IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 30, 2019, accepted December 28, 2019, date of publication January 8, 2020, date of current version January 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964849

Ordering-Based Kalman Filter Selective Ensemble for Classification

KAI YU[1], LIHONG WANG[1], AND YANWEI YU[2], (Member, IEEE)
[1]School of Computer and Control Engineering, Yantai University, Shandong 264005, China
[2]Department of Computer Science and Technology, Ocean University of China, Shandong 266100, China

Corresponding author: Yanwei Yu (yuyanwei@ouc.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61773331 and Grant 71672166.

ABSTRACT This paper investigates Kalman Filter-based Heuristic Ensemble (KFHE), which is a new perspective on multi-class ensemble classification with performance significantly better or at least as good as the state-of-the-art algorithms. We prove that the sample weight tuning method used in KFHE is a version of adaptive boosting, and the weight distribution does not change anymore and leads to redundant classifiers when the algorithm iterates enough times. This motivates us to select a sub-ensemble to alleviate the redundancy and improve the performance of the ensemble. An Ordering-based Kalman Filter Selective Ensemble (OKFSE) is proposed in this paper to select a sub-ensemble using the margin distance minimization approach. We demonstrate the effectiveness and robustness of OKFSE through extensive experiments on 20 real-world UCI datasets, and the statistical test shows that OKFSE significantly outperforms the state-of-the-art KFHE and clustering-based pruning methods on these datasets with 5% and 10% class label noise.

INDEX TERMS Machine learning, classification, selective ensemble, ordering-based pruning, Kalman filter.

I. INTRODUCTION

Ensemble is one of the most promising areas of research in machine learning and data mining. Multiple classifiers in an ensemble could be combined to achieve better performance than any individual classifier [1]. The learnability of strong learners and weak learners are equivalent, and weak learners can be boosted to be strong learners [2]. In an ensemble, if all the classifiers make the same predictions, they also make the same errors [3]. Thus, an ensemble requires its members accurate and diverse.

Diversified ensemble components can be obtained by using different training datasets, e.g., Boosting [4], [5], Bagging [6], and Random Forest [7]. Recently, a novel approach named Kalman Filter-based Heuristic Ensemble (KFHE) was proposed in [8]. KFHE used a new sample reweighting method to generate different training datasets for a diversified ensemble. Moreover, a combing rule based on Kalman filter was employed to predict the labels for new data, which is different from majoring voting in Bagging and weighted voting in AdaBoost. Kalman filter is a well-known methodology for linear Gaussian state estimation in dynamical systems [9]. The ensemble Kalman filter (EnKF) can be viewed as an approximate version of the standard Kalman filter, in which the Kalman gain is estimated based on the ensemble [10]. The ensemble in EnKF is a sample drawn from the filter distribution and then propagated forward through time and updated when new data become available [11]. Note that KFHE is different from EnKF, especially in the definition of ensemble. In KFHE, the ensemble components are classifiers. At time t, a classifier is generated on the reweighted training dataset, and then the label of new sample is predicted by combing the predictions of all classifiers using Kalman filter rules [8].

Although KFHE is robust to class-label noise and possess performance significantly better or at least as good as the state-of-the-art algorithms [8], there remains room for further improvement. We proved in this paper that, KFHE has redundant classifiers due to similar sample weights, and a sub-ensemble carefully selected is expected to have better performance than the entire ensemble. In this paper, we selected the sub-ensemble using the ordering-based algorithm margin distance minimization (MDM) to improve the performance of the ensemble generated by KFHE.

The main contributions are made in this paper as follows:

- The theoretical analysis of KFHE is presented, which shows the potential redundancy of KFHE and motivates us to prune the ensemble.
- Ordering-based Kalman Filter Selective Ensemble (OKFSE) is proposed. OKFSE has the advantages of

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

KFHE and MDM, i.e., OKFSE generates a robust ensemble of classifiers by KFHE and then effectively prunes the ensemble by MDM to improve the performance further.
- Extensive experiments are conducted on 20 real-world UCI datasets. Experiments show that OKFSE significantly outperforms KFHE and the compared clustering-based pruning algorithms on datasets with 5% and 10% class label noise.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 presents the proposed Ordering-based Kalman Filter Selective Ensemble. Experiments and result analysis are introduced in section 4. Finally, section 5 concludes this paper.

II. RELATED WORK

A. ORDERING-BASED SELECTIVE ENSEMBLE

An effective ensemble of classifiers requires its members accurate and diverse. A diversified ensemble can be obtained by using different training datasets, e.g., Boosting [4], [5], Bagging [6], and Random Forest [7]. A Boosting algorithm, e.g., AdaBoost [5], trains classifiers iteratively by focusing on the data which were misclassified by the previous classifiers, and then a biased classifier is obtained if the target class labels for fitting were corrupted by noise. The output of AdaBoost is produced by combining each weak classifier's decision using weighted majority voting. Therefore, AdaBoost is sensitive to noisy class labels and performs poorly as the level of noise increases [12]. Bagging-based algorithms sample the training data repeatedly in each iteration, and then combine all classifiers generated in iterations using majority voting to make the final decisions. Random Forest [7] can be seen as a variant of Bagging, since its components are trained by randomly sampling the dataset and its features simultaneously. Studies have shown that the classifiers in an ensemble generated by Bagging have redundancy between each other, and a sub-ensemble may outperform the entire one. For example, 20-40% of the ensemble generated by Bagging has the same or better performance than the entire ensemble [13], and 25% of the ensemble is competitive with a state-of-the-art technique for pruning Bagging ensemble in Meta-learning method [14]. One of the reasons of redundancy is related to the sampling weights, because Bagging samples the training dataset with the same sampling weights for each classifier, and the distance between two sampling probability distributions is zero in the view of Kullback-Leibler divergence [15]. Thus, the classifiers generated on similar samples will have similar structures or predictive behaviors, which lead to redundancy among classifiers.

Various methods were proposed to select the sub-ensemble and decrease the number of classifiers without worsening the performance of the ensemble [16]. There are three categories of selective ensemble or ensemble pruning techniques [17]:

(1) Optimization-based: ensemble pruning can be viewed as a combinatorial optimization problem aiming at finding a sub-ensemble that optimizes a predefined criterion. Genetic algorithms [18], [19] have been proposed to solve the problem approximately. However, these heuristic algorithms still suffer from low scalability due to the difficulty of global optimization [17].

(2) Clustering-based: classifiers with similar predictive behaviors [20] or structures [21], [22] are clustered together, and then a representative classifier is selected from each cluster to compose a sub-ensemble. As shown above, diversity and accuracy are two key factors to successful ensemble selection. Existing ensemble pruning methods consider diversity and accuracy separately or simultaneously to prune an ensemble [23]–[26]. Specially, diversity between classifiers is an important measure for clustering-based pruning algorithms. However, the effectiveness of existing diversity measures is discouraging since there seems to be no clear relation between those diversity measurements and the ensemble performance [27].

(3) Ordering-based: classifiers in the ensemble are ordered based on some predefined evaluation measures, and then members of the sub-ensemble are selected according to this order. Examples include MDM [13], [20] and orientation ordering [28]. Margin is another important measure in ensemble selection [29]–[31], and the algorithm combing margin and diversity is effective in ensemble learning [29]. As a margin ordering algorithm, MDM has been widely used for ensemble pruning for its impressing performance [13], [32]–[34]. A variant of MDM, named MDM-Imb, can handle the skewed-class distribution in imbalanced datasets [32], [33]. Zhu et al. proposed an improved discrete artificial fish swarm algorithm for ensemble pruning, which used a combination of diversity measure and MDM to find the tradeoff between diversity and accuracy of classifiers [34]. MDM is briefly summarized as follows:

MDM uses the distance among the output vectors of an ensemble to prune. The output vectors of the ensemble have the length equal to the size of training dataset. Let c_k be a classifier in the ensemble composed of T different classifiers. If the ith example is misclassified, the value of the vector of c_k at ith position is set to -1 and otherwise is 1, which is equal to the example margin. The sum of the vectors of selected classifiers is referred to as the vector of the sub-ensemble. The classifiers are added into the sub-ensemble C in order to make the corresponding vector $\langle \mathbf{c} \rangle$ be as close as possible to a reference position $\mathbf{o}$ placed in the first quadrant [20], [32]. Specially, $\mathbf{o} = (p, p, \ldots, p)$, where $p = 0.075$ as suggested [20]. The uth selected classifier is the one that minimizes

$$s_u = \arg\min_k d(\mathbf{o}, \frac{1}{T}(c_k + \sum_{t=1}^{u-1} c_{s_t})) \tag{1}$$

where s_t and s_u are the sub-ensemble with t and u classifiers respectively, k runs through the classifiers outside C and $d(x, y)$ is the Euclidean distance between x and y.

MDM is a highly efficient algorithm which outperforms most state-of-the-art pruning algorithms [13], thus we select MDM to prune the ensemble.

