Random Hyperboxes

Thanh Tung Khuat, *Student Member, IEEE*, and Bogdan Gabrys, *Senior Member, IEEE*

***Abstract*—This paper proposes a simple yet powerful ensemble classifier, called Random Hyperboxes, constructed from individual hyperbox-based classifiers trained on the random subsets of sample and feature spaces of the training set. We also show a generalization error bound of the proposed classifier based on the strength of the individual hyperbox-based classifiers as well as the correlation among them. The effectiveness of the proposed classifier is analyzed using a carefully selected illustrative example and compared empirically with other popular single and ensemble classifiers via 20 datasets using statistical testing methods. The experimental results confirmed that our proposed method outperformed other fuzzy min-max neural networks, popular learning algorithms, and is competitive with other ensemble methods. Finally, we indentify the existing issues related to the generalization error bounds of the real datasets and inform the potential research directions.**

***Index Terms*—General fuzzy min-max neural network, classification, random hyperboxes, randomization-based learning, ensemble learning.**

I. Introduction

A Random hyperboxes (RH) classifier encompasses many individual hyperbox-based learners, e.g., fuzzy min-max neural networks (FMNNs) [1]. One of the key characteristics of hyperbox-based classifiers is the single-pass through the training data learning ability. Based on this incremental learning ability, new data and classes can be added to the model without retraining the whole network. Another interesting characteristic of hyperbox-based models is their interpretability thanks to the human understandable rule sets which can be extracted directly or indirectly from hyperboxes. Interpretability is one of the key requirements when applying machine learning algorithms to high-stakes applications such as medical diagnostics, financial investment, self-driving systems, and criminal justice [2].

The random hyperboxes model can be categorized into the family of ensemble classifiers, which build many base estimators and then combine them to create a final model. It is well-known that ensemble models are usually much more accurate than their base learners [3]. There are two main methods to construct an ensemble model when using resampling methods and the same type of base learners. The first one aims to build many independent or low correlation individual estimators and combining their predictive outputs using majority voting or averaging approach. The representative models for this group include Bagging [4] and Random Forests [5]. The second paradigm consists of algorithms building base estimators in a sequential manner, where the newly added learner tries to correct errors generated by previous classifiers. Adaptive boosting (Adaboost) [6] and Gradient Boosting Machines [7] are typical algorithms under the boosting framework. Extreme Gradient Boosting (XGBoost) [8] and LightGBM [9] are two recent effective and scalable implementations of the gradient boosting algorithm.

Our random hyperboxes classifier belongs to the first group because it shares the same principle with the bagging, i.e., using individual hyperbox-based learners with low correlation and combining their outputs by the majority voting. As shown in a recent study on hyperbox-based machine learning algorithms [10], there is only one study [11] related to the use of bagging techniques with hyperbox-based models as base learners and another one which is concerned with method independent learning approaches for constructing either ensembles or individual hyperbox-based classifiers [12]. In their work, after training individual hyperbox-based estimators on different subsets of the training sets, the resulting base learners are combined at the decision level using the majority voting or averaging of membership values or combined at the model level into a single model. However, as it has been frequently shown resampling methods used with bagging like algorithms operating only in the sample space can generate a limited level of diversity amongst the base classifiers trained in this way. As the diversity amongst the base learners is of key importance [13], there is another mechanism needed for making the resulting ensembles more effective and well performing. Based on Lemma 1, adapted from [14], it can be seen that the high correlation between base learners leads to a high testing error for the average classifier. To cope with this problem, we will lower the correlation but without significantly increasing the variance σ of individual hyperbox-based learners by using only a subset of features when building base estimators. This fact can be achieved by utilizing feature subsets selected randomly for training each base classifier besides the subsets of samples. The use of a subsampling technique for both sample and feature spaces to construct the ensemble model constitutes the core principle of the random hyperboxes classifier. From surveys on hyperbox-based machine learning algorithms [10] and fuzzy min-max neural networks [15], it can be observed that this paper is the first study using randomized hyperbox estimators trained on subsets of both samples and features to construct an ensemble model.

Lemma 1. *Given m identically distributed random variables (not necessarily independent) with the variance of each variable σ^2 and positive pairwise correlation ρ, the variance of the average random variable is:*

$$\rho \cdot \sigma^2 + \frac{1-\rho}{m} \cdot \sigma^2 \tag{1}$$

T.T. Khuat (email: thanhtung.khuat@student.uts.edu.au) and B. Gabrys (email: Bogdan.Gabrys@uts.edu.au) are with Advanced Analytics Institute, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia.

Proof. See Appendix A. □

The use of subsets of features in building classifiers results in many effective models such as randomized trees on geometric feature selection [16], the random subspace-based decision forests [17], and random forests [5]. Recently, there have been several studies focusing on employing random projections of the feature vectors into a lower-dimensional space to form training data for classifiers such as Fisher's linear discriminant [18], random projection neural network [19], or a general framework of random-projection based ensemble models [20]. These results have provided further motivation for the proposed random hyperboxes classifier.

One of the interesting characteristics of the proposed classifier is that it is easy to scale with large-sized training sets because each base learner can be constructed independently, so the learning process may be parallelised easily. Our contributions in this paper can be summarized as follows:

- We propose a new ensemble classifier built from individual hyperbox-based learners using random subsets of both sample and feature spaces.
- We derive a generalization error bound of the RH classifier based on the strength and correlation between base learners.
- We analyze the effectiveness of the RH classifier in comparison to its base learners concerning the decrease in the variance of the ensemble model and the increase in the accuracy. We have also conducted extensive experiments on 20 datasets to compare the performance of the proposed method to other FMNNs as well as popular single and ensemble classifiers.
- We discuss the generalization error bounds on the real dataset and inform the open research directions.

The rest of this paper is structured as follows. Section II presents the general fuzzy min-max neural network (GFMMNN) and its learning algorithms used for base learners. In section III, the formal description of the proposed method is provided and the generalization error bounds are derived. Section IV is devoted to experimental results. We discuss several issues concerning the generalization error bounds on the real datasets and identify the open problems in Section V. Section VI concludes the findings and proposes directions for the future work.

II. Preliminaries

The RH classifier is constructed from base learners which can deploy any hyperbox-based machine learning algorithms. However, in this paper, we use the GFMMNN as base learners to assess the efficiency of the proposed method. Therefore, this part provides the readers with some basic knowledge of the GFMMNN and its learning algorithms.

The GFMMNN [1] is a generalized version of FMNNs for classification [21] and clustering [22]. Its structure includes three layers, in which the input layer can accept both crisp and fuzzy data. Therefore, the input layer contains $2p$ nodes corresponding to p features of the input data which can be represented in the form of lower and upper bounds (i.e. as a real interval). The second layer consists of hyperboxes dynamically created during the learning process. The connection weights between the first and the second layers are the minimum points $\mathbf{V}$ and the maximum points $\mathbf{W}$ of hyperboxes, which are adjusted in the learning process. The connection between the hyperbox B_i in the second layer and an output node c_i in the third layer u_{ij} is stored in the matrix $\mathbf{U}$ such that:

$$u_{ij} = \begin{cases} 1, \text{if } class(B_i) = c_j \\ 0, \text{otherwise} \end{cases} \tag{2}$$

In the GFMMNN, the degree of fit of each hyperbox $B_i = [V_i, W_i]$, where minimum point $V_i = [v_{i1}, \ldots, v_{ip}]$ and maximum point $W_i = [w_{i1}, \ldots, w_{ip}]$, with respect to each input pattern $\mathbf{x} = [\mathbf{x}^l, \mathbf{x}^u]$ is computed using a membership function as Eq. (3).

$$\begin{aligned} b_i(\mathbf{x}, B_i) = \min_{j=1}^{p}(\min([1 - f(x_j^u - w_{ij}, \gamma_j)], \\ [1 - f(v_{ij} - x_j^l, \gamma_j)])) \end{aligned} \tag{3}$$

where $f(\xi, \gamma)$ is two-parameter ramp function described in Eq. (4), $\gamma = (\gamma_1, \gamma_2, ..., \gamma_p)$ contains the sensitivity parameters regulating the decreasing speed of the membership values, and $0 \le b_i(\mathbf{x}, B_i) \le 1$.

$$f(\xi, \gamma) = \begin{cases} 1, & \text{if } \xi \cdot \gamma > 1 \\ \xi \cdot \gamma, & \text{if } 0 \le \xi \cdot \gamma \le 1 \\ 0, & \text{if } \xi \cdot \gamma < 0 \end{cases} \tag{4}$$

In the classification phase, assuming that the membership value between the input $\mathbf{x}$ and the hypberbox B_i is the highest compared to other existing hyperboxes, the predictive class of the model for the input $\mathbf{x}$ is the class of B_i.

Given a training set, there are two types of learning algorithms used to train the GFMM classifier, i.e., the incremental (online) learning [1] and agglomerative (batch) learning [23]. The batch learning algorithm starts with all of the training samples and then repeatedly merging hyperboxes with the same class satisfying the maximum hyperbox size (θ), minimum similarity threshold (σ_s), and no generation of overlapping regions with hyperboxes of other classes. The training time of this algorithm is long because of the iterative computation of membership and similarity values between all pairs of existing hyperboxes. In contrast, the online learning algorithm is much faster since it uses a single pass mechanism through learning samples to build and adjust hyperboxes. However, the hyperbox contraction process to resolve hyperbox overlapping areas causes a decrease in predictive accuracy [24]. In a recent study, an improved online learning algorithm of GFMMNN, called IOL-GFMM, has been proposed to combine the strong points of both incremental and batch learning algorithms. Therefore, in this paper, the IOL-GFMM will be used to build base hyperbox classifiers. We would like to refer the readers to [25] for more details of the IOL-GFMM algorithm.

III. Proposed Method

A. Formal Description

Let us denote by $\mathcal{T}_n = \{(\mathbf{x}_i, c_i)\}_{i=1}^n$ a training data where $x_i \in \mathbf{X} \subset \mathbb{R}^p$ is a p-dimensional vector of observations (i.e.

features) and $c_i \in \mathcal{C}$, $\mathcal{C}$ is a set of categorical variables denoting classes to which the observations fall. Given an input $\mathbf{x}$, our goal is to build an ensemble classifier which predicts class c from $\mathbf{x}$ using the training data $\mathcal{T}_n$.

Please note that for the theoretical considerations of the proposed algorithm covered in this section and the discussion of the convergence properties and the derivation of generalisation error bounds presented in Section III-C, an assumption is made that the observations are independent and identically distributed (i.i.d.) random variables.

