hybrid information gain ratio on the performance of the HCFC, we compared HCFC-K and HCFC-D to the following four models.

TABLE IV
COMPARISON OF CLASSIFICATION ACCURACY (MEAN ± STANDARD DEVIATION, %) OF SIX MODELS

Dataset	HCFC-K	HCFC-K1	HCFC-D	IK+C4.5	DB+C4.5	C4.5
Abalone	**26.05 (1) ± 0.46**	24.14 (5) ± 1.30	25.91 (2) ± 0.68	24.21 (3) ± 0.74	24.19 (4) ± 2.30	23.10 (6) ± 1.43
Auto	**76.10 (1) ± 2.02**	74.27 (3) ± 4.86	74.39 (2) ± 3.19	68.68 (6) ± 4.92	69.98 (5) ± 5.25	74.15 (4) ± 5.42
BT	**78.48 (1) ± 3.54**	78.08 (2) ± 4.21	77.94 (3) ± 3.24	77.84 (4) ± 4.39	75.89 (6) ± 5.42	77.57 (5) ± 3.01
Connectionist	**79.88 (1) ± 5.35**	77.85 (2) ± 4.21	76.68 (5) ± 6.92	77.07 (4) ± 5.89	77.24 (3) ± 5.59	72.60 (6) ± 7.39
Dermatology	93.98 (2) ± 2.90	93.75 (3) ± 1.41	**94.27 (1) ± 3.60**	76.34 (6) ± 3.93	80.41 (5) ± 3.53	90.44 (4) ± 3.13
Ecoli	83.05 (2) ± 4.50	82.03 (4) ± 2.30	**83.78 (1) ± 2.75**	81.90 (5) ± 2.69	82.47 (3) ± 4.38	79.16 (6) ± 3.12
Flags	63.43 (2) ± 3.07	57.73 (4) ± 5.24	**63.66 (1) ± 2.53**	49.24 (6) ± 2.47	53.47 (5) ± 4.15	61.88 (3) ± 2.77
Flare	84.70 (2) ± 4.25	82.99 (3) ± 2.87	**85.11 (1) ± 3.00**	80.73 (6) ± 3.73	82.43 (4) ± 8.07	81.76 (5) ± 8.98
FMF	**96.13 (1) ± 0.52**	94.83 (4) ± 0.65	95.57 (2) ± 0.82	94.87 (3) ± 0.15	93.72 (5) ± 0.67	93.66 (6) ± 0.23
Glass	70.73 (3) ± 3.55	68.61 (4) ± 8.15	70.78 (2) ± 8.64	67.88 (5) ± 1.62	**72.22 (1) ± 7.06**	60.31 (6) ± 7.87
Magic	**82.51 (1) ± 0.49**	81.23 (3) ± 0.33	80.81 (4) ± 0.91	80.41 (6) ± 0.52	81.35 (2) ± 0.36	80.68 (5) ± 0.32
Monk2	**86.36 (1) ± 2.87**	85.79 (3) ± 2.87	86.20 (2) ± 3.10	83.99 (4) ± 3.22	83.70 (5) ± 5.67	80.53 (6) ± 1.12
Parkinsons	**91.28 (1) ± 3.94**	86.92 (3) ± 5.06	87.95 (2) ± 6.13	78.13 (6) ± 4.05	86.44 (4) ± 4.33	82.05 (5) ± 5.82
Seeds	**91.90 (1) ± 2.72**	91.10 (3) ± 6.44	90.48 (6) ± 5.39	90.55 (5) ± 4.76	91.14 (2) ± 7.45	90.95 (4) ± 5.09
Soybean	**89.67 (1) ± 3.29**	87.86 (3) ± 2.08	88.96 (2) ± 2.05	82.76 (6) ± 2.43	85.37 (5) ± 4.62	85.40 (4) ± 0.77
TAE	64.90 (2) ± 3.85	59.91 (4) ± 5.68	**66.55 (1) ± 4.94**	56.10 (6) ± 6.42	57.95 (5) ± 5.71	60.26 (3) ± 4.19
UKM	**92.31 (1) ± 0.58**	91.89 (2) ± 2.71	91.07 (3) ± 0.37	84.05 (5) ± 4.92	81.70 (6) ± 5.03	89.82 (4) ± 0.09
VC2	**83.55 (1) ± 2.45**	82.18 (3) ± 4.65	82.58 (2) ± 4.90	81.40 (5) ± 1.74	80.55 (6) ± 1.86	81.45 (4) ± 4.05
VC3	81.29 (4) ± 2.65	82.11 (2) ± 4.61	**83.39 (1) ± 2.91**	81.29 (4) ± 2.61	80.87 (5) ± 3.53	79.35 (6) ± 3.51
WC	**90.68 (1) ± 2.67**	89.52 (3) ± 4.00	90.16 (2) ± 1.23	87.66 (6) ± 2.82	88.10 (5) ± 2.77	88.41 (4) ± 2.54
Wilt	**98.22 (1) ± 0.18**	97.71 (2) ± 1.47	97.59 (3) ± 0.43	96.59 (5) ± 0.48	96.65 (4) ± 0.04	96.36 (6) ± 1.78
Yeast	51.49 (3) ± 2.84	50.39 (4) ± 2.83	53.40 (2) ± 2.63	**54.81 (1) ± 2.72**	50.36 (5) ± 3.40	46.36 (6) ± 3.61
Zoo	**99.05 (1) ± 2.05**	97.39 (3) ± 5.74	97.55 (2) ± 2.73	96.15 (4) ± 4.14	94.08 (5) ± 5.11	91.10 (6) ± 2.51
Average	**80.68 (1.52) ± 2.64**	79.06 (3.13) ± 3.64	80.21 (2.26) ± 3.18	76.20 (4.83) ± 3.10	76.97 (4.35) ± 4.19	76.84 (4.96) ± 3.43