B. KFHE

KFHE [8] is the state-of-the-art multi-class ensemble classification method. It obtains an ensemble of classifiers, each of which regards the labels of the training dataset as the ideal state to be modeled. The individual component classifier c_t trained by KFHE can be viewed as an attempt to measure the ideal state with a related uncertainty indicated by the training error of c_t. KFHE uses Kalman filter to estimate the ideal state by combining these multiple noisy measurements. After all component classifiers are trained on the training dataset, the ensemble of these classifiers is used to predict the labels of the test dataset.

At the tth iteration, the classifier c_t is trained using a weighted sample of the training dataset and a Kalman filter is used to combine the prediction of c_t and *a priori* estimation to get *a posteriori* estimation.

Since the label vector of the training set is a static state, the time update equations are omitted in practice. Let $\mathbf{y}_t = [\mathbf{y}_{t1}; \mathbf{y}_{t2}; \ldots; \mathbf{y}_{tn}]$, with $\mathbf{y}_{ti}$ denoting the prediction for the ith data point x_i, and training dataset be $D = \{x_1, x_2, \ldots, x_n\}$.

The measurement $\mathbf{z}_t$ is taken as the average of the previous prediction, $\hat{\mathbf{y}}_{t-1}$, and the prediction of c_t, as in Eq.(2). The measurement step and its related error are as follows:

$$\mathbf{z}_t = \frac{1}{2}(\hat{\mathbf{y}}_{t-1} + c_t(D)), \tag{2}$$

$$R_t = \frac{1}{n}\sum_{i=1}^{n}(Y_i \neq class(\mathbf{z}_{ti})), \tag{3}$$

$$\hat{\mathbf{y}}_t = \hat{\mathbf{y}}_{t-1} + K_t(\mathbf{z}_t - \hat{\mathbf{y}}_{t-1}), \tag{4}$$

$$K_t = P_{t-1}(P_{t-1} + R_t)^{-1}, \tag{5}$$

$$P_t = (1 - K_t)P_{t-1}, \tag{6}$$

where $\mathbf{z}_t = [\mathbf{z}_{t1}; \mathbf{z}_{t2}; \ldots; \mathbf{z}_{tn}]$ represents the measurement; $c_t(D)$ indicates the predictions made by c_t for the dataset D; R_t is the misclassification rate and taken as the uncertainty related to $\mathbf{z}_t$; $class(\mathbf{z}_{ti})$ is the class prediction made by the ensemble for x_i, whose ground truth class is Y_i; P_t is the uncertainty related to $\hat{\mathbf{y}}_t$, and K_t is the Kalman gain. P_t and K_t are scalars in the KFHE implementation.

KFHE used another Kalman filter to decide the weights for the next sampling of the training dataset. Specially, $w_{t+1}(\mathbf{x}_i)$, the weight estimation of $\mathbf{x}_i$, is updated as Eq.(7) for KFHE_e [8]:

$$w_{t+1}(\mathbf{x}_i) = w_t(\mathbf{x}_i)(1 + K_t(\exp(\frac{1}{n} + (class(\mathbf{Z}_{ti}) \neq Y_i)) - 1)). \tag{7}$$

KFHE initializes the Kalman filters and generates an initial classifier on the sample of training set, and then updates the measurements and related uncertainties using Eqs.(2)-(6), and updates the sampling weights using Eq.(7), finally resamples the training dataset for a new classifier in the next iteration, see [8] for details. When 100 classifiers (decision trees) are generated, KFHE stops with Kalman gains in each iteration recorded for prediction of test samples. KFHE has been validated with classification tasks, so it is challenging to improve its performance further.

III. PROPOSED ORDERING-BASED KALMAN FILTER SELECTIVE ENSEMBLE (OKFSE)

A. OKFSE

We are interested in the potential application of MDM for KFHE pruning. Combining MDM with KFHE, we propose an algorithm called Ordering-based Kalman Filter Selective Ensemble (OKFSE) to prune the ensemble. OKFSE is composed of three stages, as shown in Figure 1, where the red line is our work and the others are compared algorithms, including KFHE itself. The first two stages are training and pruning, as shown in Algorithm 1.

In the first stage (lines 1-8 in Algorithm 1), KFHE is used to generate the pool of candidate classifiers. We stored the Kalman gains for each classifier. In the second stage (lines 9-10 in Algorithm 1), we used MDM to select the classifiers to constitute the sub-ensemble C. As described previously, MDM greedily selects the classifiers one by one to reduce the distance between vector of the sub-ensemble and the reference position.

Algorithm 1 OKFSE_Training_Pruning

Input: the training dataset D, the ensemble size T, the sub-ensemble size M
Output: sub-ensemble C, Kalman gains $\{K_t\}$

1: set equal sampling weights for all examples in D
2: train the classifier c_0 using D
3: $t = 1, C = \emptyset$
4: **while** $t \leq T$
5: use Eqs. (2)-(6) to train the classifier c_t, calculate K_t and P_t
6: use Eq. (7) to update the weights of D for training the next classifier
7: $t = t + 1$
8: **end while** // The original ensemble has been generated now
9: use MDM to select the classifiers according to Eq.(1) and add into C, until $|C| = M$
10: add the classifier c_0 into C
11: **return:** C, $\{K_t\}$

In the third stage, the outputs of the component classifiers are combined into the final decision for the test sample. Note that, Kalman filter is a sequential ensemble and the individual classifier must be sorted in the order of the generation sequence. The initial classifier c_0 is used to make the prediction of $\hat{\mathbf{y}}_0$. Eqs. (2) and (4) are used to combine the output from each selected classifier recursively. Algorithm 2 summarizes the process of OKFSE_test.

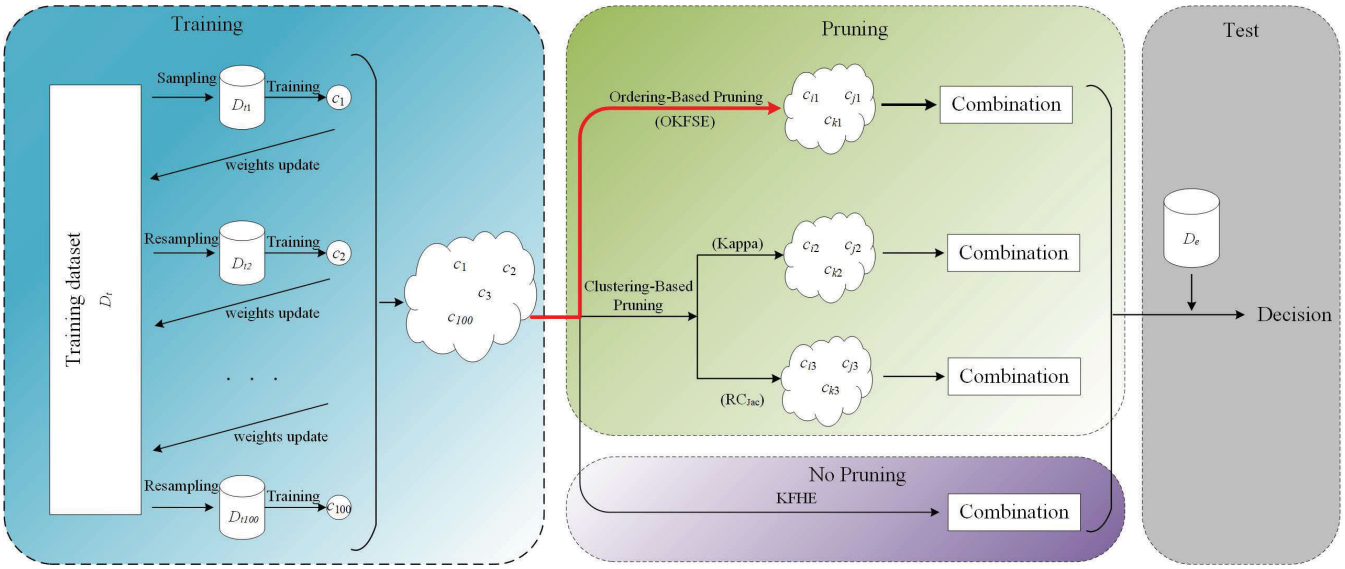


FIGURE 1. The diagram of OKFSE.

Algorithm 2 OKFSE_Test

Input: an instance $\mathbf{d}$, the sub-ensemble C, Kalman gains $\{K_t\}$
Output: $\mathbf{y_d}$, the prediction for $\mathbf{d}$
1: $\mathbf{y_d} = c_0(\mathbf{d})$
2: $t = 1$
3: **while** $t \leq T$
4: **if** $c_t \in C$
5: update $\mathbf{y_d}$ using Eqs.(2) and (4)
6: **end if**
7: **end while**
8: **return** $\mathbf{y_d}$

B. THEORETICAL BASIS OF OKFSE

We provide the theoretical basis for OKFSE in this section, which shows the potential redundancy of KFHE and motivates us to prune. We prove some theorems for KFHE with proofs followed.