A random hyperboxes model with m hyperbox-based learners is a classifier including a set of randomized base hyperbox models $h(\mathbf{x}, \Phi_1), \ldots, h(\mathbf{x}, \Phi_m)$, where $\Phi_1, \ldots, \Phi_m$ are i.i.d. random vectors of a randomizing vector Φ, independent conditionally on $\mathbf{X}, \mathcal{C}$, and $\mathcal{T}_n$. Each individual hyperbox-based learner $h(\mathbf{x}, \Phi_i)$ is constructed using the training set $\mathcal{T}_n$ and a random vector Φ_i. Φ_i introduces the randomness to the building process of hyperbox-based learners including the determining of a subset $\mathcal{T}_{\Phi_i}$ of the full training data $\mathcal{T}_n$ as well as determining a subset of features $\mathbf{x}_{\Phi_i}$ used. After a large number of hyperbox-based learners genereated, the random hyperboxes estimator takes the class with most votes among base learners as its predictive result. Formally, the definition of the random hyperboxes classifier can be stated as follows:

Definition 1. *A random hyperboxes model is a classifier including a set of hyperbox-based learners $\{h(\mathbf{x}, \Phi_i) : i = 1, \ldots, m\}$, where $\{\Phi_i\}$ are independent and identically distributed random vectors of a model random vector Φ independent conditionally on sample space $(\mathbf{X}, \mathcal{C})$ and the training set $\mathcal{T}_n$. Each hyperbox-based learner gives a unit vote based on the class of the hyperbox with the maximum membership degree with respect to the input pattern $\mathbf{x}$. The predictive result of the random hyperboxes model is the aggregation of predictive results from its base learners using a majority voting method.*

In particular, the predictive class ($c_k \in \mathcal{C}$) with respect to input data $\mathbf{x}$ of a random hyperboxes classifier including m base learners (let $\Phi^{(m)} = \{\Phi_1, \ldots, \Phi_m\}$) can be shown as follows:

$$h(\mathbf{x}, \Phi^{(m)}) = arg \max_{c_k \in \mathcal{C}} \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}(h(\mathbf{x}, \Phi_i) = c_k)$$

where $\mathbb{1}(\cdot)$ is the indicator function. According to the strong law of large numbers, when the number of base learners increases, we almost surely obtain $\lim\limits_{m \to \infty} h(\mathbf{x}, \Phi^{(m)}) = \overline{h}(\mathbf{x}, \Phi)$, where $\overline{h}(\mathbf{x}, \Phi) = arg \max\limits_{c_k \in \mathcal{C}} \mathbb{E}_\Phi[\mathbb{1}(h(\mathbf{x}, \Phi) = c_k)]$ (Here $\mathbb{E}_\Phi$ denotes the expectation with regard to the random variable Φ).

Each random hyperbox-based learner $h(\mathbf{x}, \Phi)$ is formed as follows. We select randomly a subset $\mathcal{T}_l$ including $l < n$ samples from the full training data $\mathcal{T}_n$ using subsampling method without replacement under weak assumptions $l \to 0$ and $r_s = l/n \to 0$ as $n \to \infty$. According to [26], under the weak convergence hypothesis, the sampling distributions of $\mathcal{T}_l$ and $\mathcal{T}_n$ should be close, and they will converge to the true unknown distribution of whole sample space. After that, we will select at uniformly random d ($1 \leq d \leq m_f \leq p$) features from p features of $\mathcal{T}_l$ to form a training set $\mathcal{T}_l^{(d)}$ for $h(\mathbf{x}, \Phi)$, where m_f is the maximum features used for each base learner. There are many learning algorithms which could be used to train the base hyperbox-based classifier $h(\mathbf{x}, \Phi)$ on $\mathcal{T}_l^{(d)}$. This study uses the IOL-GFMM [25] to build the base estimators. This is a new online learning algorithm of GFMM which integrates the advantages of the incremental learning and batch learning algorithms for the building process of a GFMMNN. It is noted that the base model $h(\mathbf{x}, \Phi)$ is trained on only d features of $\mathcal{T}_n$, so in the classification step, $h(\mathbf{x}, \Phi)$ only makes prediction using the same d features with respect to the unseen sample $\mathbf{x}$. The learning and classification steps for each base learner are kept the same as in the IOL-GFMM algorithm.

Algorithm 1 Training algorithm of the Random hyperboxes

```
Input: training set $\mathcal{T}$, sampling rate for samples $r_s$, maximum number of used features $m_f$, number of base estimators $m$, maximum hyperbox size $\theta$, sensitivity parameter $\gamma$
Output: A random Hyperboxes model $\mathbf{H}$

$i = 1; \mathbf{H} \leftarrow \varnothing$
for $i \leq m$ do
    $T_i \leftarrow$ Perform subsampling on $\mathcal{T}$ with rate $r_s$
    $d \leftarrow$ Generate a uniform random number in the range of $[1, m_f]$
    $T_i^d \leftarrow$ Random sampling $d$ features of $T_i$
    $h_i \leftarrow$ IOL-GFMM($T_i^d, \gamma, \theta$)
    $\mathbf{H} \leftarrow \mathbf{H} \cup h_i$
    $i = i + 1$
end for
return $\mathbf{H}$
```

The basic steps of the building process of the random hyperboxes classifier are shown in Algorithm 1.

B. Time Complexity

Based on Algorithm 1, it is easily observed that the time complexity of a random hyperboxes model depends mainly on the time complexity of the training process for each base learner. As discussed in [27], the time complexity of the IOL-GFMM algorithm trained on a dataset containing n samples with p features is $\mathcal{O}(n \cdot \mathcal{K} \cdot \mathcal{R} \cdot p)$, where $\mathcal{K}$ is the average number of expandable hyperbox candidates and $\mathcal{R}$ is the average number of hyperboxes representing classes different from the input pattern class for each iteration in the training process. For the random hyperboxes model, each base learner is trained on only $l < n$ samples with the maximum $m_f < p$ features. Therefore, the time complexity of each base learner in the worst case is $\mathcal{O}(l \cdot \mathcal{K} \cdot \mathcal{R} \cdot m_f)$. We need to build m base learners for a random hyperboxes classifier. As a result, if the base learners are sequentially constructed, the time complexity of training a random hyperboxes model in the worst case is $\mathcal{O}(m \cdot l \cdot \mathcal{K} \cdot \mathcal{R} \cdot m_f)$.

C. Properties of the Random Hyperboxes

1) The Convergence of the Random Hyperboxes Model:

Let $\mathbf{x}$ be a random sample, drawn from the sample space, to be classified with true class c. Let $\mathcal{T}_n$ be a random training set drawn i.i.d. from the true distribution of sample space $(\mathbf{X}, \mathcal{C})$. Given an ensemble of m base learners $h_1(\mathbf{x}), \ldots, h_m(\mathbf{x})$, where $h_i(\mathbf{x}) \equiv h(\mathbf{x}, \Phi_i)$, we can define a margin function of a random hyperboxes model with m base estimators for an input sample $\mathbf{x}$ as Eq. (5):

$$\mathcal{M}(\mathbf{x}, c) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}(h_i(\mathbf{x}) = c) - \max_{j \neq c} \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}(h_i(\mathbf{x}) = j) \tag{5}$$

where $\mathbb{1}(\cdot)$ is the indicator function.

Remark. *The margin can be considered as a confidence measure with respect to the classification result of the random hyperboxes model. A large margin increases the confidence in predictive results for observations and vice versa.*

Based on the above margin function, the generation error of the random hyperboxes model is defined as follows:

Definition 2. *The generalization error is the probability $\mathbf{P}_{\mathbf{X},\mathcal{C}}$ measured in the sample space $(\mathbf{X}, \mathcal{C})$ that gives a negative margin: $\mathcal{E} = \mathbf{P}_{\mathbf{X},\mathcal{C}}(\mathcal{M}(\mathbf{x}, c) < 0)$*

Lemma 2. *When the number of base estimators increases $(m \to \infty)$ and base estimators are independent, for almost surely all i.i.d. random vectors $\Phi_1, \Phi_2, \ldots$, the margin function for a random hyperboxes model $\mathcal{M}(\mathbf{x}, c)$ at each input $\mathbf{x}$ converges to:*

$$\mathcal{M}^*(\mathbf{x}, c) = \mathbf{P}_\Phi(h(\mathbf{x}, \Phi) = c) - \max_{j \neq c} \mathbf{P}_\Phi(h(\mathbf{x}, \Phi) = j) \tag{6}$$

Proof. See Appendix B. □

From definition 2 and lemma 2, we achieve the following theorem for the convergence of generalization error:

Theorem 1. *When the number of base learners increases $(m \to \infty)$, for almost surely all random vectors $\Phi_1, \Phi_2, \ldots$, the generalization error $\mathcal{E}$ converges to: $\mathcal{E}^* = \mathbf{P}_{\mathbf{X},\mathcal{C}}\left[\mathcal{M}^*(\mathbf{x}, c) < 0\right]$*

This theorem explains that the random hyperboxes model does not overfit when more base learners are added to the model if hyperbox-based learners are independent and under the i.i.d. assumption. In the next subsection, the upper bound of the generalization error will be derived.

2) Generalization Error Bound:

Based on Lemma 1, we can observe that to decrease the variance of the average classifier, we need to reduce the correlation of base learners. However, if the correlation decreases, the variance of base learners usually increases, and it makes the reduction of the prediction error harder. The correlation among base learners can be easily decreased by increasing base models' randomness. However, in this way the variance of the base learners will also be increased. Therefore, we should not let the variance increase too fast. To cope with this issue, we can inspect and monitor the change in the generalization error bound.

Instead of having a fixed number of base estimators m, let us assume that we have a fixed probability distribution for the random vector Φ from which base models are constructed. Similarly to random forests [5], we can define the strength of the random hyperbox model based on the limit of the margin function as follows:

Definition 3. *The strength of the random hyperboxes model is defined as:*

$$\mathcal{S} = \mathbb{E}_{\mathbf{X},\mathcal{C}}\mathcal{M}^*(\mathbf{x}, c) \tag{7}$$

where $\mathbb{E}_{\mathbf{X},\mathcal{C}}$ is the expectation through the $(\mathbf{X}, \mathcal{C})$ space.

Assuming that $\mathcal{S} > 0$, according to Chebyshev's inequality, we have:

$$\begin{aligned}\mathcal{E}^* &= \mathbf{P}_{\mathbf{X},\mathcal{C}}\left[\mathcal{M}^*(\mathbf{x}, c) < 0\right] \leq \mathbf{P}_{\mathbf{X},\mathcal{C}}\left[\mathcal{S} - \mathcal{M}^*(\mathbf{x}, c) \geq \mathcal{S}\right] \\ &= \mathbf{P}_{\mathbf{X},\mathcal{C}}\left[|\mathcal{M}^*(\mathbf{x}, c) - \mathcal{S}| \geq \mathcal{S}\right] \leq \frac{\mathtt{Var}_{\mathbf{X},\mathcal{C}}(\mathcal{M}^*(\mathbf{x}, c))}{\mathcal{S}^2}\end{aligned}$$

This is a weak upper bound of the generalization error, and it indicates that the prediction error is always lower than an explicit but unknown limit. The value of $\mathcal{S}$ can be estimated over the training set $\mathcal{T}_n$ as follows:

$$\begin{aligned}\overline{\mathcal{S}} &= \frac{1}{n}\sum_{i=1}^{n} \mathcal{M}(\mathbf{x}_i, c_i) \\ &= \frac{1}{nm}\sum_{i=1}^{n}\Big(\sum_{k=1}^{m} \mathbb{1}(h_k(\mathbf{x}_i) = c_i) - \max_{j \neq c_i} \sum_{k=1}^{m} \mathbb{1}(h_k(\mathbf{x}_i) = j)\Big)\end{aligned}$$

Let $J(\mathbf{x}, c) = arg\max_{j \neq c} \mathbf{P}_\Phi(h(\mathbf{x}, \Phi) = j)$ be the class j leading to the most incorrect classification of base learners with respect to the input $\mathbf{x}$. Then, we can define a raw margin function for each base learner at each input $\mathbf{x}$ as follows:

Definition 4. *The raw margin function is defined by:*

$$\begin{aligned}\mathcal{R}(\Phi) = \mathcal{R}(\mathbf{x}, c, \Phi) =& \mathbb{1}(h(\mathbf{x}, \Phi) = c) \\ &- \mathbb{1}(h(\mathbf{x}, \Phi) = J(\mathbf{x}, c))\end{aligned} \tag{8}$$

Following from the above definition,

$$\begin{aligned}\mathcal{M}^*(\mathbf{x}, c) &= \mathbf{P}_\Phi(h(\mathbf{x}, \Phi) = c) - \mathbf{P}_\Phi(h(\mathbf{x}, \Phi) = J(\mathbf{x}, c)) \\ &= \mathbb{E}_\Phi\left[\mathbb{1}(h(\mathbf{x}, \Phi) = c) - \mathbb{1}(h(\mathbf{x}, \Phi) = J(\mathbf{x}, c))\right] \\ &= \mathbb{E}_\Phi \mathcal{R}(\Phi)\end{aligned}$$

It means that the limit of the margin values is the expectation of raw margin values computed over all realizations of Φ.