1) HCFC-K1, which uses the classical K-means algorithm directly during the clustering step (without refining the initial cluster centers), as opposed to the hybrid information gain ratio, which is used during the classification step, similar to HCFC-K.
2) IK+C4.5, which is similar to K-means+ID3 but uses the improved K-means algorithm and classical C4.5 algorithm during the clustering and classification steps, respectively.
3) DB+C4.5, which is similar to K-means+ID3 but uses DBSCAN and the classical C4.5 algorithm during the clustering and classification step, respectively.
4) C4.5 (i.e., the classical C4.5 model).

Table IV shows the classification accuracy (mean ± standard deviation) of the six different models on the 23 UCI datasets. For each dataset, the highest classification accuracy is shown in bold.

To verify whether there are statistically significant differences in the classification performance among the six models, we adopted nonparametric methods, namely, the Friedman test [39] and the Iman–Davenport test [40]. As the null hypothesis of the Friedman and Iman–Davenport tests, the six classification models achieve the same classification performance. If this null hypothesis is rejected, an additional Nemenyi *post hoc* test [41] will be conducted to compare the six different models with each other. In this article, we let the significance level $\alpha = 0.05$. We sort the classification accuracy of the models on each dataset, starting with the highest (=1); ties receive a rank equal to the average values they span. The final row of Table IV shows the average rank of each model.

During this experiment, the Friedman test and Iman-Davenport test statistics follow a χ^2 distribution with five degrees of freedom and an F-distribution with 5×110 degrees of freedom, respectively. Table V shows the test results. Because the test values of both tests exceed the corresponding critical values, we reject the null hypothesis. Thus, we can conclude that, at a 95% confidence level, statistically significant differences occur in the classification performance of the six models. Next, we applied a Nemenyi *post hoc* test to compare the six different models with each other. When the number of classification models is six, the critical value is 2.85, and the critical difference (CD) is $2.85\sqrt{(6 \times 7/(6 \times 23))} = 1.57$. Fig. 2 shows the test results; here, there is no statistically significant difference among the models connected by a line segment. From Fig. 2, we can draw the following conclusions.

1) At a 95% confidence level, there is no statistically significant difference between the HCFC-K and HCFC-D,

TABLE V
RESULTS OF THE FRIEDMAN AND IMAN-DAVENPORT TESTS FOR COMPARING PERFORMANCE

Method	Test value	Distribution value	Hypothesis
Friedman	68.93	11.07	reject
Iman-Davenport	32.91	2.30	reject

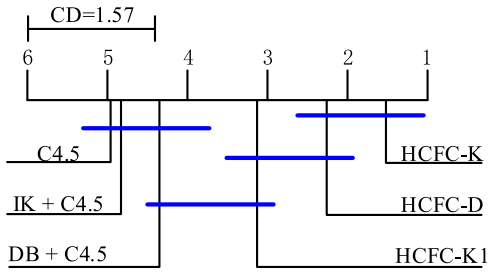

Fig. 2. Nemenyi *post hoc* test results for six models.