Theorem 1: $\lim_{t\to\infty} K_t = 0$, $\lim_{t\to\infty} P_t = 0$, *where* K_t *and* P_t *are calculated by Eqs.(5) and (6) respectively.*

Proof: Since R_t is a misclassification rate, then $0 \leq R_t \leq 1$. P_{t-1} is the covariance representing the uncertainty of $\hat{\mathbf{y}}_{t-1}$ and is initialized to 1, thus $0 \leq P_{t-1} \leq 1$, and then $0 \leq K_t \leq 1$ from Eq. (5).

Hence $P_t = (1 - K_t)P_{t-1} \leq P_{t-1}$.

Therefore, $\{P_t, t = 1, 2, \ldots\}$ is a monotonically nonincreasing sequence with limited lower bound, then P_t has a limitation.

We assume that $\lim_{t\to\infty} P_t = \beta$, then $\beta \geq 0$.

First, we prove $\lim_{t\to\infty} P_t = 0$.

If β>0, let $t \to \infty$ in $P_t = (1 - K_t)P_{t-1}$, thus, $\lim_{t\to\infty} K_t$ exists.

Let $\lim_{t\to\infty} K_t = \alpha$, thus, $\beta = (1 - \alpha)\beta$, and then $\alpha = 0$.

Substituting $\alpha = 0$ into $\alpha = \lim_{t\to\infty} K_t = \lim_{t\to\infty} P_{t-1}(P_{t-1} + R_t)^{-1}$, then $\beta = 0$.

It is conflicted with the hypothesis $\beta > 0$. Hence, $\beta = 0$, i.e., $\lim_{t\to\infty} P_t = 0$.

Next, we prove that $\lim_{t\to\infty} K_t = 0$.

(1) If $R_t > 0$ holds for any t, i.e., not all the training instances are classified correctly, then $R_t \geq 1/n$ according to its definition and $K_t = P_{t-1}(P_{t-1} + R_t)^{-1} \to 0$.

(2) If $R_{t0} = 0$ holds for some t_0, i.e., all the training instances are classified correctly by the ensemble, then $K_{t0} = P_{t0-1}(P_{t0-1} + R_{t0})^{-1} = 1$.
Hence, $P_{t0} = (1 - K_{t0})P_{t0-1} = 0$ and $K_{t0+1} = P_{t0}(P_{t0} + R_{t0+1})^{-1} = 0$. To make sure the denominator is not equal to 0 in this special case, a tiny positive is added into the denominator in [8]. From now on, P_t and K_t will keep their values as 0.

In summary, $\lim_{t\to\infty} K_t = 0$, and $\lim_{t\to\infty} P_t = 0$. □

Theorem 1 proves that, both P_t and K_t will converge to 0, no matter what sampling method is used to generate the training dataset for the next iteration.

Theorem 2: In the t*th iteration of KFHE, if* x_i *is misclassified by the ensemble (i.e.,* $class(\mathbf{z}_{ti}) \neq Y_i$*), then the weight of* x_i *will increase, i.e.,* $w_{t+1}(x_i) \geq w_t(x_i)$ *in the next resampling for the* $(t + 1)$*th classifier training. Hence the KFHE is an adaptive boosting algorithm.*

Proof: According to Eq. (7),

$$w_{t+1}(\mathbf{x}_i) = w_t(\mathbf{x}_i)(1 + K_t(\exp(\frac{1}{n} + (class(\mathbf{Z}_{ti}) \neq Y_i)) - 1))$$

where, $(class(\mathbf{z}_{ti}) \neq Y_i)$ is an indicator with value 1 if $(class(\mathbf{z}_{ti}) \neq Y_i)$ is true, and 0, otherwise.

(1) If $class(\mathbf{z}_{ti}) \neq Y_i$, i.e., the instance $\mathbf{x}_i$ is misclassified by the ensemble with t classifiers, then

$$w_{t+1}(\mathbf{x}_i) = w_t(\mathbf{x}_i)(1 + K_t(\exp(\frac{1}{n} + 1) - 1)). \quad (8)$$

(2) If $class(\mathbf{z}_{ti}) = Y_i$, then

$$w_{t+1}(\mathbf{x}_i) = w_t(\mathbf{x}_i)(1 + K_t(\exp(\frac{1}{n}) - 1)). \quad (9)$$

Let

$$Q_t = \sum_{class(\mathbf{z}_{ti}) \neq Y_i} w_{t+1}(\mathbf{x}_i) + \sum_{class(\mathbf{z}_{ti}) = Y_i} w_{t+1}(\mathbf{x}_i).$$

Substituting Eqs.(8) and (9) into Q_t, we can obtain

$$Q_t = D_{1t}(1+K_t(\exp(\frac{1}{n} + 1)-1))+D_{2t}(1+K_t(\exp(\frac{1}{n})-1))$$

where

$$D_{1t} = \sum_{class(\mathbf{z}_{ti}) \neq Y_i} w_t(\mathbf{x}_i),$$
$$D_{2t} = \sum_{class(\mathbf{z}_{ti}) = Y_i} w_t(\mathbf{x}_i),$$

and $D_{1t} + D_{2t} = 1$, due to the normalization of weights.

The normalized weight of an instance x_i which was misclassified becomes:

$$\begin{aligned} & w'_{t+1}(\mathbf{x}_i) \\ &= \frac{w_t(\mathbf{x}_i)(1 + K_t(\exp(\frac{1}{n} + 1) - 1))}{D_{1t}(1 + K_t(\exp(\frac{1}{n}+1) - 1))+D_{2t}(1+K_t(\exp(\frac{1}{n}) - 1))} \\ &= \frac{w_t(\mathbf{x}_i)}{D_{1t} + D_{2t}\frac{1+K_t(\exp(\frac{1}{n})-1)}{1+K_t(\exp(\frac{1}{n}+1)-1)}} \end{aligned} \quad (10)$$

Because $K_t \geq 0, \frac{1+K_t(\exp(\frac{1}{n})-1)}{1+K_t(\exp(\frac{1}{n}+1)-1)} \leq 1$, thus, $w'_{t+1}(\mathbf{x}_i) \geq w_t(\mathbf{x}_i)$. Here, $w'_{t+1}(\mathbf{x}_i) = w_t(\mathbf{x}_i)$ if and only if $K_t = 0$.

Similarly, the normalized weight of an instance x_i which was classified correctly becomes:

$$w'_{t+1}(\mathbf{x}_i) = \frac{w_t(\mathbf{x}_i)}{D_{1t}\frac{1+K_t(\exp(\frac{1}{n}+1)-1)}{1+K_t(\exp(\frac{1}{n})-1)} + D_{2t}} \leq w_t(\mathbf{x}_i)$$

□

Theorem 2 shows that, the $(t + 1)$th classifier will focus on the instances misclassified by the ensemble of t classifiers, so KFHE_e will generate diversified classifiers. The update of weights is a characteristic of the adaptive boosting algorithms [35]. Hence, KFHE is also an adaptive boosting algorithm.

KL-divergence (Kullback-Leibler divergence) can be used to calculate the distance between two probability distributions P and Q [15].

$$KL(P(D)||Q(D)) = \sum_{\mathbf{x}_i \in D} [P(\mathbf{x}_i) \log \frac{P(\mathbf{x})}{Q(\mathbf{x})}].$$

Theorem 3: When $t \to \infty$, $KL(w_t(D)||w_{t+1}(D)) \to 0$, where $w_t(D)$ and $w_{t+1}(D)$ denote the sampling weights of the training datasets in two successive iterations respectively.

Proof: We use $w_{t+1}(\mathbf{x}_i)$ to represent $w'_{t+1}(\mathbf{x}_i)$ for convenience.

The KL-divergence between the $w_t(D)$ and $w_{t+1}(D)$ becomes:

$$\begin{aligned} & KL(w_t(D)||w_{t+1}(D)) \\ &= \sum_{\mathbf{x}_i \in D} [w_t(\mathbf{x}_i) \log(\frac{w_t(\mathbf{x}_i)}{w_{t+1}(\mathbf{x}_i)})] \\ &= \sum_{class(\mathbf{z}_{ti}) \neq Y_i} [w_t(\mathbf{x}_i) \log (D_{1t} + D_{2t}\frac{1+K_t(\exp(\frac{1}{n})-1)}{1+K_t(\exp(\frac{1}{n}+1)-1)})] \\ &\quad + \sum_{class(\mathbf{z}_{ti}) = Y_i} [w_t(\mathbf{x}_i) \log (D_{1t}\frac{1+K_t(\exp(\frac{1}{n}+1)-1)}{1+K_t(\exp(\frac{1}{n})-1)} + D_{2t})] \\ &= D_{1t} \log (D_{1t} + D_{2t}\frac{1 + K_t(\exp(\frac{1}{n}) - 1)}{1 + K_t(\exp(\frac{1}{n} + 1) - 1)}) \\ &\quad + D_{2t} \log (D_{1t}\frac{1 + K_t(\exp(\frac{1}{n} + 1) - 1)}{1 + K_t(\exp(\frac{1}{n}) - 1)} + D_{2t}) \end{aligned}$$

where D_{1t} and D_{2t} are defined as above, and $D_{1t} + D_{2t} = 1$.