From the above raw margin function, we now can define the correlation between two hyperbox-based learners $h(\mathbf{x}, \Phi_i)$ and $h(\mathbf{x}, \Phi_j)$ generated from two i.i.d. random vectors Φ_i and Φ_j as follows:

Definition 5. *The correlation between two hyperbox-based learners $h(\mathbf{x}, \Phi_i)$ and $h(\mathbf{x}, \Phi_j)$ of a random hyperboxes model can be calculated from the raw margin function through all observations as follows:*

$$\rho_{\mathbf{X},\mathcal{C}}(\Phi_i, \Phi_j) = \frac{\mathtt{Cov}_{\mathbf{X},\mathcal{C}}(\mathcal{R}(\Phi_i), \mathcal{R}(\Phi_j))}{\sigma_{\mathbf{X},\mathcal{C}}(\mathcal{R}(\Phi_i))\sigma_{\mathbf{X},\mathcal{C}}(\mathcal{R}(\Phi_j))} \tag{9}$$

where $\mathtt{Cov}$ is the covariance, $\sigma_{\mathbf{X},\mathcal{C}}(\mathcal{R}(\Phi_i))$ denotes the standard deviation of $\mathcal{R}(\Phi_i)$, holding Φ_i fixed, computed over observations.

Generally, the average correlation between base learners in the random hyperboxes models is computed through all pairs of two i.i.d. random vectors Φ and Φ' as follows:

$$\overline{\rho} = \mathbb{E}_{\Phi,\Phi'}[\rho_{\mathbf{X},\mathcal{C}}(\Phi, \Phi')] \tag{10}$$

From the average correlation between base learners and the strength $\mathcal{S}$, we have the following theorem for the upper bound of the generalization error:

Theorem 2. *An upper bound of the generalization error for the random hyperboxes model can be estimated from the strength of base learners and correlation between base learners as follows:*

$$\mathcal{E}^* \leq \overline{\rho}\left(\frac{1}{\mathcal{S}^2} - 1\right) \tag{11}$$

Proof. See Appendix C. □

IV. Experimental Results

It is noted that the derivations and proofs in the previous section have been carried out under the i.i.d. assumption which in practice is difficult to verify and is very often not satisfied. In this section and the appendices we are, therefore, conducting extensive benchmarking and experimental evaluation of the proposed method to also verify its practical characteristics and performance.

A. Analyzing the Random Hyperboxes Classifier

1) The Decrease in the Variance Compared to Base Learners: To conduct this experiment, we used six datasets with diversity in the numbers of samples, features, and classes. All of the experimental results are shown in Appendix D-A. This section only illustrates the results for a dataset of the one-hundred plant species leaves for margin [28]. This dataset includes 1600 samples with 64 features and 100 classes. We performed 10 times repeated 4-fold cross-validation to evaluate the ensemble model with 100 base learners. Therefore, there are 4000 base learners using the IOL-GFMM algorithm and 40 random hyperboxes models generated. The variance values in terms of weighted-F1 scores of base learners and the random hyperboxes models are shown in Fig. 1. The variance values of other datasets are shown in Fig. 11 in Appendix D-A. These results confirmed that the variance of random hyperboxes models using simple majority voting is significantly reduced compared to their base learners, so its classification accuracy is also higher than that of base estimators.

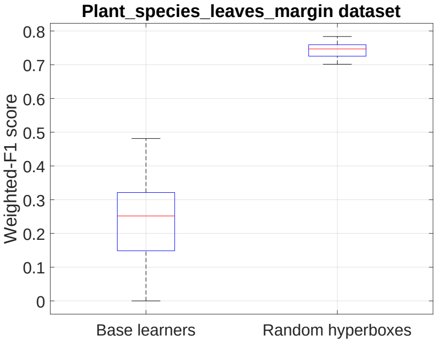


Fig. 1. The variances of RH models and their base learners (*plant_species_leaves_margin* dataset).

In this experiment, we set the maximum number of used features $m_f = 2\sqrt{p} = 16$ (for the *plant_species_leaves_margin* dataset) and 50% of the training data samples were randomly selected to train each base learner. The probability of the number of features, d, used to build the 4000 base learners is shown in Fig. 12 in Appendix D-A. The importance scores of features through all base learners can be identified using the used probability of each feature, as shown in Fig. 13 in Appendix D-A.

Based on the probability that each feature is used in 4000 base learners, we can determine the contribution of the combination of features to the performance of each classifier. Therefore, we have trained a single model using the IOL-GFMM algorithm using top-K most used features $(K = 1, \dots, p)$ $(p = 64$ for the *plant_species_leaves_margin* dataset) in each iteration. Fig. 2 shows the average weighted-F1 scores for 40 testing folds (10 times repeated 4-fold cross-validation) for each top-K of the most often used features in the *plant_species_leaves_margin* dataset. The results for the other datasets can be found in Fig. 14 in Appendix D-A. It can be seen that the single model usually achieves the best performance if it is trained on all features. However, by using the random hyperboxes method with base learners trained on only a maximum of m_f features, we can obtain a higher accuracy than the single model trained on all features. Furthermore, in several datasets such as *ringnorm* and *connectionist_bench_sonar*, the best performance is often obtained when using a subset of the most crucial features. It is due to the fact that the redundant features can prevent the single GFMM model from learning the true distribution of the underlying data with a given finite number of training samples. Therefore, the use of the random hyperboxes model of which base learners are trained on a subset of features can capture the data distribution more effectively and achieve better classification performance compared to the case of employing of a single GFMM model.

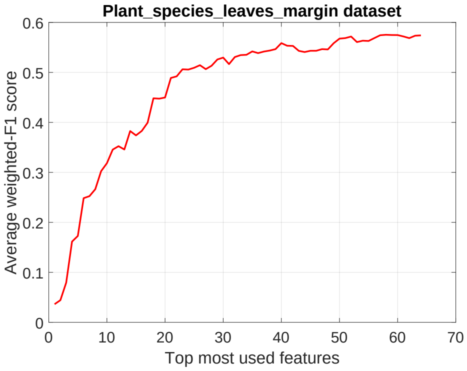


Fig. 2. Average weighted-F1 scores through 40 testing folds of a single model using training sets with top-k most used features (*plant_species_leaves_margin* dataset).

In general, the RH classifier can achieve much better performance compared to the single IOL-GFMM classifier trained on full feature space, especially for very high dimensional datasets. These results are shown in Appendix D-B.

2) *The Roles of the Number of Base Learners and Maximum Number of Used Features:*

This experiment is to assess the sensitivity of hyper-parameters such as the number of base learners and the maximum number of used features on the performance of the random hyperboxes model. We used eight datasets with diversity in the numbers of samples, classes, and features for this purpose. All of the empirical results can be found in Appendix D-C. This section only illustrates the outcomes of the same dataset used in subsection IV-A1. To evaluate the impact of the number of base learners on the performance of the random hyperboxes model, we kept the maximum number of used features $m_f = 2 \cdot \sqrt{p}$ ($m_f = 16$ in this case), the maximum hyperbox size of each base learner $\theta = 0.1$, and 50% of samples were randomly selected to train each base estimator. The number of base learners is set from 5 to 200 with step 5. Fig. 3 shows the average weighted-F1 scores over 10 times repeated 4-fold cross-validation at each threshold for the *plant_species_leaves_margin* dataset. The results for the other datasets can be found in Fig. 17 in Appendix D-C. It can be observed that the performance of the random hyperboxes classifier is not reduced as more base learners are added. These figures confirm that the random hyperboxes classifier does not overfit when adding more base learners.

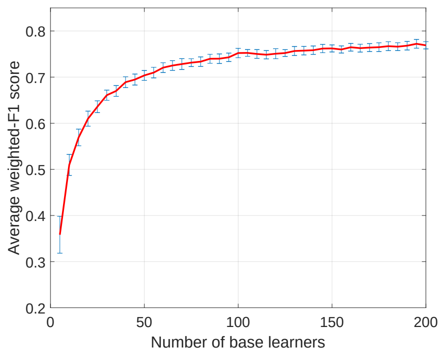


Fig. 3. The change in the average weighted-F1 scores when increasing the number of base learners (*plant_species_leaves_margin* dataset).

To assess the influence of the maximum number of used features m_f, we kept the number of base learners $m = 100$, $\theta = 0.1$, $r_s = 0.5$, and changed the maximum numbers of used features from 1 to p ($p = 64$ in this case). Fig. 4 depicts the average weighted-F1 scores for 10 times repeated 4-fold cross-validation at each value of the maximum number of used features for the *plant_species_leaves_margin* dataset. The outcomes for the remaining datasets are shown in Fig. 18 in Appendix D-C.

It can be easily observed that the overall trend when increasing the maximum number of used features is that the accuracy of the random hyperboxes classifier only increases to a certain threshold, and then its accuracy will decrease. It is due to the fact that the correlation between base learners will be higher when we use too many features for each base learner. In contrast, if too few features are used, the strength of each base learner gets a low value, so the error of the ensemble model will increase. This fact confirms that the maximum number of used features is an important parameter, which needs to be carefully selected to achieve the high accuracy for the random hyperboxes classifier.

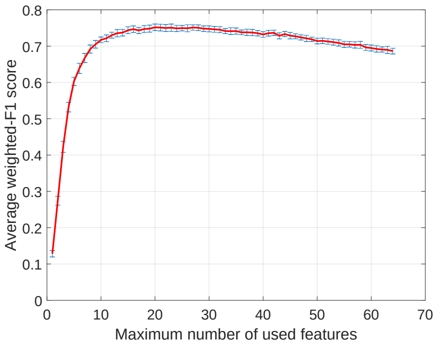


Fig. 4. The change in the average weighted-F1 scores when increasing the maximum number of used dimensions (*plant_species_leaves_margin* dataset).