TABLE VI
COMPARISON OF CLASSIFICATION ACCURACY (MEAN ± STANDARD DEVIATION, %) OF 11 MODELS ON 23 UCI DATASETS

Dataset	HCFC-K	HCFC-D	C4.5 [10]	SVM [16]	LR [18]	KNN [13]	NB [17]	MLP [35]	RK+SVM [25]	K-means+ID3 [20]	TBWC [21]
Abalone	26.05 (2) ± 0.46	25.91 (3) ± 0.68	23.10 (10) ± 1.43	25.16 (4) ± 2.44	23.80 (8) ± 1.61	23.37 (9) ± 1.90	24.42 (7) ± 0.68	**27.45 (1) ± 1.12**	24.85 (6) ± 2.47	22.55 (11) ± 1.68	24.89 (5) ± 1.26
Auto	**76.10 (1) ± 2.02**	74.39 (2) ± 3.19	74.15 (3) ± 5.42	65.37 (6) ± 6.13	63.90 (7) ± 5.77	60.48 (8) ± 7.67	38.05 (11) ± 9.19	58.29 (9) ± 7.89	73.17 (4) ± 8.69	69.34 (5) ± 3.43	46.59 (10) ± 4.98
BT	**78.48 (1) ± 3.54**	77.94 (2) ± 3.24	77.57 (4) ± 3.01	76.20 (7) ± 3.63	77.80 (3) ± 4.08	75.27 (10) ± 1.86	75.95 (9) ± 4.28	77.27 (5) ± 3.68	70.73 (11) ± 4.12	76.18 (8) ± 3.27	77.20 (6) ± 3.76
Connectionist	79.88 (2) ± 5.35	76.68 (5) ± 6.92	72.60 (9) ± 7.39	76.41 (7) ± 7.88	76.46 (6) ± 3.25	**80.98 (1) ± 3.82**	69.23 (11) ± 6.85	78.42 (3) ± 2.83	69.73 (10) ± 10.27	76.82 (4) ± 3.81	74.54 (8) ± 6.02
Dermatology	93.98 (4) ± 2.90	94.27 (3) ± 3.60	90.44 (6) ± 3.13	96.99 (2) ± 2.27	**98.36 (1) ± 1.24**	87.43 (8) ± 3.32	87.97 (7) ± 2.62	84.42 (10) ± 6.08	93.03 (5) ± 2.28	87.15 (9) ± 4.99	80.56 (11) ± 3.52
Ecoli	83.05 (4) ± 4.50	83.78 (3) ± 2.75	79.16 (9) ± 3.12	86.89 (2) ± 3.49	75.30 (11) ± 4.43	**87.34 (1) ± 3.93**	78.60 (10) ± 3.81	81.26 (8) ± 6.03	82.88 (5) ± 4.08	81.89 (7) ± 5.73	81.99 (6) ± 2.18
Flags	63.43 (2) ± 3.07	**63.66 (1) ± 2.53**	61.88 (3) ± 2.77	32.54 (11) ± 4.37	44.37 (6) ± 7.82	38.64 (7) ± 4.84	45.88 (5) ± 6.28	36.35 (8) ± 7.18	33.79 (10) ± 1.95	54.64 (4) ± 2.25	33.82 (9) ± 3.51
Flare	84.70 (2) ± 4.25	**85.11 (1) ± 3.00**	81.76 (9) ± 8.98	84.23 (4) ± 3.62	84.54 (3) ± 4.27	83.75 (6) ± 3.42	51.05 (11) ± 5.74	83.30 (8) ± 8.66	83.46 (7) ± 3.69	81.50 (10) ± 6.04	83.91 (5) ± 3.75
FMF	96.13 (3) ± 0.52	95.57 (4) ± 0.82	93.66 (8) ± 0.23	93.48 (9) ± 0.59	92.38 (10) ± 0.42	**98.95 (1) ± 0.51**	87.46 (11) ± 0.95	97.21 (2) ± 0.41	93.91 (7) ± 0.58	95.53 (5) ± 1.02	94.25 (6) ± 1.19
Glass	70.73 (2) ± 3.55	**70.78 (1) ± 8.64**	60.31 (6) ± 7.87	49.56 (10) ± 8.39	55.63 (9) ± 4.22	66.43 (4) ± 7.55	45.75 (11) ± 11.83	57.92 (7) ± 6.63	61.41 (5) ± 8.16	68.67 (3) ± 4.02	57.19 (8) ± 6.10
Magic	**82.51 (1) ± 0.49**	80.81 (3) ± 0.91	80.68 (4) ± 0.32	69.28 (10) ± 0.59	78.87 (8) ± 0.86	80.60 (5) ± 0.38	72.83 (9) ± 1.17	79.57 (7) ± 0.70	67.16 (11) ± 0.63	81.06 (2) ± 0.52	79.82 (6) ± 0.75
Monk2	86.36 (2) ± 2.87	86.20 (3) ± 3.10	80.53 (4) ± 1.12	64.73 (8) ± 1.18	65.39 (6) ± 1.37	59.73 (11) ± 2.30	63.73 (10) ± 2.48	**86.87 (1) ± 7.98**	64.39 (9) ± 1.41	80.37 (5) ± 6.49	64.89 (7) ± 2.49
Parkinsons	**91.28 (1) ± 3.94**	87.95 (2) ± 6.13	82.05 (6) ± 5.82	73.33 (10) ± 4.17	84.62 (5) ± 1.76	85.64 (4) ± 1.79	72.31 (11) ± 8.63	78.46 (8) ± 5.24	76.67 (9) ± 3.71	85.86 (3) ± 1.29	81.79 (7) ± 3.18
Seeds	91.90 (5) ± 2.72	90.48 (8) ± 5.39	90.95 (7) ± 5.09	**94.76 (1) ± 4.12**	92.38 (3) ± 2.39	89.05 (11) ± 2.79	89.52 (9) ± 6.08	92.14 (4) ± 1.40	94.29 (2) ± 3.21	91.43 (6) ± 6.28	89.29 (10) ± 5.33
Soybean	89.67 (2) ± 3.29	88.96 (3) ± 2.05	85.40 (6) ± 0.77	**91.63 (1) ± 4.97**	87.90 (5) ± 2.22	77.58 (9) ± 3.42	84.52 (7) ± 4.47	74.18 (10) ± 2.45	88.25 (4) ± 4.67	83.81 (8) ± 2.16	69.67 (11) ± 2.71
TAE	64.90 (2) ± 3.85	**66.55 (1) ± 4.94**	60.26 (3) ± 4.19	42.90 (9) ± 3.61	44.41 (7) ± 9.48	39.66 (11) ± 7.38	52.95 (4) ± 9.44	41.63 (10) ± 9.84	50.71 (6) ± 2.61	52.87 (5) ± 7.31	42.94 (8) ± 3.09
UKM	92.31 (3) ± 0.58	91.07 (4) ± 0.37	89.82 (5) ± 0.09	84.86 (10) ± 5.01	75.97 (11) ± 8.10	86.74 (8) ± 3.39	89.10 (6) ± 4.88	**95.29 (1) ± 2.17**	87.11 (7) ± 6.46	85.61 (9) ± 4.24	92.69 (2) ± 2.66
VC2	83.55 (2) ± 2.45	82.58 (3) ± 4.90	81.45 (5) ± 4.05	79.35 (9) ± 4.53	80.97 (8) ± 4.27	**83.72 (1) ± 1.92**	78.39 (10) ± 5.97	81.61 (4) ± 5.60	73.39 (11) ± 6.82	81.01 (7) ± 2.01	81.29 (6) ± 5.24
VC3	81.29 (6) ± 2.65	**83.39 (1) ± 2.91**	79.35 (10) ± 3.51	80.32 (8) ± 8.79	81.94 (4) ± 4.48	83.23 (2) ± 2.91	82.58 (3) ± 4.25	81.61 (5) ± 7.48	76.61 (11) ± 8.56	80.23 (9) ± 2.24	80.81 (7) ± 2.10
WC	90.68 (2) ± 2.67	90.16 (4) ± 1.23	88.41 (7) ± 2.54	72.73 (11) ± 5.38	89.32 (6) ± 4.32	89.77 (5) ± 3.13	90.23 (3) ± 2.25	**90.80 (1) ± 5.17**	86.97 (8) ± 5.58	86.82 (9) ± 1.99	74.32 (10) ± 2.93
Wilt	**98.22 (1) ± 0.18**	97.59 (3) ± 0.43	96.36 (6) ± 1.78	94.17 (10) ± 0.41	95.02 (9) ± 0.32	96.50 (5) ± 0.02	89.73 (11) ± 0.27	97.71 (2) ± 0.28	95.05 (8) ± 0.55	96.86 (4) ± 2.44	95.87 (7) ± 1.29
Yeast	51.49 (6) ± 2.84	53.40 (4) ± 2.63	46.36 (10) ± 3.61	**58.42 (1) ± 2.60**	50.68 (7) ± 2.30	57.28 (2) ± 1.75	22.05 (11) ± 2.48	55.83 (3) ± 2.66	49.73 (9) ± 2.68	53.27 (5) ± 5.08	50.27 (8) ± 2.10
Zoo	**99.05 (1) ± 2.05**	97.55 (2) ± 2.73	91.10 (7) ± 2.51	95.00 (5) ± 7.49	87.00 (11) ± 7.16	89.43 (8) ± 9.19	96.00 (4) ± 3.57	89.02 (9) ± 3.56	87.29 (10) ± 7.33	96.73 (3) ± 3.74	92.10 (6) ± 6.84
Average	**80.68 (2.48) ± 2.64**	80.21 (2.87) ± 3.18	76.84 (6.39) ± 3.43	73.41 (6.74) ± 4.16	74.22 (6.70) ± 3.74	74.85 (5.96) ± 3.44	69.06 (8.30) ± 4.70	75.07 (5.48) ± 4.57	73.24 (7.61) ± 4.37	76.96 (6.13) ± 3.57	71.77 (7.35) ± 3.35