Theorem 1 shows that $\lim_{t\to\infty} K_t = 0$, therefore, if $t \to \infty$,

$$\frac{1 + K_t(\exp(\frac{1}{n} + 1) - 1)}{1 + K_t(\exp(\frac{1}{n}) - 1)} \to 1.$$

Thus, $KL(w_t(D)||w_{t+1}(D)) \to 0$. □

From Theorem 3, the resampling weights do not change significantly if the ensemble size is large enough. The new classifier is trained using a training dataset which is similar to the one used to train the previous classifiers. Thus, the generalization abilities of the classifiers are similar to each other and some of the classifiers are redundant.

IV. EXPERIMENTS AND DISCUSSIONS

A. DATASETS AND PERFORMANCE MEASURE

We carried out experiments on 20 real-world UCI datasets [36] with different number of instances, attributes and classes. Table 1 describes the details of these datasets.

Following [8], an extended macro-averaged $F_1^{(macro)}$ is used to evaluate the performance:

$$F_1^{(macro)} = \frac{1}{l}\sum_{i=1}^{l} 2 \times \frac{precision^{(i)} \times recall^{(i)}}{precision^{(i)} + recall^{(i)}},$$

where $precision^{(i)}$ and $recall^{(i)}$ are the precision and recall values for the ith class respectively, and l is the number of classes. $F_1^{(macro)}$ is an extended multi-class version of the macro-averaged F_1-score for binary classification, and is appropriate for potential class imbalance.

B. EXPERIMENTAL SETUP

We employed KFHE to generate an ensemble of decision trees for classification, and then tried to prune the ensemble to improve the performance. Clustering-based pruning algorithms were used as a benchmark in the experiments, which used Kappa [13], [37] and RC_{Jac} [21] as diversity measures

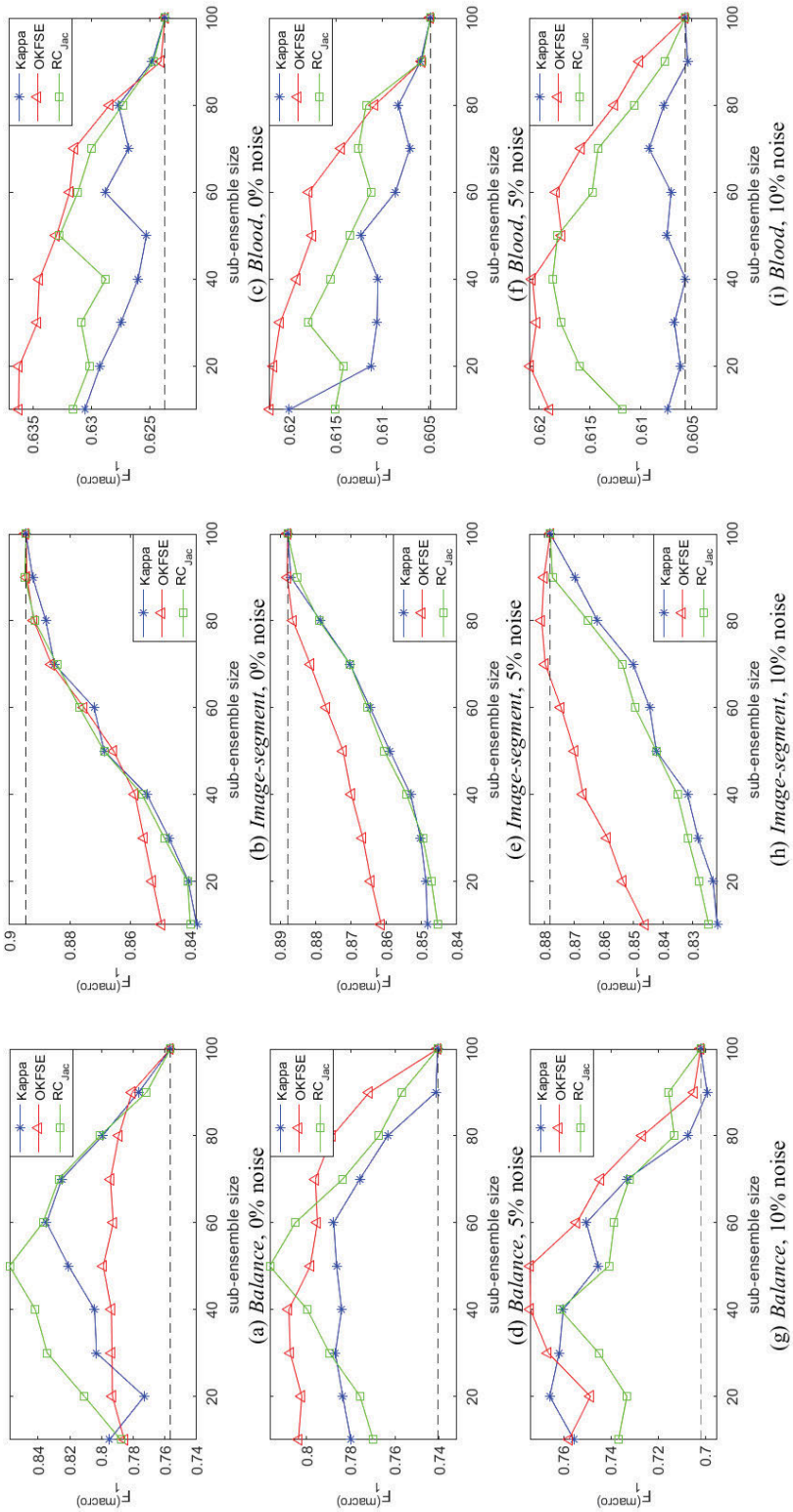


FIGURE 2. Performance changes of the algorithms, the dashed horizonal line is KFHE.

respectively. Diversity of decision trees can be classified into two categories, i.e., semantic and structural diversity. Kappa is a semantic diversity which is measured by the κ statistic [13], [37]. The lower κ, the higher diversity, thus we used $1 - \kappa$ to measure the distance between two trees in the experiments. $\mathrm{RC}_{\mathrm{Jac}}$ is a structural diversity measure

for decision trees, because it concerns the distributions of instances in the leaves and calculates the distance between these distributions [21]. In the experiments, we calculated the diversity of each pair of classifiers using Kappa or RC_{Jac} and created a similarity matrix, then called the hierarchical clustering algorithm to divide the classifiers into different clusters, finally, we selected one classifier from each cluster to construct a sub-ensemble. Figure 1 illustrates the pruning process.

To allow a fair comparison of the techniques, all evaluations use the same experimental protocol, i.e., the same division of datasets for training and test, as well as the same pool of classifiers. The component learners are decision trees and the size of pool is set to 100 as suggested in [8]. The sub-ensemble size is the number of clusters in the clustering-based ensemble pruning algorithm and also is the number of trees in MDM for ordered sub-ensemble generation. The parameter p in MDM was set to 0.075 as suggested in [13], [20]. The combination of sub-ensemble used Kalman Filter scheme to predict the labels of test samples. Similar to KFHE, for a pair of dataset and algorithm, a 20 times 4-fold cross-validation experiment was performed, and the mean of the $F_1^{(macro)}$-scores across the folds were computed. Furthermore, 5% and 10% class-label noise were introduced synthetically into each of the datasets in Table 1 respectively. Taking 5% as an example, we selected 5% instances of the dataset and replaced the class label of each instance with a random label other than its ground truth class. For each of these noisy datasets, a 20 times 4-fold cross-validation experiment was performed. For each fold, the noisy class labels were used in training, but the $F_1^{(macro)}$-scores were computed with respect to the dataset without label noise [8].

TABLE 1. The datasets used in this paper.

Dataset	# instances	# attributes	# classes
Balance	625	4	3
Blood	748	4	2
BreastTissue	106	9	6
Bupa	345	6	2
Ecoli	336	7	8
Ferbility	100	9	2
Glass	214	9	6
Haberman	306	3	2
Hayes-roth	132	5	3
Heart	270	13	2
Image-segment	210	19	7
Ionosphere	351	34	2
Iris	150	4	3
Pima Indians	768	8	2
Seed	210	7	3
Surgery	470	16	2
Tae	151	5	3
Wdbc	569	30	2
Wine	178	13	3
Zoo	101	16	7

C. EXPERIMENTAL RESULTS

For simplification, we denote the compared clustering-based algorithms as Kappa and RC_{Jac} in the experiments. We tested OKFSK, Kappa and RC_{Jac} by varying the sub-ensemble size from 10 to 90 with step length 10, and computed the $F_1^{(macro)}$-scores of each algorithm on each dataset for three levels of class label noise. Due to space limits, we showed only the best performances over all tested sub-ensemble sizes in Tables 5-7 in Appendix for all algorithm-dataset pairs.

1) W/T/L(WINS/TIES/LOSES)

From Tables 5-7, we observed that, from the view of W/T/L, OKFSE performs significantly better than other methods on datasets with 5% and 10% noise level. Specially, OKFSE wins on 9 and 10 datasets when the noise level is 5% and 10% respectively, and ties on 5 and 4 datasets at the noise level 5% and 10% respectively. In summary, OKFSE has higher wins and ties than the others with the noisy data. Hence, in practice, OKFSE can be more advantageous due to more or less class label noise included in the obtained raw datasets.