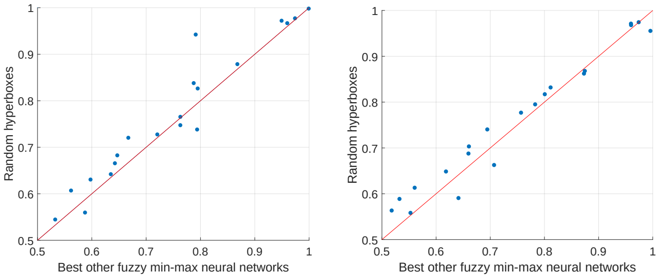
B. *Comparing the Performace of the Random Hyperboxes to Other Classifiers*

The datasets used and parameter settings for models are presented in Appendix D-D1. The following results are the average weighted-F1 scores using 10 times repeated 4-fold cross-validation. In this study, we consider the multi-class classification problem, so the weighted-F1 measure is more suitable and less biased than the often used classification accuracy. Weighted F1-score is the average F1-score of each class weighted by the support which is the number of patterns of each class. In each iteration, three folds were used for training and one remaining fold was used as a testing set.

1) *A Comparison of the Random Hyperboxes With Other FMNNs:*

This experiment compares the RH model with FMNN [21], online learning version of GFMMNN (Onln-GFMM) [1], agglomerative learning algorithm version 2 of GFMMNN (AGGLO-2) [23], combination of Onln-GFMM at $\theta = 0.05$ and AGGLO-2 [23], IOL-GFMM [25], enhanced fuzzy min-max neural network (EFMNN) [29], enhanced fuzzy min-max neural network with k-nearest hyperbox selection rules (KNEFMNN) [30], and refined fuzzy min-max neural network (RFMNN) [31]. The classification accuracy results of fuzzy min-max neural networks at low values of θ are usually better than those at high values of θ [32]. Therefore, in this experiment, we will compare the RH model with other FMNNs using $\theta = 0.1$ and $\theta = 0.7$. The average weighted-F1 scores of classifiers, as well as their ranks, are shown in Tables from VI to IX in Appendix D-D2. Fig. 5 summarizes these results by comparing the results of the RH classifier with the best values of other FMNNs. We can see that in both subplots most points locate above the diagonal line, these figures illustrate the efficiency and robustness of the random hyperboxes for both low and high thresholds of θ.

Using the Friedman rank-sum test [33], we can compute the F-distribution value $F_F = 10.1868$ from the average ranks of models at $\theta = 0.1$. Since the critical value of $F(8, 152)$ for the significance level $\alpha = 0.05$ is 1.9998, the null hypothesis



(a) $\theta = 0.1$ (b) $\theta = 0.7$

Fig. 5. Comparison of average weighted-F1 scores of the random hyperboxes and the best value from single FMNNs.

is rejected. It means that there are significant differences between the average weighted-F1 scores of these models. To further compare the peformance of the RH model to other FMNNs at $\theta = 0.1$, the Critical Difference (CD) diagram with Bonferroni-Dunn test [34] for $\alpha = 0.05$ is computed and shown in Fig. 6.

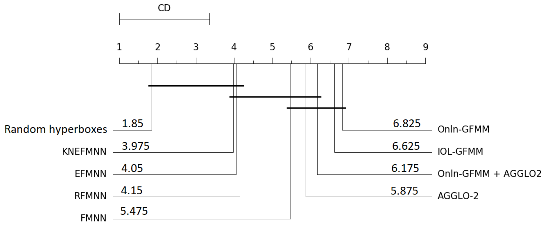


Fig. 6. Critical difference diagram for the performance of the RH classifier and other FMNNs ($\theta = 0.1$).

Similarly, with results of average ranks at $\theta = 0.7$, we can calculate the F-distribution value using the Friedman test $F_F = 14.0148 > F(8, 152) = 1.9998$. Therefore, there are significant differences among models using $\theta = 0.7$. By applying the Bonferroni-Dunn test, we can draw the CD diagram shown in Fig. 7.

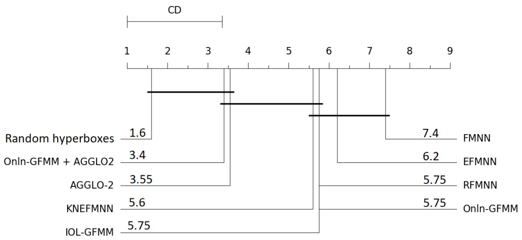


Fig. 7. Critical difference diagram for the performance of the RH classifier and other FMNNs ($\theta = 0.7$).

It can be seen that at the low value of θ, the RH classifier is significantly better than Onln-GFMM, IOL-GFMM, FMNN, AGGLO-2, and Onln-GFMM + AGGLO2 in terms of average weighted-F1 score. However, its performance still has no significant difference compared to EFMNN, KNEFMNN, and RFMNN, although the average ranking of RH classifier is lowest among nine fuzzy min-max models over 20 considered datasets. With a high value of θ, the RH model is significantly better than KNEFMNN, IOL-GFMM, Onln-GFMM, RFMNN, EFMNN, and FMNN. In this case, however, there is no statistical difference in the accuracy among the RH model, Onln-GFMM + AGGLO2 and AGGLO-2, although the performance of the RH classifier outperforms those of Onln-GFMM + AGGLO2 and AGGLO-2.

2) A Comparison of the Random Hyperboxes With Other Ensemble Classifiers:

This experiment compares the perfomance of the random hyperboxes classifier with other prevalent ensemble models including Random Forest [5], Rotation Forest [35], XGBoost [8], LightGBM [9], Gradient Boosting [7], and ensemble of base IOL-GFMM classifiers at the decision level (Ens-IOL-GFMM (DL)) and at the model level (Ens-IOL-GFMM (ML)) [11].

The average weighted-F1 scores of classifiers through 10 times repeated 4-fold cross-validation and their ranking are given in Tables X and XI in Appendix D-D3. Based on their average rank for 20 datasets, we can apply Friedman rank-sum test to calculate the F-distribution value $F_F = 4.7288 > F(7, 133) = 2.0791$. Therefore, there are differences in the performance of classifiers. Using the Bonferroni-Dunn test, we have the CD diagram of the RH model and other ensemble classifiers as Fig. 8.

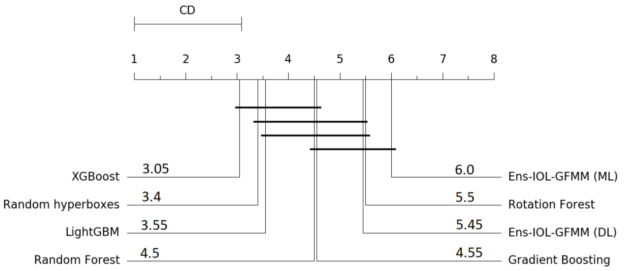


Fig. 8. Critical difference diagram for the performance of the RH classifier and other ensemble models.

Although the average rank of over 20 datasets of the RH model is higher than XGBoost, there are no significant differences in the accuracy values among XGBoost, LightGBM, Random Forest, and Gradient Boosting. In contrast, the RH classifier is statistically better than Rotation Forest and ensemble methods of IOL-GFMM base learners using full features on 20 considered datasets.

3) A Comparison of the Random Hyperboxes With Other Machine Learning Algorithms:

This experiment compares the RH classifier with other popular machine learning algorithms including Decision Tree [36], Naive Bayes [37], support vector machine (SVM) [38], K-nearest neighbors (KNN) [39], and Linear Discriminant Analysis (LDA) [40]. The experimental results of classifiers and their ranking are shown in Tables XII and XIII in Appendix D-D4.

Using Friedman rank-sum test, we get the F-distribution value $F_F = 4.4485 > F(5, 95) = 2.3102$. Hence, there are statistical differences in the performance of classifiers. Similarly, using the Bonferroni-Dunn test, we obtain the CD diagram in this case as Fig. 9.

Although the average rank of the RH classifier over 20 datasets is lowest among methods, there is no significant

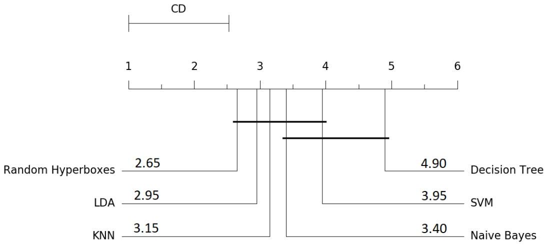


Fig. 9. Critical difference diagram for the performance of the RH classifier and other popular learning algorithms.

difference in the performance of the RH compared to LDA, KNN, SVM, and Naive Bayes. However, the RH classifier is much better than the decision tree.

V. On the Estimation of Generalization Error Bounds and Open Problems

The upper generalization error bound of the random hyperboxes model is computed based on the i.i.d. assumption of samples in both training and testing sets. However, in practice, this assumption is usually violated for the real world datasets. This means that it is very difficult to obtain the training and testing sets which are representatives of a true distribution of the sample space. In this section, we will estimate the upper generalization error bounds of datasets used for the experiments in section IV. The purpose of this section is to identify the effectiveness of the upper generalization error bound on real datasets and the existing problems when applying a strong assumption from the theoretical derivations to the practical issues. The upper bound values were estimated from the training set and 100 base learners trained by the IOL-GFMM algorithm with $\theta = 0.1$. The estimated results of the upper generalization error bound are the average values from 40 iterations (10 times repeated 4-fold cross-validation). To strengthen the comparison and conclusion, we also estimated the upper generalization error bounds from the base learners trained in turn on each of four folds generated by using the density preserving sampling (DPS) method [41]. The DPS method aims to preserve the data density and the classes shapes when splitting an original dataset into many folds, so it is possible to create the testing sets which are representatives for the training data. Hence, the testing errors on the DPS folds are usually smaller than those calculated from folds of the cross-validation method. This fact is confirmed with the results shown in Table I. This table presents the real average testing errors of 4-DPS fold cross-validation and 10 times repeated 4-fold cross-validation as well as their upper generalization error bounds estimated from corresponding training sets.

In general, we have ten datasets in which the estimated upper bounds are higher than real testing errors. Among them, there are a number of datasets with real errors close to the estimated upper bounds, such as $heart$, $pima\_diabetes$, $landsat\_satelite$, and $twonorm$. One explanation for these good estimations is that the training sets and testing sets are good representatives of each other and the whole sample space. It can be seen that, for these datasets, the real testing errors of 10 times repeated 4-fold cross-validation and 4-DPS fold cross-validation are relatively close to each other.

In the ten remaining datasets, the estimated values of upper bounds are much lower than the real testing errors when applying the 10 times repeated 4-fold cross-validation method. The same behavior but with a smaller error can be found with the 4-DPS fold cross-validation method on eight datasets. Interestingly, there are two datasets, $wovel$ and $movement\_libras$, in which the estimated values are very bad when using 10 times repeated 4-fold cross-validation, but we can obtain very good estimated upper bounds when deploying the 4-DPS fold cross-validation. This fact indicates that if the representativeness of training sets with regard to the whole sample space is good, we can achieve a much better estimation of the upper generalization error bounds which is close to the testing error on unseen data with the same distribution.