both of which show a good classification performance. This result indicates that the framework proposed in this article does not have strict restrictions on the selection of the clustering algorithms and is extremely flexible in industrial applications.

2) The classification performance of the HCFC-K is significantly better than that of IK+C4.5. Likewise, the classification performance of the HCFC-D is significantly better than that of DB+C4.5. These results indicate that the hybrid information gain ratio is a more effective method for constructing a hybrid model.
3) No statistically significant difference can be seen between the HCFC-D and the HCFC-K1, but the classification performance of the HCFC-K is significantly better than that of the HCFC-K1. This may be due to the improved K-means algorithm used in the HCFC-K obtaining more reasonable clustering results than the classic K-means approach.

D. Comparison of Classification Performance to That of Other Models

Table VI compares the classification accuracy of the HCFC-K and HCFC-D to that of the three hybrid and six single models on all datasets. The HCFC-K and HCFC-D have a higher average classification accuracy with a lower average standard deviation than the other models. Using the same process described in Section IV-C, we use a Friedman test and an Iman–Davenport test to study whether the 11 models exhibit any statistically significant differences; Table VII shows the test results. Because test values of both tests exceed the corresponding critical values, we reject the null hypothesis. Thus, we can conclude that, at a 95% confidence level, statistically significant differences occur among the classification performances of the 11 models considered. Next, we conducted a Nemenyi *post hoc* test to compare the 11 different models. From Fig. 3, we can draw the following conclusions.

TABLE VII
RESULTS OF THE FRIEDMAN AND IMAN-DAVENPORT TESTS FOR COMPARING PERFORMANCE

Method	Test value	Distribution value	Hypothesis
Friedman	69.82	18.31	reject
Iman-Davenport	9.59	1.87	reject

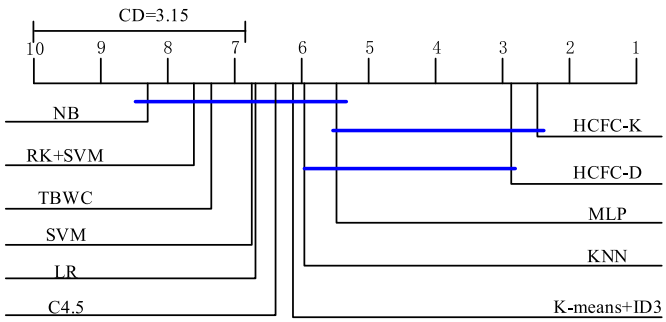

Fig. 3. Nemenyi *post hoc* test results for all models.