2) FRIEDMAN TEST AND FRIEDMAN ALIGNED TEST

We used Friedman rank test [38] for the statistical comparison of these techniques over the 20 datasets. Table 2 presents the average Friedman ranks summarized from Tables 5-7 in Appendix. Lower ranks are better, the best performing algorithm is the one presenting the lowest average rank. OKFSE is the algorithm with the lowest average ranks at noise levels 5% and 10% (average rank = 1.58 and 1.53 respectively). From the view of average rank, we observed that OKFSE is better than the others when the training datasets were corrupted by noise.

To compare the four algorithms (i.e., KFHE, OKFSE, Kappa and RC_{Jac}), we evaluated the following hypothesis H_0 using Friedman test.

Null hypothesis H_0:

The four algorithms do not show any significant difference when used for classification on the datasets.

We calculated p-value for each test, and the hypothesis was checked at $\alpha = 0.05$ significance level, as shown in Table 2. At three noise levels, Friedman test results are all significant at $\alpha = 0.05$. Thus, we rejected Null hypothesis H_0. Namely, the four algorithms perform significantly different from each other at the tested noise levels. We noticed that OKFSE is the best one when the noise levels are 5% and 10%, hence, OKFSE significantly outperforms the compared algorithms with noisy data.

Generally, Friedman rank test is recommended if k>5 and N>10, where k is the number of algorithms and N is the number of datasets. In our experiments, $k = 4$, therefore we further conducted Friedman aligned rank test to evaluate the above hypothesis H_0. Tables 5-7 in Appendix also show the

TABLE 2. Average ranks and p-values for different levels of class label noise. Best ranks are high-lighted in boldface.

	Noise level	KFHE	OKFSE	Kappa	RC_{Jac}	p-value
Rank	0%	2.83	2.53	2.85	**1.80**	0.0347
	5%	2.80	**1.58**	2.95	2.68	0.0027
	10%	3.10	**1.53**	3.13	2.25	9.7288e-05
Aligned Rank	0%	48.18	42.78	45.08	**25.98**	0.0373
	5%	52.45	**22.65**	49.40	37.50	0.0010
	10%	55.85	**23.33**	51.40	31.43	7.8068e-05

TABLE 3. Adjusted p-values for the Friedman test (OKFSE is the control method).

Noise level	Friedman	Unadjusted	Bonferroni	Holm	Finner
0%	RC_{Jac}	0.075753	0.227258	0.227258	0.210477
	Kappa	0.425983	1.0	0.851966	0.565102
	KFHE	0.462433	1.0	0.851966	0.565102
5%	Kappa	0.000757	0.002271	0.002271	0.002269
	KFHE	0.002694	0.008083	0.005389	0.004039
	RC_{Jac}	0.007051	0.021152	0.007051	0.007051
10%	Kappa	0.000089	0.000267	0.000267	0.000267
	KFHE	0.000114	0.000343	0.000229	0.000172
	RC_{Jac}	0.075753	0.227258	0.075753	0.075753

TABLE 4. Adjusted p-values for the Friedman aligned test (OKFSE is the control method).

Noise level	Friedman aligned	Unadjusted	Bonferroni	Holm	Finner
0%	RC_{Jac}	7.1227e-06	2.1368e-05	2.1368e-05	2.1368e-05
	KFHE	1.4896e-01	4.4688e-01	2.9792e-01	2.1490e-01
	Kappa	5.3875e-01	1.0	5.3875e-01	5.3875e-01
5%	KFHE	1.5543e-15	4.6629e-15	4.6629e-15	4.6629e-15
	Kappa	8.7264e-13	2.6179e-12	1.7453e-12	1.3090e-12
	RC_{Jac}	7.2227e-05	2.1668e-04	7.2227e-05	7.2227e-05
10%	KFHE	0.0	0.0	0.0	0.0
	Kappa	6.2173e-14	1.8653e-13	1.2435e-13	9.3259e-14
	RC_{Jac}	3.0402e-02	9.1205e-02	3.0402e-02	3.0402e-02

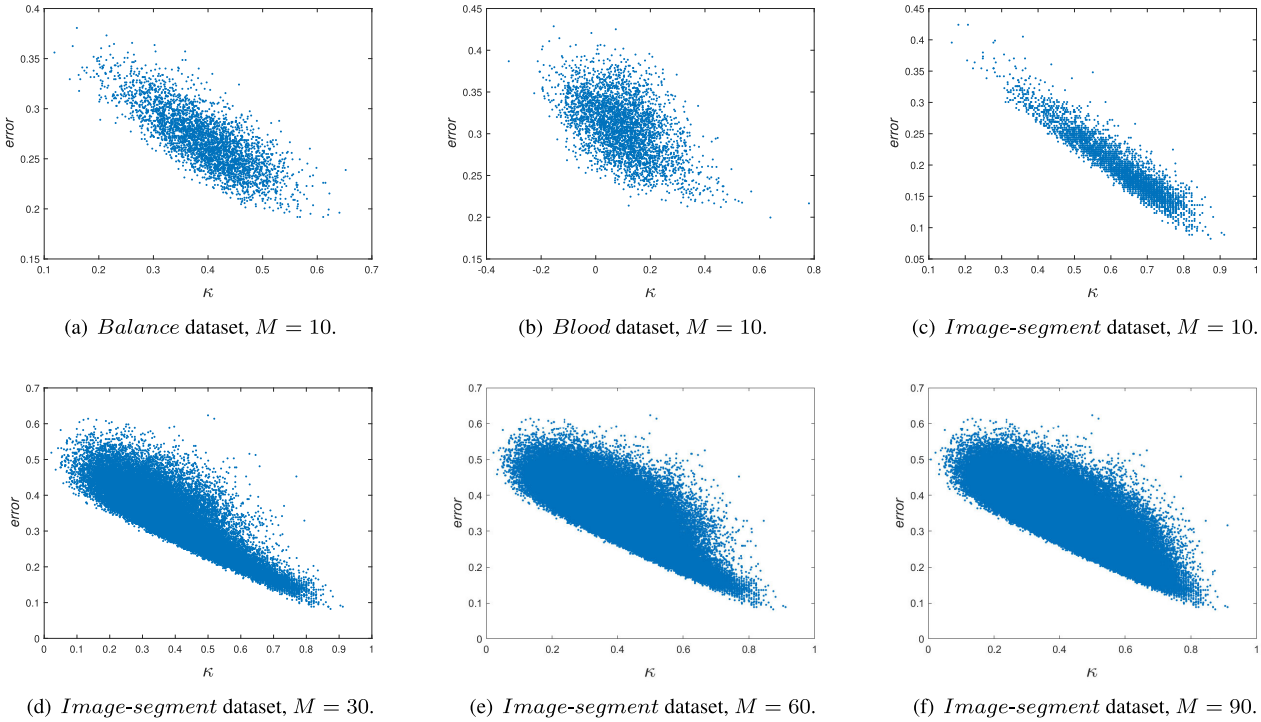
Friedman aligned rank for each combination of algorithms and datasets. Additionally, Table 2 presents the average Friedman aligned ranks summarized from Tables 5-7. As expected, OKFSE is the algorithm with the lowest average ranks at noise levels 5% and 10% (average aligned rank = 22.65 and 23.33 respectively). The average aligned ranks confirm that OKFSE is better than the others when the training datasets were corrupted by noise.

We also calculated p-value for each Friedman aligned test, and the hypothesis was checked at $\alpha = 0.05$ significance level, as shown in Table 2. At three noise levels, Friedman aligned test results are all significant at $\alpha = 0.05$. Thus, we reject Null hypothesis H_0 based on Friedman aligned test.

3) POST-HOC TESTS

Both Friedman test and Friedman Aligned test detected significant differences between the four algorithms, then post-hoc tests are applied to compare OKFSE (the control method) with other algorithms. The p-values can be obtained using the ranks computed by the Friedman and Friedman Aligned tests, respectively. Tables 3 and 4 show the unadjusted p-values and p-values adjusted by Bonferroni-Dunn, Holm and Finner procedures.

In Table 3, at noise level 5%, the Friedman test shows a significant improvement of OKFSE over KFHE, Kappa,and RC_{Jac} for all the post-hoc procedures considered, using a significance level of 0.05. However, At noise level 10%,



(a) *Balance* dataset, $M = 10$. (b) *Blood* dataset, $M = 10$. (c) *Image-segment* dataset, $M = 10$.

(d) *Image-segment* dataset, $M = 30$. (e) *Image-segment* dataset, $M = 60$. (f) *Image-segment* dataset, $M = 90$.

FIGURE 3. Kappa-error diagrams for OKFSE.

the Friedman test fails to show a significant improvement of OKFSE over RC_{Jac} for all the post-hoc procedures considered.