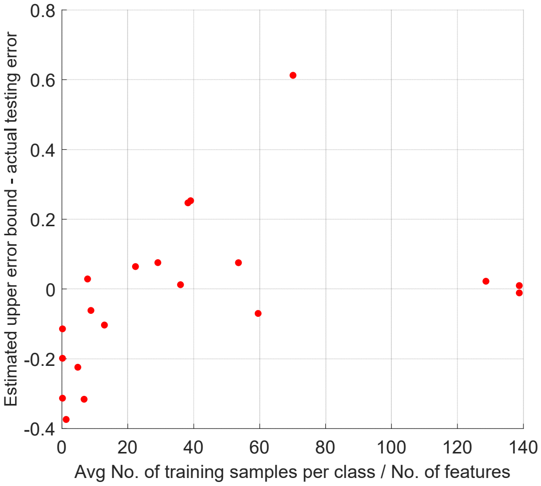


Fig. 10. The corelation of the difference in the estimated upper error bound and actual testing error with respect to the ratio of the average number of training samples per class and the number of features.

One general characteristic of datasets resulting in the poor estimated upper bounds is their sparsity with regard to a small number of samples and a relatively high number of dimensions. For these datasets, we do not have sufficient number of samples to accurately enough capture the underlying distribution of the whole sample space. As a result, the base estimators overfit with their training data, and the estimated values of the upper error bounds are usually small. Meanwhile, the testing errors on unseen data are fairly high. Here, one open problem identified is the relationship between the number of samples, classes, and dimensions so that we can obtain a good estimation of the generalization error bounds from the training data. This is a critical issue that needs to be tackled in future work. As an example demonstration for this issue, Fig. 10 shows the correlation of the difference in the estimated upper error bound and actual testing error to the ratio of the average training samples per class and the number of features for 20 datasets used in this experiment. We can see that a good estimation of the upper error bound can be obtained if the ratio of the average training samples per class and the number of features is larger than 20. If this ratio is higher than 120, it is

TABLE I
ESTIMATED UPPER GENERALIZATION ERROR BOUNDS, REAL TESTING ERROR, AND THEIR STANDARD DEVIATIONS COMPUTED FROM DIFFERENT ASSESSMENT METHODS

ID	**Dataset**	**10 times repeated 4-fold cross-validation**		**4-DPS fold cross-validation**	
		Testing error	**Estimated upper bound**	**Testing error**	**Estimated upper bound**
1	Balance_scale	0.225205 ± 0.08439	0.47831 ± 0.042218	0.113598 ± 0.010918	0.406031 ± 0.040286
2	banknote_authentication	0.001821 ± 0.001832	0.024073 ± 0.003918	0.001458 ± 0.002915	0.021189 ± 0.005162
3	blood_transfusion	0.269997 ± 0.041152	0.882506 ± 0.086337	0.215241 ± 0.014064	0.88731 ± 0.092977
4	breast_cancer_wisconsin	0.033258 ± 0.018076	0.108871 ± 0.016842	0.028604 ± 0.00805	0.109176 ± 0.0185528
5	BreastCancerCoimbra	0.323276 ± 0.088239	0.099026 ± 0.011883	0.241379 ± 0.116086	0.117675 ± 0.022436
6	connectionist_bench_sonar	0.443689 ± 0.107638	0.069858 ± 0.011676	0.125 ± 0.036824	0.073405 ± 0.009768
7	haberman	0.354808 ± 0.074583	0.601802 ± 0.064146	0.251581 ± 0.015168	0.529377 ± 0.071196
8	heart	0.170758 ± 0.024509	0.199591 ± 0.021899	0.174056 ± 0.024932	0.185977 ± 0.016658
9	movement_libras	0.415556 ± 0.097833	0.102645 ± 0.0173813	0.136111 ± 0.042913	0.147793 ± 0.022053
10	pima_diabetes	0.257552 ± 0.02993	0.269897 ± 0.023865	0.239583 ± 0.020395	0.260163 ± 0.019503
11	plant_species_leaves_margin	0.242875 ± 0.018245	0.128801 ± 0.010176	0.226875 ± 0.031516	0.118236 ± 0.008702
12	plant_species_leaves_shape	0.37025 ± 0.02875	0.171654 ± 0.007884	0.34125 ± 0.040337	0.184498 ± 0.005001
13	ringnorm	0.059649 ± 0.006073	0.048538 ± 0.002107	0.073514 ± 0.005635	0.05311 ± 0.001684
14	landsat_satelite	0.116943 ± 0.006915	0.181342 ± 0.015345	0.104273 ± 0.004502	0.181983 ± 0.019838
15	twonorm	0.027892 ± 0.003177	0.037681 ± 0.000687	0.029189 ± 0.002457	0.038032 ± 0.001046
16	vehicle_silhouettes	0.267981 ± 0.028846	0.206567 ± 0.013313	0.251716 ± 0.028308	0.205494 ± 0.008012
17	vertebral_column	0.229125 ± 0.0483	0.125849 ± 0.009933	0.209665 ± 0.051705	0.147310 ± 0.014369
18	vowel	0.363582 ± 0.06567	0.047757 ± 0.002076	0.023247 ± 0.011655	0.054667 ± 0.001788
19	waveform	0.158041 ± 0.006471	0.087998 ± 0.003282	0.1636 ± 0.006804	0.089598 ± 0.005328
20	wireless_indoor_localization	0.02275 ± 0.008566	0.098253 ± 0.014660	0.0155 ± 0.004435	0.089935 ± 0.003942

more likely to achieve an estimated upper error bound close to the actual testing error.

In summary, the i.i.d. assumption of training and testing sets is usually not met in practical datasets. Therefore, to reduce the classification error on unseen data, we need to use several methods to guarantee the representativeness of the training and testing sets when assessing the performance of models. Moreover, identification of the relationship between the numbers of samples, classes, and features is crucial to building a representative training set.

One of the strong points of the general fuzzy min-max neural network is the interpretability. However, the significantly improved predictive accuracy of the proposed random hyperboxes method comes at a price of loss of interpretability as is common with other ensemble methods. As previously shown in [11], hyperbox representation allows for combination at the model level rather than the decision level and therefore retaining the interpretability of the final model. Nonetheless, the combination of the individual hyperbox-based learners which are built from different random subspaces of features is not a trivial problem. Therefore, the future study should focus on building interpretable random hyperboxes models.

VI. CONCLUSION AND FUTURE WORK

This paper proposed a novel random hyperboxes classifier, discussed its properties and provided derivations of its generalization error bounds. The experimental results confirmed the efficiency of the proposed method in comparison to other single fuzzy min-max neural networks as well as single learning algorithms. The random hyperboxes model is also competitive with other popular ensemble methods. Furthermore, we provided several discussion on the estimation of the upper generalization error bounds for real-world datasets, and identified some open issues for future work.

There are still many opportunities for improvement of the proposed classifier. The relationship between correlation and variance between base learners as well as the trade-off between variance and bias of the random hyperboxes model need to be analyzed in more details. In addition, the influence of hyperparameters of the random hyperboxes model should be assessed by a comparative study. In this paper, we assumed that the strength $\mathcal{S} > 0$ when analyzing the generalization error bound. In the case of highly imbalanced classes, this assumption may be false because the strength usually focuses on the majority class. Therefore, the efficiency of the random hyperboxes classifier and its theoretical results should be investigated and extended for imbalanced datasets.

ACKNOWLEDGMENT

T.T. Khuat acknowledges FEIT-UTS for awarding his PhD scholarships (IRS and FEIT scholarships).

APPENDIX A
PROOF OF LEMMA 1

This section provides the readers with the proof of Lemma 1 in the main paper.

Lemma 3. *Given m identically distributed random variables (not necessarily independent) with the variance of each variable σ^2 and positive pairwise correlation ρ, the variance of the average random variable is:*

$$\rho \cdot \sigma^2 + \frac{1-\rho}{m} \cdot \sigma^2$$

Proof. Supposing that $\Phi = (\Phi_1, \ldots, \Phi_m)$ is a set of m random variables with given covariances $\sigma_{ij} = \texttt{Cov}(\Phi_i, \Phi_j)$, we need to find variance of an average variable $\mathcal{L}(\Phi_1, \ldots, \Phi_m)$ obtained as a linear combination of m random variables, i.e.,

$$\mathcal{L}(\Phi_1, \ldots, \Phi_m) = \sum_{i=1}^{m} (\lambda_i \cdot \Phi_i)$$

We can rewrite this formula in a compact way using matrix and vector notations as follows:

$$\mathcal{L}(\Phi) = \mathbf{\Lambda}^T \cdot \Phi$$

where $\mathbf{\Lambda}^T = (\lambda_1, \ldots, \lambda_m)$. And then, we have the expected value:

$$\mathbb{E}(\mathcal{L}(\Phi)) = \mathbb{E}(\mathbf{\Lambda}^T \cdot \Phi) = \mathbf{\Lambda}^T \cdot \mathbb{E}(\Phi)$$

and the variance:

$$\begin{aligned}
\texttt{Var}(\mathcal{L}(\Phi)) &= \mathbb{E}(\mathcal{L}^2(\Phi)) - [\mathbb{E}(\mathcal{L}(\Phi))]^2 \\
&= \mathbb{E}(\mathbf{\Lambda}^T \Phi \Phi^T \mathbf{\Lambda}) - \mathbb{E}(\mathbf{\Lambda}^T \Phi)[\mathbb{E}(\mathbf{\Lambda}^T \Phi)]^T \\
&= \mathbf{\Lambda}^T \mathbb{E}(\Phi \Phi^T) \mathbf{\Lambda} - \mathbf{\Lambda}^T \mathbb{E}(\Phi)(\mathbb{E}(\Phi))^T \mathbf{\Lambda} \\
&= \mathbf{\Lambda}^T [\mathbb{E}(\Phi \Phi^T) - \mathbb{E}(\Phi)(\mathbb{E}(\Phi))^T] \mathbf{\Lambda} \\
&= \mathbf{\Lambda}^T \texttt{Cov}(\Phi) \mathbf{\Lambda} \\
&= \mathbf{\Lambda}^T \Sigma \mathbf{\Lambda}
\end{aligned}$$

where $\Sigma = (\sigma_{ij})$ is the covariance of Φ

In this lemma, $\sigma_{ij} = \rho \cdot \sigma^2$ when $i \neq j$. We also have $\sigma_{ii} = \texttt{Cov}(\Phi_i, \Phi_i) = \sigma^2 = [\rho + (1-\rho)]\sigma^2$. Hence, we may decompose the covariance matrix Σ into the sum of two matrices, i.e., one includes ρ in every entry and the other includes $(1-\rho)$ on the main diagonal and zeros for the rest. Formally, we achieve:

$$\Sigma = \sigma^2 [\rho \mathbf{1}_m \mathbf{1}_m^T + (1-\rho)\mathbf{I}_m]$$

where $\mathbf{1}_m$ is a column vector containing m 1's and $\mathbf{I}_m$ is an identity matrix with size $m \times m$. Then we get:

$$\begin{aligned}
\texttt{Var}(\mathcal{L}(\Phi)) &= \mathbf{\Lambda}^T \sigma^2 [\rho \mathbf{1}_m \mathbf{1}_m^T + (1-\rho)\mathbf{I}_m] \mathbf{\Lambda} \\
&= (\mathbf{\Lambda}^T \mathbf{1}_m \mathbf{1}_m^T \mathbf{\Lambda}) \rho \sigma^2 + (\mathbf{\Lambda}^T \mathbf{I}_m \mathbf{\Lambda})(1-\rho)\sigma^2
\end{aligned}$$

For $\mathbf{\Lambda}^T = (1/m, \ldots, 1/m)$, we get:

$$\mathbf{\Lambda}^T \mathbf{1}_m \mathbf{1}_m^T \mathbf{\Lambda} = (\mathbf{\Lambda}^T \mathbf{1}_m)^2 = (m \cdot 1/m)^2 = 1$$

and

$$\mathbf{\Lambda}^T \mathbf{I}_m \mathbf{\Lambda} = 1/m^2 + \ldots + 1/m^2 = m \cdot 1/m^2 = 1/m$$

Therefore,

$$\texttt{Var}(\mathcal{L}(\Phi)) = \rho \sigma^2 + \frac{1-\rho}{m} \sigma^2$$

The lemma is proved. □

Appendix B
Proof of Lemma 2

This section provides the proof of Lemma 2 in the main paper.