1) At a 95% confidence level, no statistically significant difference is shown among HCFC-K, HCFC-D, and MLP, which achieved the best classification performances. Nevertheless, the advantage of HCFC-K and HCFC-D over the MLP is that their results are easier to interpret.
2) The classification performances of the HCFC-K and HCFC-D are significantly better than those of the hybrid models K-means+ID3 and RK+SVM, and TBWC, which indicates that the HCFC framework proposed in this article has clear advantages over the existing hybrid models.
3) No statistically significant difference among MLP, KNN, K-means+ID3, C4.5, LR, SVM, TBWC, RK+SVM, and NB is demonstrated. Likewise, no statistically significant difference among HCFC-D, MLP, and KNN is shown. Overall, the classification performances of the hybrid

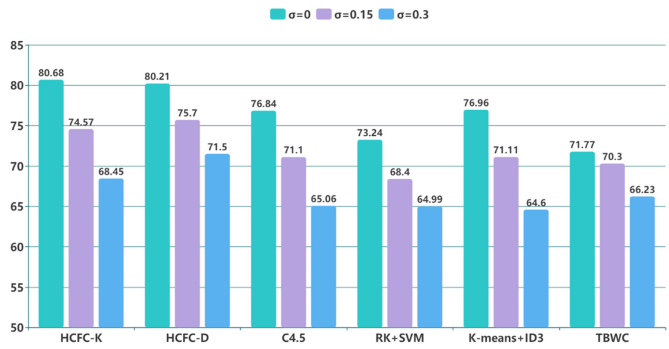

Fig. 4. Average accuracy of six models on 23 datasets when the noise level $\sigma = 0$, 0.15, and 0.3.

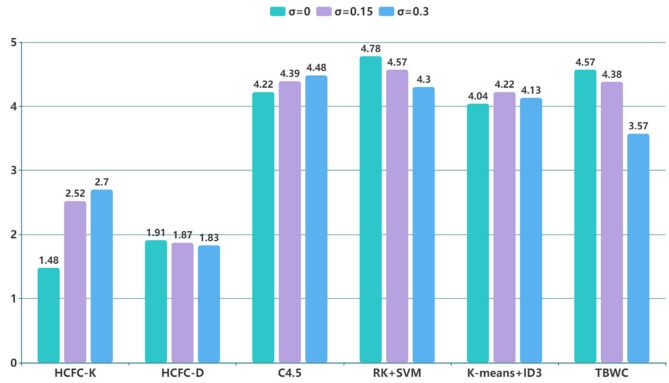

Fig. 5. Average ranking of six models on 23 datasets when the noise level $\sigma = 0$, 0.15, and 0.3.

models HCFC-K, HCFC-D, and K-means+ID3 are relatively good, whereas the classification performances of the hybrid models RK+SVM and TBWC are relatively poor. This indicates that it is extremely important to choose a proper method for building a hybrid model: if an appropriate method is not chosen, it may not be possible to improve the classification performance of the hybrid model by very much.

E. Comparison of Classification Performance in Noisy Environment

In reality, classification problems often contain a considerable amount of noise, and it is very meaningful to compare the performance of classification models using noisy datasets. In general, noise within a dataset can be classified as class or attribute noise. Some studies have shown that class noise is potentially more harmful than attribute noise [42]; therefore, we mainly compared the classification accuracies of HCFC-K, HCFC-D, C4.5, and the three hybrid models on datasets containing class noise. Because most of the UCI datasets applied in this article do not contain class noise, to add such noise we used a manual mechanism adopted by Zhu and Wu [43]. Fig. 4 shows the average accuracy of the six models on 23 datasets when the noise level is $\sigma = 0$, 0.15, and 0.3. Fig. 5 shows the average ranking of the six models on the 23 datasets when the noise level is $\sigma = 0$, 0.15, and 0.3.

TABLE VIII
RESULTS OF THE FRIEDMAN AND IMAN-DAVENPORT TESTS ($\sigma = 0.15$)

Method	Test value	Distribution value	Hypothesis
Friedman	41.82	11.07	reject
Iman-Davenport	12.57	2.30	reject

TABLE IX
RESULTS OF THE FRIEDMAN AND IMAN-DAVENPORT TESTS ($\sigma = 0.3$)

Method	Test value	Distribution value	Hypothesis
Friedman	35.84	11.07	reject
Iman-Davenport	9.96	2.30	reject

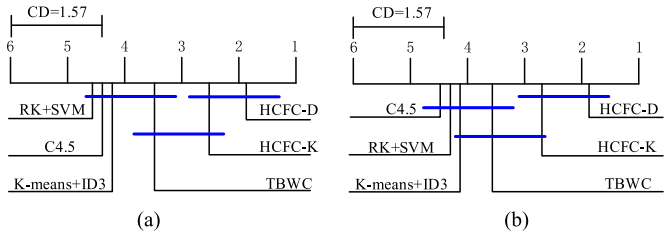

Fig. 6. Nemenyi *post hoc* test results for the six models. (a) $\sigma = 0.15$. (b) $\sigma = 0.3$.