In Table 4, at noise levels 5% and 10%, the Friedman aligned test shows a significant improvement of OKFSE over KFHE, Kappa,and RC_{Jac} for every post-hoc procedure considered, except Bonferroni at noise level 10%, which fails to highlight the difference between OKFSE and RC_{Jac} as significant. Since Friedman aligned test is desirable in our experiments due to the small number of algorithms, again, we confirmed that OKFSE is better than the others when the training datasets were corrupted by noise.

4) EVIDENCE OF OVERFITTING

We summarized three typical trends of $F_1^{(macro)}$-scores over the sub-ensemble size. Figure 2 shows these trends for each algorithm on *Balance*, *Blood* and *Image-segment* datasets respectively. The first, second and third rows of Figure 2 are the performances on the datasets with class-label noise levels 0%, 5%,10% respectively. The dashed horizonal line in each subplot shows the performance of KFHE on the corresponding dataset. From the results, we observed:

- *Balance* dataset reaches its highest performance when the sub-ensemble size is round 50. Increasing the tree number increases $F_1^{(macro)}$ when the number is less than 50, but decreases it when the number is larger than 50. Three pruning algorithms support the similar observations, and the advantages of OKFSE over the other two compared clustering-based algorithms become obvious when the noise level increases. In addition, the performances of three pruning algorithms are all better than KFHE. Datasets with similar trend also include *BreastTissue*, *Ecoli*, *Glass*, *Iris* and *Seed*.
- On *Image-segment*, increasing the tree number increases $F_1^{(macro)}$ monotonically, and KFHE achieves the best performance. We will discuss the reason of pruning failure later. Datasets with similar trend also include *Bupa*, *Hayes-roth*, *Ionosphere*, *Tae* and *Wdbc*.
- *Blood* obtains its best performance with very small sub-ensemble size, and decreases the performance monotonically as the tree number increases, which demonstrates overfitting of the ensemble. Moreover, OKFSE is better than the others at all noise levels, and the performances of three pruning algorithms are all better than KFHE. Datasets with similar trend also include *Ferbility*, *Haberman* and *Surgery*.

In summary, on more than half of the tested datasets, performances reach the peaks when the sub-ensemble sizes reach some thresholds, which are closely related to the datasets, and then decrease as the tree number increases. The decreasing of generalization performance shows overfitting of the ensemble with larger size.

5) REDUNDANCY VS. DIVERSITY OF A SUB-ENSEMBLE

Previous studies as well as our experiments have shown that the classifiers in an ensemble have redundancy between each other, and diversity is a key factor for measuring redundancy. Generally, the more diversified, the less redundancy. We used

TABLE 5. $F_1^{(macro)}$-scores for each algorithm performed on each dataset without noise. The values in parenthesis are the Friedman ranking (FR) and the Friedman aligned ranking (FAR),respectively.

Data sets	KFHE	OKFSE	Kappa	RC_{Jac}
Balance	0.757 ± 0.13 (4, 79)	0.799 ± 0.12 (3, 77)	0.835 ± 0.11 (2, 5)	**0.858** ± 0.09 (1, 1)
Blood	0.624 ± 0.03 (4, 74)	**0.636** ± 0.03 (1, 8)	0.631 ± 0.03 (3, 40.5)	0.633 ± 0.03 (2, 17)
BreastTissue	0.688 ± 0.07 (3, 48)	0.691 ± 0.07 (2, 15)	0.682 ± 0.07 (4, 73)	**0.693** ± 0.08 (1, 9)
Bupa	0.700 ± 0.05 (4, 51)	**0.701** ± 0.05 (2, 37)	**0.701** ± 0.05 (2, 37)	**0.701** ± 0.05 (2, 37)
Ecoli	0.805 ± 0.05 (4, 70)	**0.811** ± 0.05 (1.5, 13.5)	0.806 ± 0.05 (3, 65.5)	**0.811** ± 0.05 (1.5, 13.5)
Ferbility	0.843 ± 0.12 (4, 78)	0.877 ± 0.10 (3, 75)	**0.925** ± 0.05 (1, 2)	0.893 ± 0.08 (2, 7)
Glass	0.743 ± 0.08 (3.5, 62)	**0.748** ± 0.08 (1, 12)	0.743 ± 0.08 (3.5, 62)	0.746 ± 0.08 (2, 26)
Haberman	0.563 ± 0.05 (4, 76)	0.569 ± 0.06 (3, 59.5)	0.570 ± 0.06 (2, 51)	**0.581** ± 0.06 (1, 6)
Hayes-roth	**0.828** ± 0.05 (1, 16)	0.825 ± 0.06 (3, 51)	0.825 ± 0.06 (3, 51)	0.825 ± 0.06 (3, 51)
Heart	**0.808** ± 0.04 (1, 19)	0.806 ± 0.04 (3, 44)	0.804 ± 0.04 (4, 65.5)	0.807 ± 0.04 (2, 31.5)
Image-segment	**0.895** ± 0.04 (1.5, 26)	0.894 ± 0.04 (3, 40.5)	0.892 ± 0.04 (4, 62)	**0.895** ± 0.04 (1.5, 26)
Ionosphere	**0.926** ± 0.03 (1, 18)	0.923 ± 0.03 (3, 56)	0.922 ± 0.03 (4, 64)	0.925 ± 0.03 (2, 29)
Iris	0.947 ± 0.04 (3.5, 54.5)	0.947 ± 0.04 (3.5, 54.5)	**0.949** ± 0.04 (1.5, 26)	**0.949** ± 0.04 (1.5, 26)
PimaIndians	0.716 ± 0.03 (2, 34)	0.715 ± 0.03 (3, 47)	0.713 ± 0.03 (4, 67)	**0.717** ± 0.03 (1, 20)
Seed	0.928 ± 0.03 (3, 44)	**0.929** ± 0.03 (1.5, 31.5)	0.927 ± 0.03 (4, 57)	**0.929** ± 0.03 (1.5, 31.5)
Surgery	0.624 ± 0.17 (4, 80)	0.711 ± 0.20 (2, 4)	**0.721** ± 0.20 (1, 3)	0.677 ± 0.19 (3, 72)
Tae	**0.567** ± 0.08 (1, 10)	0.561 ± 0.08 (3, 59.5)	0.560 ± 0.08 (4, 69)	0.563 ± 0.08 (2, 39)
Wdbc	0.957 ± 0.02 (3, 44)	**0.958** ± 0.02 (1, 31.5)	0.957 ± 0.02 (3, 44)	0.957 ± 0.02 (3, 44)
Wine	0.951 ± 0.04 (3, 58)	0.950 ± 0.04 (4, 68)	0.953 ± 0.04 (2, 35)	**0.956** ± 0.03 (1, 11)
Zoo	**0.928** ± 0.07 (2, 22)	0.923 ± 0.07 (4, 71)	**0.928** ± 0.07 (2, 22)	**0.928** ± 0.07 (2, 22)
W/T/L	4/2/14	3/3/14	2/3/15	5/6/9
average FR	2.83	2.53	2.85	**1.80**
average FAR	48.18	42.78	45.08	**25.98**

TABLE 6. The results of each algorithm for each dataset with 5% noise included.The values in parenthesis are the Friedman ranking (FR) and the Friedman aligned ranking (FAR),respectively.