Lemma 4. *When the number of base estimators increases* $(m \to \infty)$ *and base estimators are independent, for almost surely all i.i.d. random vectors* $\Phi_1, \Phi_2, \ldots$*, the margin function for a random hyperboxes model* $\mathcal{M}(\mathbf{x}, c)$ *at each input* $\mathbf{x}$ *converges to:*

$$\mathcal{M}^*(\mathbf{x}, c) = \mathbf{P}_\Phi(h(\mathbf{x}, \Phi) = c) - \max_{j \neq c} \mathbf{P}_\Phi(h(\mathbf{x}, \Phi) = j)$$

Proof. We have the margin function of the random hyperboxes model with m base learners at each input sample $\mathbf{x}$ as follows:

$$\mathcal{M}(\mathbf{x}, c) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}(h_i(\mathbf{x}) = c) - \max_{j \neq c} \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}(h_i(\mathbf{x}) = j)$$

For random vectors $\Phi_1, \Phi_2, \ldots$ and for all input vectors $\mathbf{x}$, to prove Lemma 2, it suffices to show

$$\frac{1}{m}\sum_{i=1}^{m} \mathbb{1}(h_i(\mathbf{x}) = j) \xrightarrow{m \to \infty} \mathbf{P}_\Phi(h(\mathbf{x}, \Phi) = j)$$

where $h_i(\mathbf{x}) \equiv h(\mathbf{x}, \Phi_i)$, and $\mathbb{1}(\cdot)$ is an indicator function.

For each hyperbox-based learner, $h(\mathbf{x}, \Phi_i) = j$ is union of hyerboxes with class j and their neighborhood regions which generate the maximum membership value from these hyperboxes to an input $\mathbf{x}$ in comparison to hyperboxes representing other classes. Assuming a finite number of random vectors Φ (the finite number of sample subsets and finite number of feature subsets) from which any hyperbox-based learner $h(\mathbf{x}, \Phi_i)$ $(\Phi_i \subset \Phi)$ is constructed, then there exists a finite number K of such unions of hyperboxes and neighbourhood regions, called $S_1, \ldots, S_K$.

Let define:

$$\varphi(\Phi) = k \text{ if } \{\mathbf{x} : h(\mathbf{x}, \Phi) = j\} = S_k$$

Let N_k be the number of times that $\varphi(\Phi_i) = k$ in the first m trials, then we obtain:

$$\frac{1}{m}\sum_{i=1}^{m} \mathbb{1}(h(\mathbf{x}, \Phi_i) = j) = \frac{1}{m}\sum_{k} N_k \mathbb{1}(\mathbf{x} \in S_k)$$

According to the strong law of large numbers when m increases,

$$N_k = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}(\varphi(\Phi_i) = k)$$

converges almost surely (a.s.) with probability 1 to

$$\mathbb{E}_\Phi[\mathbb{1}(\varphi(\Phi) = k)] = \mathbf{P}_\Phi(\varphi(\Phi) = k)$$

Therefore,

$$\begin{aligned} \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}(h(\mathbf{x}, \Phi_i) = j) &\xrightarrow{a.s.} \sum_{k} \mathbf{P}_\Phi(\varphi(\Phi) = k)\mathbb{1}(\mathbf{x} \in S_k) \\ &= \mathbf{P}_\Phi(h(\mathbf{x}, \Phi) = j) \end{aligned}$$

The lemma is proved. □

Appendix C
Proof of Theorem 2

This section shows the proof for Theorem 2 from the main paper.

Theorem 2. *An upper bound of the generalization error for the random hyperboxes model can be estimated from the strength of base learners and correlation between base learners as follows:*

$$\mathcal{E}^* \leq \overline{\rho}\left(\frac{1}{\mathcal{S}^2} - 1\right)$$

Proof. From lemma 2, we have:

$$\mathcal{M}^*(\mathbf{x}, c) = \mathbf{P}_\Phi(h(\mathbf{x}, \Phi) = c) - \max_{j \neq c} \mathbf{P}_\Phi(h(\mathbf{x}, \Phi) = j)$$

With the assumption of the strength $\mathcal{S} = \mathbb{E}_{\mathbf{X},\mathcal{C}}\mathcal{M}^*(\mathbf{x}, c) > 0$, according to Chebyshev's inequality, we have:

$$\begin{aligned}\mathcal{E}^* &= \mathbf{P}_{\mathbf{X},\mathcal{C}}\left[\mathcal{M}^*(\mathbf{x}, c) < 0\right] \leq \mathbf{P}_{\mathbf{X},\mathcal{C}}\left[\mathcal{S} - \mathcal{M}^*(\mathbf{x}, c) \geq \mathcal{S}\right] \\ &= \mathbf{P}_{\mathbf{X},\mathcal{C}}\left[|\mathcal{M}^*(\mathbf{x}, c) - \mathcal{S}| \geq \mathcal{S}\right] \leq \frac{\mathtt{Var}_{\mathbf{X},\mathcal{C}}(\mathcal{M}^*(\mathbf{x}, c))}{\mathcal{S}^2}\end{aligned}$$

For any function f and two i.i.d. random variables Φ and Φ', we have:

$$\mathbb{E}_\Phi[f(\Phi)]^2 = \mathbb{E}_{\Phi,\Phi'}[f(\Phi)f(\Phi')]$$

In the main paper, we get $\mathcal{M}^*(\mathbf{x}, c) = \mathbb{E}_\Phi\mathcal{R}(\Phi)$, thus

$$[\mathcal{M}^*(\mathbf{x}, c)]^2 = \mathbb{E}_\Phi\mathcal{R}(\Phi)^2 = \mathbb{E}_{\Phi,\Phi'}[\mathcal{R}(\Phi)\mathcal{R}(\Phi')]$$

Now, we can compute $\mathtt{Var}_{\mathbf{X},\mathcal{C}}(\mathcal{M}^*(\mathbf{x}, c))$ as follows:

$$\begin{aligned}\mathtt{Var}_{\mathbf{X},\mathcal{C}}(\mathcal{M}^*(\mathbf{x}, c)) &= \mathbb{E}_{\mathbf{X},\mathcal{C}}([\mathcal{M}^*(\mathbf{x}, c)]^2) - \Big[\mathbb{E}_{\mathbf{X},\mathcal{C}}(\mathcal{M}^*(\mathbf{x}, c))\Big]^2 \\ &= \mathbb{E}_{\mathbf{X},\mathcal{C}}\Big[\mathbb{E}_{\Phi,\Phi'}[\mathcal{R}(\Phi)\mathcal{R}(\Phi')]\Big] - \Big[\mathbb{E}_{\mathbf{X},\mathcal{C}}(\mathbb{E}_\Phi\mathcal{R}(\Phi))\Big]^2 \\ &= \mathbb{E}_{\Phi,\Phi'}\Big[\mathbb{E}_{\mathbf{X},\mathcal{C}}[\mathcal{R}(\Phi)\mathcal{R}(\Phi')]\Big] - \Big[\mathbb{E}_\Phi(\mathbb{E}_{\mathbf{X},\mathcal{C}}\mathcal{R}(\Phi))\Big]^2 \\ &= \mathbb{E}_{\Phi,\Phi'}\Big[\mathbb{E}_{\mathbf{X},\mathcal{C}}[\mathcal{R}(\Phi)\mathcal{R}(\Phi')]\Big] - \mathbb{E}_{\Phi,\Phi'}\Big[\mathbb{E}_{\mathbf{X},\mathcal{C}}\mathcal{R}(\Phi)\mathbb{E}_{\mathbf{X},\mathcal{C}}\mathcal{R}(\Phi')\Big] \\ &= \mathbb{E}_{\Phi,\Phi'}\Big[\mathbb{E}_{\mathbf{X},\mathcal{C}}[\mathcal{R}(\Phi)\mathcal{R}(\Phi')] - \mathbb{E}_{\mathbf{X},\mathcal{C}}\mathcal{R}(\Phi)\mathbb{E}_{\mathbf{X},\mathcal{C}}\mathcal{R}(\Phi')\Big] \\ &= \mathbb{E}_{\Phi,\Phi'}\Big[\mathtt{Cov}_{\mathbf{X},\mathcal{C}}(\mathcal{R}(\Phi)\mathcal{R}(\Phi'))\Big] \\ &= \mathbb{E}_{\Phi,\Phi'}\Big[\rho_{\mathbf{X},\mathcal{C}}(\Phi, \Phi')\sigma_{\mathbf{X},\mathcal{C}}(\mathcal{R}(\Phi))\sigma_{\mathbf{X},\mathcal{C}}(\mathcal{R}(\Phi'))\Big] \\ &= \overline{\rho}\Big[\mathbb{E}_\Phi(\sigma_{\mathbf{X},\mathcal{C}}(\mathcal{R}(\Phi)))\Big]^2\end{aligned}$$

where $\overline{\rho} = \mathbb{E}_{\Phi,\Phi'}[\rho_{\mathbf{X},\mathcal{C}}(\Phi, \Phi')]$

For any random variable $\mathbf{Z}$, $\mathtt{Var}(\mathbf{Z}) \geq 0 \Rightarrow \mathbb{E}(\mathbf{Z}^2) - \mathbb{E}(\mathbf{Z})^2 \geq 0 \Rightarrow \mathbb{E}(\mathbf{Z})^2 \leq \mathbb{E}(\mathbf{Z}^2)$. Therefore,

$$\mathtt{Var}_{\mathbf{X},\mathcal{C}}(\mathcal{M}^*(\mathbf{x}, c)) = \overline{\rho}\Big[\mathbb{E}_\Phi(\sigma_{\mathbf{X},\mathcal{C}}(\mathcal{R}(\Phi)))\Big]^2 \leq \overline{\rho}\ \mathbb{E}_\Phi(\sigma_{\mathbf{X},\mathcal{C}}(\mathcal{R}(\Phi))^2) = \overline{\rho}\ \mathbb{E}_\Phi(\mathtt{Var}_{\mathbf{X},\mathcal{C}}(\mathcal{R}(\Phi)))$$