We used a Friedman test and an Iman–Davenport test to further investigate whether the six models exhibit any statistically significant differences. Tables VIII and IX show the test results, respectively. From these tables, it can be seen that, under different noise levels, the test values of both tests exceed the corresponding critical values, thereby, indicating the presence of statistically significant differences among the six models. Next, we used a Nemenyi *post hoc* test to compare the six models. Fig. 6 shows the test results when the noise level is $\sigma = 0.15$ and 0.3. Finally, we can draw the following conclusions.

1) When $\sigma = 0.15$, the classification performance of the HCFC-D is significantly better than that of the TBWC, K-means+ID3, C4.5, and RK+SVM at a 95% confidence level. Likewise, the classification performance of the HCFC-K is significantly better than that of K-means+ID3, C4.5, and RK+SVM. No statistically significant difference is shown between the HCFC-D and HCFC-K, or between the HCFC-K and TBWC. In addition, no statistically significant difference is demonstrated among TBWC, K-means+ID3, C4.5, and RK+SVM.
2) When $\sigma = 0.3$, the classification performance of the HCFC-D is significantly better than that of the TBWC, RK+SVM, K-means+ID3, and C4.5, and the classification performance of the HCFC-K is significantly better than that of the RK+SVM and C4.5. In addition, there is no statistically significant difference between the HCFC-D and HCFC-K, and there is no statistically significant difference among HCFC-K, TBWC, and K-means+ID3. Likewise, no statistically significant difference is shown among TBWC, K-means+ID3, C4.5, and RK+SVM.

3) As indicated in Figs. 4 and 5, when the noise level increases, the average rankings of the HCFC-K and C4.5 increase gradually, whereas the average rankings of HCFC-D, RK+SVM, and TBWC decrease. In addition, the average accuracy and ranking of the HCFC-D is better than those of the HCFC-K in a noisy environment. Therefore, the antinoise ability of the HCFC-D is better than that of the HCFC-K. When a large amount of class noise is present in a dataset, the HCFC-D may be a better choice.

V. Conclusion

In this article, we proposed an HCFC. Depending on the differences of the clustering algorithms used, we constructed two versions of the HCFC (HCFC-K and HCFC-D), which use an improved K-means approach and DBSCAN, respectively. Unlike a traditional supervised classification model, the HCFC combines the advantages of supervised and unsupervised learning, and integrates the information obtained through clustering into the training process of the C4.5 decision tree classifier. We conducted a series of experiments on eight benchmark datasets from the healthcare and disease diagnosis industry, and 15 datasets from other industries to evaluate the classification performance of the HCFC. The experiment results showed that, when the training sets do not contain class noise, HCFC-K and HCFC-D exhibit a comparable or even better classification performance than other single and hybrid models. When the training sets contain class noise, the classification performances of the HCFC-K and HCFC-D are still superior to those of C4.5 and the existing three hybrid models. In addition, HCFC-D has a stronger ability to resist class noise compared with the HCFC-K. Therefore, compared with the existing single and hybrid classification models, the HCFC has clear advantages in its classification performance under various environments, making it a favorable choice for practical industrial use.

In future research, we will focus on the following four aspects. First, to further enhance the practicability of the hybrid models proposed in this article, we will explore a better way to automatically determine the parameters of the hybrid models and further reduce their time complexity. In addition, ensemble learning has been one of the hotspots in the field of machine learning in recent years, the basic idea of which is to combine a series of weak learners to enhance their performance [44], [45]. To further improve the classification performance of a hybrid model, we will attempt to combine the hybrid information gain ratio with the random forest. Next, in this article, we did not consider the class imbalance of datasets during the modeling process, and thus, we plan to preprocess the unbalanced datasets and consider new evaluation indicators such as the area under the receiver operating characteristic in our next study. Finally, we will also explore the application of hybrid models in semisupervised learning. In many practical classification problems, it is frequently difficult to obtain labeled samples; however, numerous unlabeled samples are available [46], [47]. In this case, using semisupervised learning techniques to build a classification model is a common practice. In this article, we found that the use of clustering information can improve the classification performance of the proposed model, and that the clustering does not need to apply the class labels of the samples; for these reasons, research into a semisupervised hybrid model holds considerable promise.

Acknowledgment

The authors would like to thank the anonymous referees and the editor for their helpful comments, which have played an important role in improving this article.

References

[1] M. Kafai and B. Bhanu, "Dynamic Bayesian networks for vehicle classification in video," *IEEE Trans. Ind. Inform.*, vol. 8, no. 1, pp. 100–109, Feb. 2012.

[2] Z. Ge, S. Zhong, and Y. Zhang, "Semi-supervised kernel learning for FDA model and its application for fault classification in industrial processes," *IEEE Trans. Ind. Inform.*, vol. 12, no. 4, pp. 1403–1411, Aug. 2016.

[3] B. Chen, J. Li, G. Wei, and B. Ma, "A novel localized and second order feature coding network for image recognition," *Pattern Recognit.*, vol. 76, no. 4, pp. 339–348, Apr. 2018.