Data sets	KFHE	OKFSE	Kappa	RC_{Jac}
Balance	0.741 ± 0.13 (4, 79)	0.808 ± 0.11 (2, 4)	0.788 ± 0.16 (3, 46)	**0.817** ± 0.11 (1, 3)
Blood	0.605 ± 0.04 (4, 77)	**0.622** ± 0.04 (1, 8)	0.620 ± 0.04 (2, 10)	0.618 ± 0.04 (3, 21.5)
BreastTissue	0.690 ± 0.08 (2, 43)	**0.693** ± 0.08 (1, 13.5)	0.689 ± 0.09 (3, 54.5)	0.688 ± 0.09 (4, 62.5)
Bupa	0.684 ± 0.05 (2.5, 35)	**0.685** ± 0.05 (1, 24)	0.681 ± 0.05 (4, 69)	0.684 ± 0.05 (2.5, 35)
Ecoli	0.797 ± 0.06 (3.5, 54.5)	**0.799** ± 0.06 (1.5, 28.5)	0.797 ± 0.05 (3.5, 54.5)	**0.799** ± 0.06 (1.5, 28.5)
Ferbility	0.792 ± 0.14 (4, 80)	0.841 ± 0.13 (3, 58)	**0.882** ± 0.11 (1, 1)	0.854 ± 0.13 (2, 5)
Glass	0.731 ± 0.07 (3, 72)	**0.746** ± 0.07 (1, 6)	0.733 ± 0.07 (2, 62.5)	0.730 ± 0.06 (4, 76)
Haberman	0.563 ± 0.05 (3.5, 73.5)	0.570 ± 0.06 (2, 17)	0.563 ± 0.06 (3.5, 73.5)	**0.574** ± 0.05 (1, 7)
Hayes-roth	0.810 ± 0.06 (2, 18.5)	**0.811** ± 0.06 (1, 12)	0.805 ± 0.06 (3.5, 70.5)	0.805 ± 0.06 (3.5, 70.5)
Heart	0.793 ± 0.04 (2, 41)	0.792 ± 0.04 (3.5, 51.5)	0.792 ± 0.04 (3.5, 51.5)	**0.795** ± 0.04 (1, 20)
Image-segment	**0.888** ± 0.05 (1.5, 28.5)	**0.888** ± 0.05 (1.5, 28.5)	0.887 ± 0.04 (3, 43)	0.885 ± 0.05 (4, 62.5)
Ionosphere	**0.912** ± 0.03 (2, 39)	**0.912** ± 0.03 (2, 39)	**0.912** ± 0.03 (2, 39)	0.911 ± 0.03 (4, 49.5)
Iris	0.936 ± 0.03 (4, 75)	**0.945** ± 0.03 (1, 9)	0.939 ± 0.03 (3, 59.5)	0.943 ± 0.03 (2, 18.5)
Pima Indians	0.709 ± 0.03 (3, 49.5)	**0.711** ± 0.03 (1.5, 25.5)	0.708 ± 0.03 (4, 59.5)	**0.711** ± 0.03 (1.5, 25.5)
Seed	0.918 ± 0.03 (3.5, 67.5)	**0.923** ± 0.03 (1, 16)	0.918 ± 0.04 (3.5, 67.5)	0.922 ± 0.03 (2, 23)
Surgery	0.587 ± 0.13 (4, 78)	0.627 ± 0.16 (2, 11)	**0.656** ± 0.17 (1, 2)	0.624 ± 0.16 (3, 35)
Tae	**0.549** ± 0.08 (1, 15)	0.547 ± 0.08 (2, 31.5)	0.545 ± 0.08 (3, 57)	0.544 ± 0.08 (4, 65.5)
Wdbc	**0.952** ± 0.02 (1.5, 35)	**0.952** ± 0.02 (1.5, 35)	0.951 ± 0.02 (3.5, 47.5)	0.951 ± 0.02 (3.5, 47.5)
Wine	0.956 ± 0.04 (3, 45)	**0.958** ± 0.04 (1, 21.5)	0.954 ± 0.04 (4, 65.5)	0.957 ± 0.04 (2, 31.5)
Zoo	0.909 ± 0.07 (2, 43)	**0.912** ± 0.07 (1, 13.5)	0.908 ± 0.07 (3, 54.5)	0.907 ± 0.07 (4, 62.5)
W/T/L	1/3/16	9/5/6	2/1/17	3/2/15
average FR	2.80	**1.58**	2.95	2.68
average FAR	52.45	**22.65**	49.40	37.50

the kappa-error diagrams to visualize the diversity-accuracy patterns of the ensemble classifiers [37]. A kappa-error diagram is a scatter plot where each point corresponds to a pair of classifiers c_i and c_j. The x-axis coordinate of the point is the diversity between c_i and c_j measured by the statistic kappa κ, the same measure used in the compared method Kappa. The

TABLE 7. The results of each algorithm for each dataset with 10% noise included.The values in parenthesis are the Friedman ranking (FR) and the Friedman aligned ranking (FAR),respectively.

Data sets	KFHE	OKFSE	Kappa	RC_{Jac}
Balance	0.702 ± 0.13 (4, 80)	**0.774** ± 0.11 (1, 2)	0.766 ± 0.16 (2, 4)	0.761 ± 0.11 (3, 7)
Blood	0.606 ± 0.04 (4, 75)	**0.621** ± 0.04 (1, 10)	0.609 ± 0.04 (3, 70)	0.619 ± 0.04 (2, 13)
BreastTissue	0.673 ± 0.08 (2.5, 48.5)	**0.676** ± 0.08 (1, 20.5)	0.672 ± 0.09 (4, 55)	0.673 ± 0.09 (2.5, 48.5)
Bupa	0.665 ± 0.05 (3.5, 52.5)	**0.667** ± 0.05 (1.5, 33)	**0.667** ± 0.05 (1.5, 33)	0.665 ± 0.05 (3.5, 52.5)
Ecoli	0.783 ± 0.06 (3.5, 56.5)	0.785 ± 0.06 (2, 44)	0.783 ± 0.05 (3.5, 56.5)	**0.788** ± 0.06 (1, 19)
Ferbility	0.797 ± 0.14 (4, 79)	0.831 ± 0.13 (3, 71)	**0.861** ± 0.11 (1, 1)	0.856 ± 0.13 (2, 3)
Glass	0.719 ± 0.07 (4, 65)	**0.728** ± 0.07 (1, 11)	0.720 ± 0.07 (3, 58.5)	0.722 ± 0.06 (2, 46)
Haberman	0.546 ± 0.05 (4, 78)	**0.572** ± 0.06 (1, 5)	0.563 ± 0.06 (2.5, 26)	0.563 ± 0.05 (2.5, 26)
Hayes-roth	0.752 ± 0.06 (3, 61)	**0.765** ± 0.06 (1, 6)	0.753 ± 0.06 (2, 54)	0.748 ± 0.06 (4, 73)
Heart	0.767 ± 0.04 (2, 40)	0.766 ± 0.04 (3, 47)	0.763 ± 0.04 (4, 66)	**0.770** ± 0.04 (1, 17.5)
Image-segment	0.878 ± 0.05 (2, 29)	**0.881** ± 0.05 (1, 14)	0.870 ± 0.04 (4, 74)	0.877 ± 0.05 (3, 42)
Ionosphere	**0.891** ± 0.03 (2, 36.5)	**0.891** ± 0.03 (2, 36.5)	0.888 ± 0.03 (4, 58.5)	**0.891** ± 0.03 (2, 36.5)
Iris	0.922 ± 0.03 (4, 76)	**0.938** ± 0.03 (1, 9)	0.931 ± 0.03 (2.5, 40)	0.931 ± 0.03 (2.5, 40)
PimaIndians	0.694 ± 0.03 (3, 45)	0.695 ± 0.03 (2, 33)	0.691 ± 0.03 (4, 64)	**0.696** ± 0.03 (1, 26)
Seed	0.899 ± 0.03 (4, 69)	**0.909** ± 0.03 (1, 12)	0.900 ± 0.04 (3, 67)	0.906 ± 0.03 (2, 20.5)
Surgery	0.558 ± 0.13 (4, 77)	0.576 ± 0.16 (2, 16)	0.573 ± 0.17 (3, 36.5)	**0.582** ± 0.16 (1, 8)
Tae	0.528 ± 0.08 (3, 50.5)	**0.531** ± 0.08 (1.5, 23)	0.525 ± 0.08 (4, 68)	**0.531** ± 0.08 (1.5, 23)
Wdbc	**0.947** ± 0.02 (1.5, 30.5)	**0.947** ± 0.02 (1.5, 30.5)	0.943 ± 0.02 (4, 63)	0.946 ± 0.02 (3, 43)
Wine	0.932 ± 0.04 (3, 50.5)	**0.937** ± 0.04 (1, 15)	0.927 ± 0.04 (4, 72)	0.935 ± 0.04 (2, 23)
Zoo	**0.911** ± 0.07 (1, 17.5)	0.909 ± 0.07 (2, 28)	0.905 ± 0.07 (3.5, 61)	0.905 ± 0.07 (3.5, 61)
W/T/L	1/2/17	10/4/6	1/1/18	4/2/14
average FR	3.10	**1.53**	3.13	2.25
average FAR	55.85	**23.33**	51.40	31.43

y-axis coordinate of the point is the mean error of c_i and c_j. Both κ and the error rates are measured on the training data set [37], [39]. According to the definition of κ, the lower κ, the higher diversity. Thus, the desirable points indicating better accuracy and higher diversity should lie in the bottom left corner of the scatter plot.

Figure 3 is the kappa-error diagrams for OKFSE on the *Balance*, *Blood* and *Image-segment* datasets at noise level 5%. We performed 20 times 4-fold cross validations on these datasets, and set the sub-ensemble size M for OKFSE to 10 at Figure 3(a)-(c). From the Figure 2(d)-(f), we observed that the ensembles can be pruned to small size sub-ensembles on the *Balance* and *Blood* datasets, and the ensemble for *Image-segment* can't be pruned. Correspondingly, in Figure 3, the kappa-error diagram calculated on *Blood* has significantly better diversity than those on *Balance* and *Image-segment*, and the diversity of *Balance* is better than that of *Image-segment*. Low diversity means the sub-ensemble on *Image-segment* are highly redundant, thus it is very likely poor performed on the tested dataset. However, as the tree number increases on *Image-segment*, more diverse decision trees are added gradually into the sub-ensemble, as shown in Figure 3(d)-(f), and the performance increases accordingly.

D. DISCUSSIONS

Theorem 3 illustrates the potential redundancy of the ensemble generated by KFHE, and our experiment results validated the existence of redundancy and showed the effectiveness and robustness of OKFSE on 20 UCI datasets with class-label noises. Due to class-label noise included in the training dataset, the trained classifier deviates from the true data model in order to fit the noisy data. As the number of biased classifiers increases, the votes on wrong labels increase, thus the possibility of wrong ensemble decision increases. However, if only a suitable subset of the biased classifiers is selected, the sub-ensemble will make a tradeoff between generalization ability and deviation, and thus the decisions will be more reliable.