In addition, using the definition of the variance for a random variable and inequality $\mathbb{E}(\mathbf{Z})^2 \leq \mathbb{E}(\mathbf{Z}^2)$, we can write:

$$\begin{aligned}\mathbb{E}_\Phi(\mathtt{Var}_{\mathbf{X},\mathcal{C}}(\mathcal{R}(\Phi))) &= \mathbb{E}_\Phi\Big[\mathbb{E}_{\mathbf{X},\mathcal{C}}[\mathcal{R}(\Phi)^2] - \mathbb{E}_{\mathbf{X},\mathcal{C}}[\mathcal{R}(\Phi)]^2\Big] \\ &= \mathbb{E}_\Phi\Big[\mathbb{E}_{\mathbf{X},\mathcal{C}}[\mathcal{R}(\Phi)^2]\Big] - \mathbb{E}_\Phi\Big[[\mathbb{E}_{\mathbf{X},\mathcal{C}}\mathcal{R}(\Phi)]^2\Big] \\ &\leq \mathbb{E}_\Phi\Big[\mathbb{E}_{\mathbf{X},\mathcal{C}}[\mathcal{R}(\Phi)^2]\Big] - \Big[\mathbb{E}_\Phi(\mathbb{E}_{\mathbf{X},\mathcal{C}}[\mathcal{R}(\Phi)])\Big]^2 \\ &= \mathbb{E}_\Phi\Big[\mathbb{E}_{\mathbf{X},\mathcal{C}}[\mathcal{R}(\Phi)^2]\Big] - \Big[\mathbb{E}_{\mathbf{X},\mathcal{C}}(\mathbb{E}_\Phi[\mathcal{R}(\Phi)])\Big]^2 \\ &= \mathbb{E}_\Phi\Big[\mathbb{E}_{\mathbf{X},\mathcal{C}}[\mathcal{R}(\Phi)^2]\Big] - \Big[\mathbb{E}_{\mathbf{X},\mathcal{C}}\mathcal{M}^*(\mathbf{x}, c)\Big]^2 \\ &\leq 1 - \mathcal{S}^2\end{aligned}$$

due to $\mathcal{R}(\Phi) \leq 1$ and $\mathcal{S} = \mathbb{E}_{\mathbf{X},\mathcal{C}}\mathcal{M}^*(\mathbf{x}, c)$. As a result,

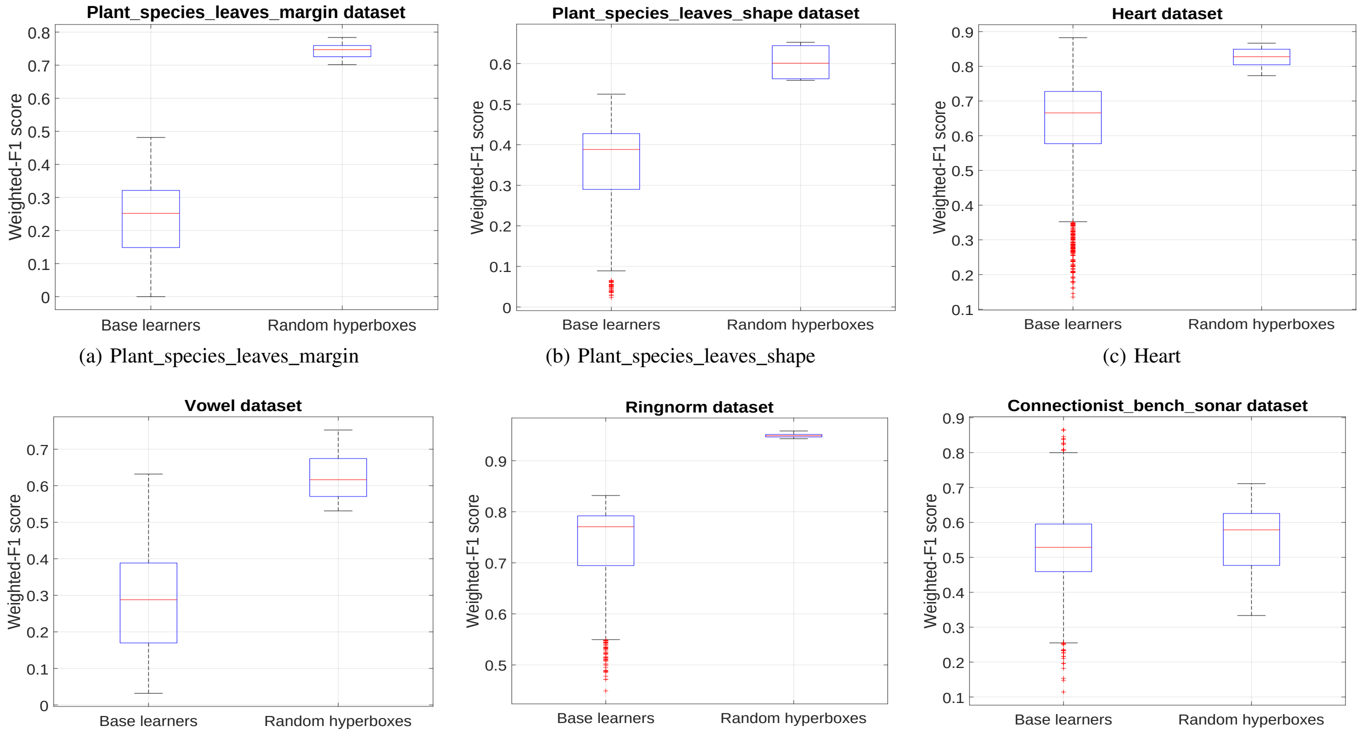
$$\mathcal{E}^* \leq \frac{\mathtt{Var}_{\mathbf{X},\mathcal{C}}(\mathcal{M}^*(\mathbf{x}, c))}{\mathcal{S}^2} \leq \frac{\overline{\rho}\ \mathbb{E}_\Phi(\mathtt{Var}_{\mathbf{X},\mathcal{C}}(\mathcal{R}(\Phi)))}{\mathcal{S}^2} \leq \frac{\overline{\rho}\ (1 - \mathcal{S}^2)}{\mathcal{S}^2} = \overline{\rho}\left(\frac{1}{\mathcal{S}^2} - 1\right)$$

The theorem is proved. □

Appendix D
Additional Experimental Results

A. Supplementary Part for Analyzing the Variance of the Random Hyperboxes Classifier

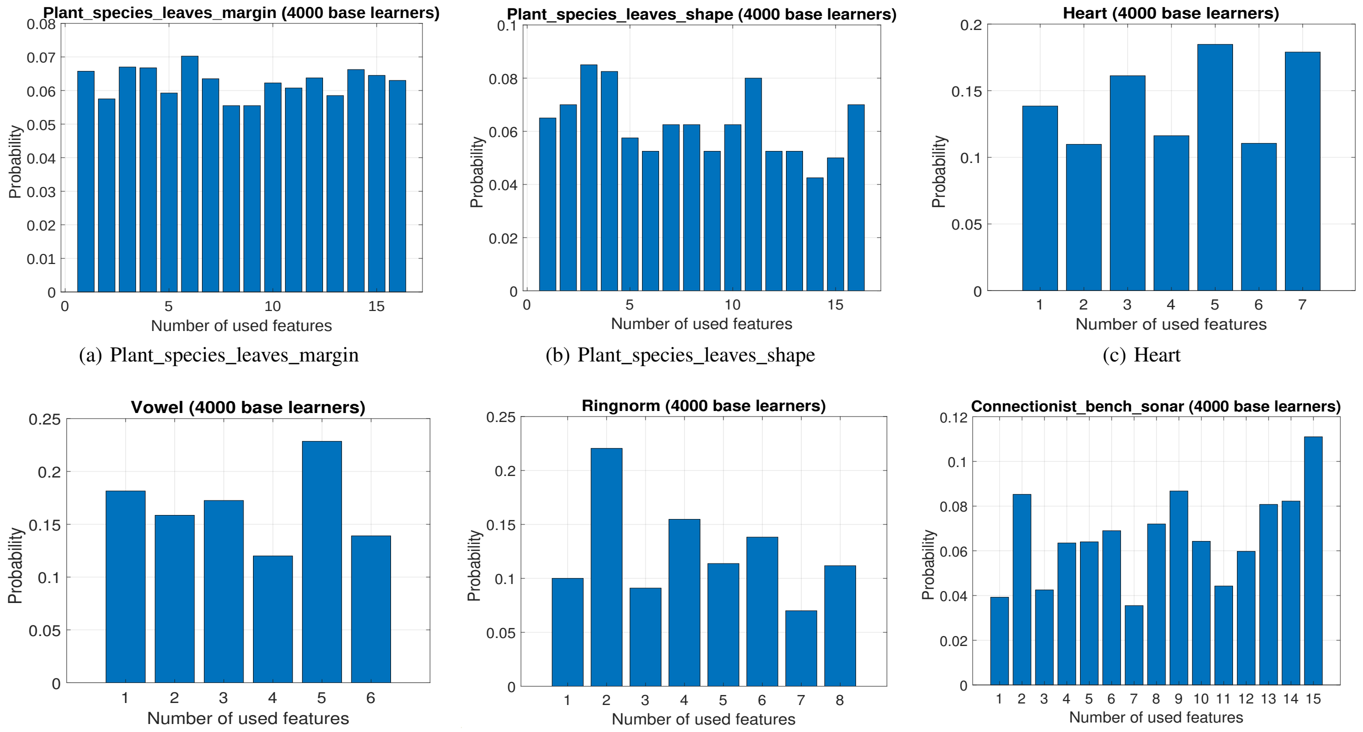
This part provides some supplementary figures for subsection IV.A.1 from the main paper. This experiment was performed on six datasets with diversity in the numbers of samples, features, and classes, i.e., *plant_species_leaves_margin*, *plant_species_leaves_shape*, *heart*, *vowel*, *ringnorm*, and *connectionist_bench_sonar*. Fig. 11 shows the variance values in terms of weighted-F1 scores using the 10 times repeated 4-fold cross-validation of base classifiers and the random hyperboxes



(a) Plant_species_leaves_margin (b) Plant_species_leaves_shape (c) Heart

(d) Vowel (e) Ringnorm (f) Connectionist_bench_sonar

Fig. 11. The variances of the random hyperboxes models and their base learners for different datasets.