[4] T. Shon and J. Moon, "A hybrid machine learning approach to network anomaly detection," *Inf. Sci.*, vol. 177, no. 18, pp. 3799–3821, Sep. 2007.

[5] M. Pinol, R. Alves, I. Teixidó, J. Mateo, F. Solsona, and E. Vilaprinyó, "Rare disease discovery: An optimized disease ranking system," *IEEE Trans. Ind. Inform.*, vol. 13, no. 3, pp. 1184–1192, Jun. 2017.

[6] F. Luo *et al.*,"Advanced pattern discovery-based fuzzy classification method for power system dynamic security assessment," *IEEE Trans. Ind. Inform.*, vol. 11, no. 2, pp. 416–426, Feb. 2015.

[7] I. Bose and X. Chen, "Hybrid models using unsupervised clustering for prediction of customer churn," *J. Org. Comput. Electron. Commerce*, vol. 19, no. 2, pp. 133–151, Apr. 2009.

[8] Y. Huang and T. Kechadi, "An effective hybrid learning system for telecommunication churn prediction," *Expert Syst. Appl.*, vol. 40, no. 14, pp. 5635–5647, Oct. 2013.

[9] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986.

[10] J. Quinlan, *C4. 5: Programs for Machine Learning*. New York, NY, USA: Elsevier, 2014.

[11] P. Clark and T. Niblett, "The CN2 induction algorithm," *Mach. Learn.*, vol. 3, no. 4, pp. 261–283, Mar. 1989.

[12] J. R. Quinlan and R. M. Cameron-Jones, "FOIL: A midterm report," in *Proc. Eur. Conf. Mach. Learn.*, Apr. 1993, pp. 1–20.

[13] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Trans. Neural Netw. Learn.*, vol. 29, no. 5, pp. 1774–1785, May 2018.

[14] S. Cost and S. Salzberg, "A weighted nearest neighbor algorithm for learning with symbolic features," *Mach. Learn.*, vol. 10, no. 1, pp. 57–78, Jan. 1993.

[15] L. Breiman, *Classification and Regression Trees*. Evanston, IL, USA: Routledge, 2017.

[16] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul./Aug. 1998.

[17] S. S. Y. Ng, Y. Xing, and K. L. Tsui, "A naive Bayes model for robust remaining useful life prediction of lithium-ion battery," *Appl. Energy*, vol. 118, no. 4, pp. 114–123, Apr. 2014.

[18] P. Peduzzi, J. Concato, E. Kemper, T. R. Holford, and A. R. Feinstein, "A simulation study of the number of events per variable in logistic regression analysis," *J. Clin. Epidemiol.*, vol. 49, no. 12, pp. 1373–1379, Dec. 1996.

[19] Z. H. Zhou and J. Feng, "Deep forest: Towards an alternative to deep neural networks," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3553–3559.

[20] S. R. Gaddam, V. V. Phoha, and K. S. Balagani, "K-Means+ID3: A novel method for supervised anomaly detection by cascading K-Means clustering and ID3 decision tree learning methods," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 3, pp. 345–354, Mar. 2007.

[21] C. Kaewchinporn, N. Vongsuchoto, and A. Srisawat, "A combination of decision tree learning and clustering for data classification," in *Proc. 8th Int. Joint Conf. Comput. Sci. Softw. Eng. (JCSSE)*, May 2011, pp. 363–367.

[22] G. Wang, J. Hao, J. Ma, and L. Huang, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering," *Expert Syst. Appl.*, vol. 37, no. 9, pp. 6225–6232, Sep. 2010.

[23] R. Eslamloueyan, "Designing a hierarchical neural network based on fuzzy clustering for fault diagnosis of the Tennessee–Eastman process," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 1407–1415, Jan. 2011.

[24] T. Ma, F. Wang, J. Cheng, Y. Yu, and X. Chen, "A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks," *Sensors*, vol. 16, no. 10, pp. 1701–1724, Oct. 2016.

[25] R. Rajamohamed and J. Manokaran, "Improved credit card churn prediction based on rough clustering and supervised learning techniques," *Cluster Comput.*, vol. 21, no. 1, pp. 65–77, Mar. 2018.

[26] X. Xu, J. Li, M. Zhou, J. Xu, and J. Cao, "Accelerated two-stage particle swarm optimization for clustering not-well-separated data," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2018.2839618.

[27] P. S. Bradley and U. M. Fayyad, "Refining initial points for K-Means clustering," in *Proc. 15th Int. Conf. Mach. Learn.*, vol. 98, Jul. 1998, pp. 91–99.

[28] M. Ester, H. P. Kriegel, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, vol. 96, no. 34, Aug. 1996, pp. 226–231.

[29] X. Ai, W. Jian, Z. Cui, and V. S. Sheng, "Broaden the minority class space for decision tree induction using antigen-derived detectors," *Knowl.-Based Syst.*, vol. 137, no. 12, pp. 196–205, Dec. 2017.

[30] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert Syst. Appl.*, vol. 40, no. 1, pp. 200–210, Jan. 2013.

[31] J. Su and H. Zhang, "A fast decision tree learning algorithm," in *Proc. Nat. Conf. Artif. Intell.*, vol. 6, Jul. 2006, pp. 500–505.