Since in practical classification tasks, the obtained datasets are more or less corrupted by noise, the ground truth model of these data maybe far from the classifiers built on the training data, an effective and robust ensemble technique is preferred to alleviate the impact of noise. Our experiments illustrated that all pruning algorithms show similar performance changes on a given dataset, however, our proposed OKFSE is usually the one with highest performance.

V. CONCLUSION

We analyzed the state-of-the-art multi-class ensemble classification KFHE and found that KFHE is an adaptive boosting algorithm and generates redundant classifiers when it iterates enough times. An ordering-based pruning method, OKFSE, is proposed in this paper to reduce the redundancy of the ensemble and further improve the performance of KFHE. Extensive experiments were conducted on 20 real-world datasets to compare OKFSE with the state-of-the-art

methods. The results show that, OKFSE is more effective and robust on the datasets with class-label noise than all baselines, including KFHE.

In the future work, it would be interesting to extend OKFSE to the dynamic ensemble selection, which is more promising in classification task than static ensemble selection.

APPENDIX A
EXPERIMENTAL RESULTS

See Tables 5–7.

REFERENCES

[1] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, Oct. 1990.
[2] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.
[3] M. Asafuddoula, B. Verma, and M. Zhang, "A divide-and-conquer-based ensemble classifier learning by means of many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 5, pp. 762–777, Oct. 2018.
[4] J. J. Rodríguez and J. Maudes, "Boosting recombined weak classifiers," *Pattern Recognit. Lett.*, vol. 29, no. 8, pp. 1049–1059, Jun. 2008.
[5] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.
[6] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
[7] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[8] A. Pakrashi and B. M. Namee, "Kalman filter-based heuristic ensemble (KFHE): A new perspective on multi-class ensemble classification using Kalman filters," *Inf. Sci.*, vol. 485, pp. 456–485, Jun. 2019.
[9] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.
[10] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics," *J. Geophys. Res.*, vol. 99, no. C5, p. 10143, Feb. 2004.
[11] M. Katzfuss, J. R. Stroud, and C. K. Wikle, "Understanding the ensemble Kalman filter," *Amer. Statist.*, vol. 70, no. 4, pp. 350–357, 2016.
[12] Y. Freund, "An adaptive version of the boost by majority algorithm," *Mach. Learn.*, vol. 43, no. 3, pp. 293–318, 2001.
[13] G. Martinez-Muoz, D. Hernandez-Lobato, and A. Suarez, "An analysis of ensemble pruning techniques based on ordered aggregation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 245–259, Feb. 2009.
[14] F. Pinto, C. Soares, and J. Mendes-Moreira, "Pruning bagging ensembles with metalearning," in *Multiple Classifier Systems*. Cham, Switzerland: Springer, 2015, pp. 64–75.
[15] Z. Huan, Z. Pengzhou, and G. Zeyang, "K-means text dynamic clustering algorithm based on KL divergence," in *Proc. IEEE/ACIS 17th Int. Conf. Comput. Inf. Sci. (ICIS)*, Jun. 2018, pp. 659–663.
[16] G. D. Cavalcanti, L. S. Oliveira, T. J. Moura, and G. V. Carvalho, "Combining diversity measures for ensemble pruning," *Pattern Recognit. Lett.*, vol. 74, pp. 38–45, Apr. 2016.
[17] L. Xu, B. Li, and E. Chen, "Ensemble pruning via constrained Eigen-optimization," in *Proc. IEEE 12th Int. Conf. Data Mining*, Dec. 2012, pp. 715–724.
[18] U. Bhowan, M. Johnston, M. Zhang, and X. Yao, "Reusing genetic programming for ensemble selection in classification of unbalanced data," *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 893–908, Dec. 2014.
[19] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artif. Intell.*, vol. 137, nos. 1–2, pp. 239–263, May 2002.
[20] G. Martınez-Munoz and A. Suárez, "Aggregation ordering in bagging," in *Proc. Int. Conf. Artif. Intell. Appl. (IASTED)*, 2004, pp. 258–263.
[21] L. Wang, Q. Li, Y. Yu, and J. Liu, "Region compatibility based stability assessment for decision trees," *Expert Syst. Appl.*, vol. 105, pp. 112–128, Sep. 2018.
[22] T. Sun and Z.-H. Zhou, "Structural diversity for decision tree ensemble learning," *Frontiers Comput. Sci.*, vol. 12, no. 3, pp. 560–570, Jun. 2018.
[23] R. Soares, A. Santana, A. Canuto, and M. De Souto, "Using accuracy and diversity to select classifiers to build ensembles," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Oct. 2006, pp. 1310–1316.
[24] S. T. Zouggar and A. Adla, "A new function for ensemble pruning," in *Decision Support Systems VIII: Sustainable Data-Driven and Evidence-Based Decision Support*. Cham, Switzerland: Springer, 2018, pp. 181–190.
[25] Q. Dai, R. Ye, and Z. Liu, "Considering diversity and accuracy simultaneously for ensemble pruning," *Appl. Soft Comput.*, vol. 58, pp. 75–91, Sep. 2017.
[26] Z. Lu, X. Wu, X. Zhu, and J. Bongard, "Ensemble pruning via individual contribution ordering," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2010, pp. 871–880.
[27] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Mach. Learn.*, vol. 51, no. 2, pp. 181–207, 2003.
[28] G. Martínez-Muñoz and A. Suárez, "Pruning in ordered bagging ensembles," in *Proc. 23rd Int. Conf. Mach. Learn.*, New York, NY, USA, 2006, pp. 609–616.
[29] H. Guo, H. Liu, R. Li, C. Wu, Y. Guo, and M. Xu, "Margin & diversity based ordering ensemble pruning," *Neurocomputing*, vol. 275, pp. 237–246, Jan. 2018.
[30] L. Guo and S. Boukir, "Margin-based ordered aggregation for ensemble pruning," *Pattern Recognit. Lett.*, vol. 34, no. 6, pp. 603–609, Apr. 2013.
[31] R. Hu, S. Zhou, Y. Liu, and Z. Tang, "Margin-based Pareto ensemble pruning: An ensemble pruning algorithm that learns to search optimized ensembles," *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–12, Jun. 2019.
[32] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "New ordering-based pruning metrics for ensembles of classifiers in imbalanced datasets," in *Proc. 9th Int. Conf. Comput. Recognit. Syst. (CORES)*. Cham, Switzerland: Springer, 2016, pp. 3–15.
[33] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "Ordering-based pruning for improving the performance of ensembles of classifiers in the framework of imbalanced datasets," *Inf. Sci.*, vol. 354, pp. 178–196, Aug. 2016.
[34] X. Zhu, Z. Ni, L. Ni, F. Jin, M. Cheng, and J. Li, "Improved discrete artificial fish swarm algorithm combined with margin distance minimization for ensemble pruning," *Comput. Ind. Eng.*, vol. 128, pp. 32–46, Feb. 2019.
[35] R. E. Schapire and Y. Freund, *Boosting: Foundations and Algorithms*. Cambridge, MA, USA: MIT Press, 2014, pp. 4–14.
[36] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml
[37] D. D. Margineantu and T. G. Dietterich, "Pruning adaptive boosting," in *Proc. ICML*, vol. 97, 1997, pp. 211–218.
[38] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Stat. Assoc.*, vol. 32, no. 200, pp. 675–701, Dec. 1937.
[39] C.-X. Zhang and J.-S. Zhang, "A novel method for constructing ensemble classifiers," *Stat. Comput.*, vol. 19, no. 3, pp. 317–327, Sep. 2009.

KAI YU was born in Shouguang, China, in 1995. He received the B.S. degree in computer science from Yantai University, China, in 2018, where he is currently pursuing the M.S. degree in computer science under supervision of Prof. L. Wang. His main research interests include data mining and ensemble techniques.

LIHONG WANG was born in Jinlin, China, in 1970. She received the B.S. degree from Tsinghua University, China, in 1990, the M.S. degree from the University of Science and Technology of China (USTC), in 1993, and the Ph.D. degree from Shanghai University, China, in 2004. She is currently a Professor with the School of Computer and Control Engineering, Yantai University. Her research interests include data mining and machine learning.

YANWEI YU (Member, IEEE) was born in Heze, China, in 1986. He received the B.S. degree from Liaocheng University, China, in 2008, and the Ph.D. degree from the University of Science and Technology Beijing, China, in 2014. From 2012 to 2013, he was a Visiting Scholar with the Department of Computer Science, Worcester Polytechnic Institute. From 2016 to 2018, he was a Postdoctoral Researcher with the College of Information Sciences and Technology, Pennsylvania State University. He is currently an Associate Professor with the Department of Computer Science and Technology, Ocean University of China. His research interests include data mining, machine learning, and database systems.