[32] S. Ruggieri, "Efficient C4.5," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 2, pp. 438–444, Mar. 2002.

[33] A. Cherfi, K. Nouira, and A. Ferchichi, "Very fast C4.5 decision tree algorithm," *Appl. Artif. Intell.*, vol. 32, no. 2, pp. 119–137, Mar. 2018.

[34] A. Asuncion and D. Newman, *UCIMachine Learning Repository*, University of California Irvine, Irvine, CA, USA, 2007.

[35] R. Collobert and S. Bengio, "Links between perceptrons, MLPs and SVMs," in *Proc. 21st Int. Conf. Mach. Learn.*, Sep. 2004, p. 23.

[36] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, Apr. 2011, Art. no. 27.

[37] L. Wang, Z. Zhang, and H. Long, "Wind turbine gearbox failure identification with deep neural networks," *IEEE Trans. Ind. Inform.*, vol. 13, no. 3, pp. 1360–1368, Jun. 2017.

[38] U. Fayyad, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, 1993, pp. 1022–1027.

[39] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *Ann. Math. Statist.*, vol. 11, no. 1, pp. 86–92, 1940.

[40] R. L. Iman and J. M. Davenport, "Approximations of the critical region of the fbietkan statistic," *Commun. Statist.*, vol. 9, no. 6, pp. 571–595, Jul. 1979.

[41] P. Nemenyi, "Distribution-free multiple comparisons," *Biometrics*, vol. 18, no. 2, p. 263, 1962.

[42] J. A. Sez, M. Galar, J. Luengo, and F. Herrera, "Analyzing the presence of noise in multi-class problems: Alleviating its influence with the one-vs-one decomposition," *Knowl. Inf. Syst.*, vol. 38, no. 1, pp. 179–206, Jan. 2014.

[43] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artif. Intell. Rev.*, vol. 22, no. 3, pp. 177–210, Nov. 2004.

[44] J. Xiao, Y. Li, L. Xie, D. Liu, and J. Huang, "A hybrid model based on selective ensemble for energy consumption forecasting in China," *Energy*, vol. 159, no. 9, pp. 534–546, Sep. 2018.

[45] L. Mo, L. Xie, X. Jiang, G. Teng, and J. Xiao, "GMDH-based hybrid model for container throughput forecasting: Selective combination forecasting in nonlinear subseries," *Appl. Soft Comput.*, vol. 62, no. 1, pp. 478–490, Jan. 2018.

[46] D. Wu, X. Luo, G. Wang, M. Shang, Y. Yuan, and H. Yan, "A highly accurate framework for self-labeled semi-supervised classification in industrial applications," *IEEE Trans. Ind. Inform.*, vol. 14, no. 3, pp. 909–920, Mar. 2018.

[47] J. Xiao, H. Cao, X. Jiang, X. Gu, and L. Xie, "GMDH-based semi-supervised feature selection for customer classification," *Knowl.-Based Syst.*, vol. 132, no. 9, pp. 236–248, Sep. 2017.

Jin Xiao (M'19) received the Ph.D. degree in management from the School of Business, Sichuan University, Chengdu, China, in 2010.

He is currently a Professor with the Department of Management Science, School of Business, Sichuan University, and a Postdoctoral Research Fellow with the Department of System Science, Chinese Academy of Sciences, Beijing, China. He is an Outstanding Contribution Expert of Sichuan Province, Tianfu Ten-Thousand Talents Program of Sichuan Province, and the Candidate of Academic and Technological Leaders of Sichuan Province, China. He has authored and coauthored more than 70 papers in leading journals, and serves as the anonymous reviewer of more than 50 journals. His research interests include business intelligence, data mining, customer relationship management, and economic forecasting.

Yuhang Tian (S'19) is currently working toward the master's degree in management science with the School of Business, Sichuan University, Chengdu, China.

His research interests include machine learning and fraud detection.

Ling Xie received the Ph.D. degree in management from the School of Business, Sichuan University, Chengdu, China, in 2019.

She is is currently a Lecturer in the Medical Big Data Center, School of Medical Information Engineering, Zunyi Medical University, Zunyi, China. Her research interests include business intelligence and data mining.

Xiaoyi Jiang (SM'09) received the bachelor's degree in computer science from Peking University, Beijing, China, and the Ph.D. and Venia Docendi (Habilitation) degrees in computer science from the University of Bern, Bern, Switzerland, in 1989 and 1997, respectively.

He was an Associate Professor with the Technical University of Berlin, Berlin, Germany. Since 2002, he has been a Full Professor of Computer Science with the University of Münster, Münster, Germany, where he is currently the Dean with the Faculty of Mathematics and Computer Science.

Dr. Jiang is a Fellow of the International Association for Pattern Recognition. He is currently the Editor-in-Chief for the *International Journal of Pattern Recognition and Artificial Intelligence*. In addition, he also serves on the Advisory Board and Editorial Board of several journals, including IEEE TRANSACTIONS ON MEDICAL IMAGING and *International Journal of Neural Systems*.

Jing Huang received the Ph.D. degree in economics from the School of Economics, Sichuan University, Chengdu, China, in 2014.

She is currently an Associate Professor with the Department of Public Utilities Management and Public Policy, School of Public Administration, Sichuan University, Chengdu, China. Her main research interests include public policy, government surpervision, as well as others.