(a) Plant_species_leaves_margin (b) Plant_species_leaves_shape (c) Heart

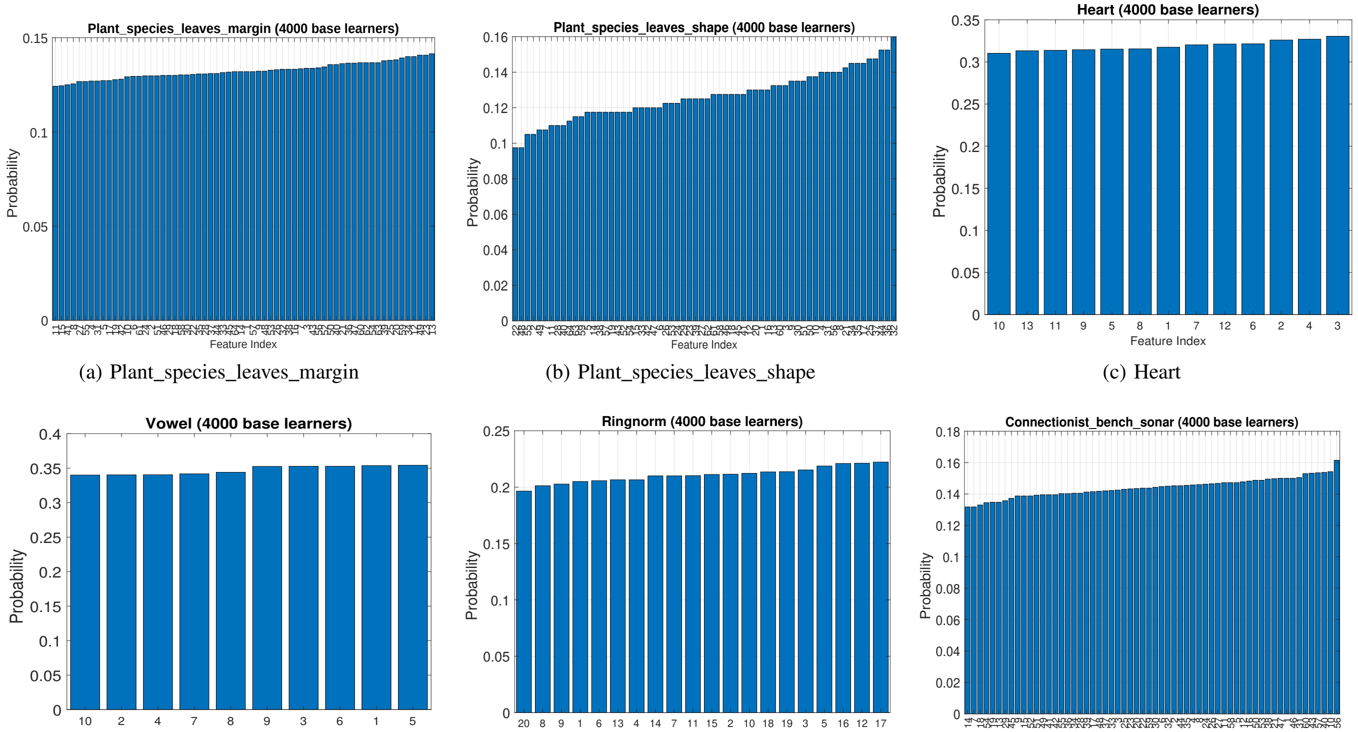
(d) Vowel (e) Ringnorm (f) Connectionist_bench_sonar

Fig. 12. The probability of the number of used features for all base learners over different datasets.

models over different datasets. These results confirm that the random hyperboxes model is able to reduce the variance in its base learners, and so it can achieve better performance than its base models.

Fig. 12 shows the probability of the number of features, d, used to build the 4000 base learners for the experiment shown in subsection IV.A.1 from the main paper. It can be observed that the probability distribution of the number of used features is nearly uniform in all 4000 base learners.

We can also identify the used probability of each feature over 4000 base learners to find the importance scores of features with respect to the performance of the ensemble model. This information is given in Fig. 13. From the importance scores of features, we built a single model using top-K of the most important features to assess the performance of the random hyperboxes and the use of single models. We can observe that in many datasets, the single model often achieves better performance when it is trained on more features. However, in several cases such as in *ringnorm* and *connectionist_bench_sonar* datasets, the best performance of the single model is obtained if it is trained on a subset of the most important features. From Figs. 11 and 14, it is easily seen that the random hyperboxes model trained using a subset of features usually achieves higher classification accuracy than the single model trained on the same dataset using all of the available features.



(a) Plant_species_leaves_margin

(b) Plant_species_leaves_shape

(c) Heart

(d) Vowel

(e) Ringnorm

(f) Connectionist_bench_sonar

Fig. 13. The probability of each feature used for all base learners over different datasets.

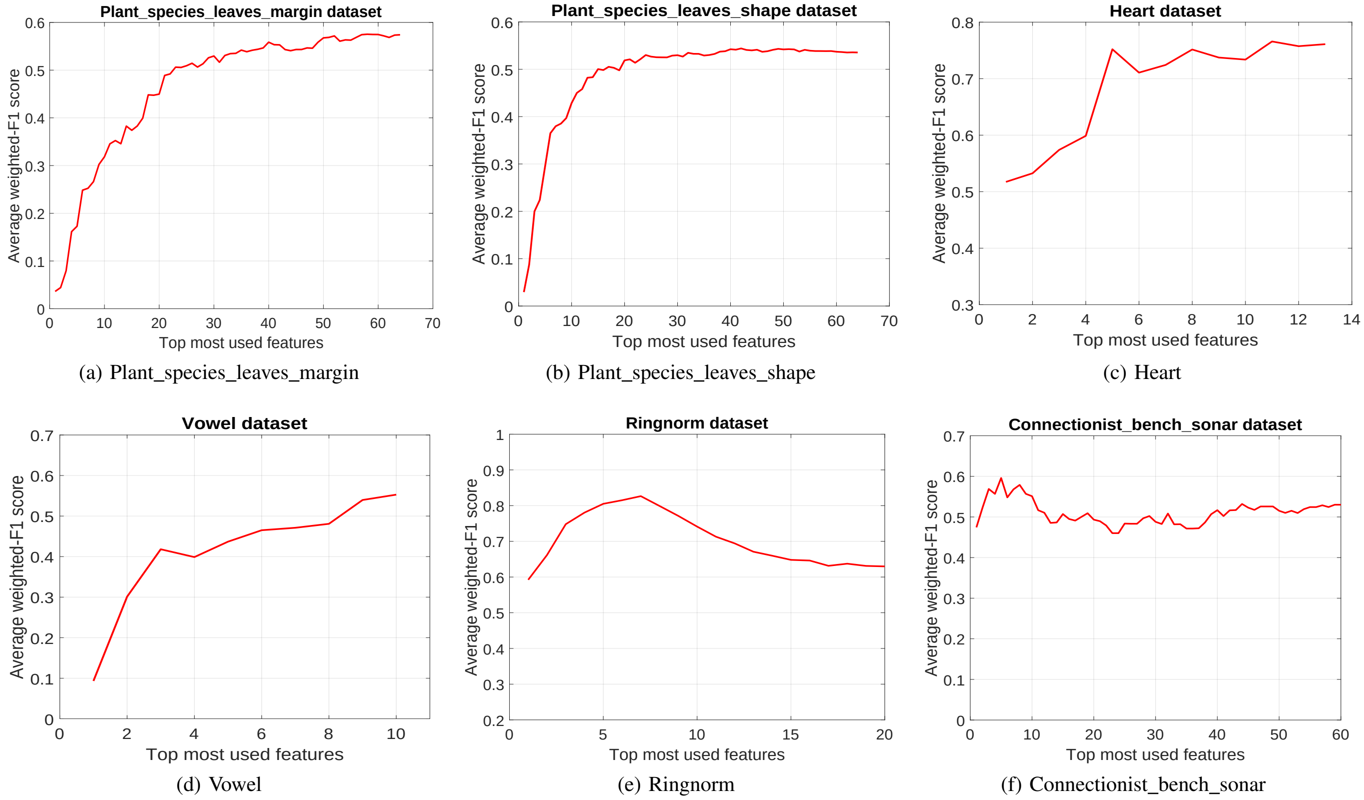
B. Analyzing the Effectiveness of the Random Hyperboxes on High Dimensional Data

When building predictive models for problems with very high dimensional data, the performance of models is negatively influenced by the redundancy of features. This problem is known as the Curse of Dimensionality [42]. This experiment is to assess the robustness of the random hyperboxes classifier for high dimensional data in comparison to the single IOL-GFMM model. We used two very high dimensional dataset, i.e., *PEMS database* [43] and *Complex Hydraulic System* [44]. 80% of samples in each dataset were used as training data and the remaining 20% of samples were testing data. The summaries of these datasets are shown in Table II.

TABLE II
SUMMARIZE INFORMATION OF HIGH DIMENSIONAL DATASETS

Dataset	**#samples**	**#features**	**#classes**	**#training**	**#testing**
PEMS database	440	138 672	7	352	88
Complex Hydraulic System	2205	43 680	2	1764	441

In this experiment, each base learner in the random hyperboxes model is trained on 50% of samples randomly selected from the training data. The maximum number of used features for each base learner is set to $2\sqrt{p}$, where p is the number



(a) Plant_species_leaves_margin (b) Plant_species_leaves_shape (c) Heart

(d) Vowel (e) Ringnorm (f) Connectionist_bench_sonar

Fig. 14. Average weighted-F1 scores over 40 testing folds of a single model using training sets with top-k most used features over different datasets.

of dimensions of the dataset. The number of base learners for each random hyperboxes model is $m = 100$. The weighted-F1 scores of the random hyperboxes and single IOL-GFMM model through different values of θ are given in Fig. 15 for the *PEMS database* dataset and in Fig. 16 for the *Complex Hydraulic System* dataset.

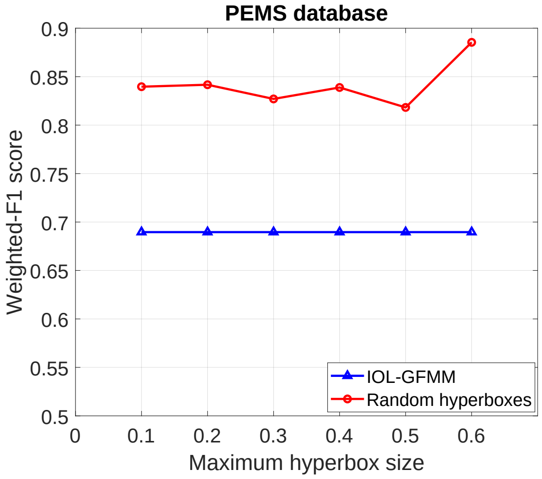


Fig. 15. Weighted-F1 score of the random hyperboxes and IOL-GFMM for the *PEMS database* dataset

It can be observed that the IOL-GFMM has consistently lower performance than RH with the very high dimensional data. In contrast, the random hyperboxes can achieve high accuracy using only $2\sqrt{p}$ random features at most for each base learner. The diversity in the base learners and the use of a low number of features allow the random hyperboxes to obtain better performance across the maximum hyperbox size values. Because each base learner in the random hyperboxes model uses a much smaller number of features compared to the IOL-GFMM model trained using all features, training time and testing time of the random hyperboxes is faster than that of the IOL-GFMM model. The training and testing time of each classifier is given in Tables III and IV. Fast training and testing time along with better accuracy confirm the efficiency of the ensemble model

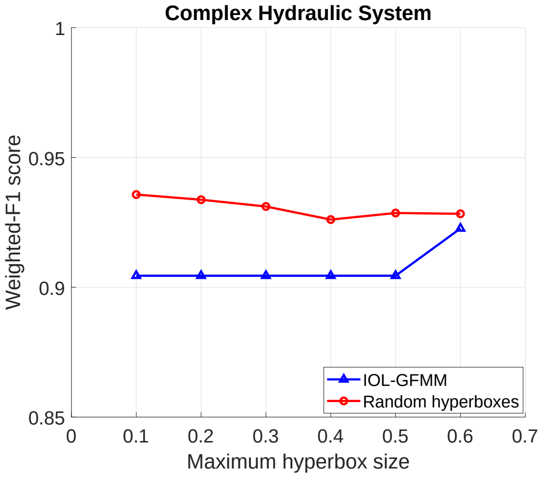


Fig. 16. Weighted-F1 score of the random hyperboxes and IOL-GFMM for the *Complex Hydraulic System* dataset

in comparison to the single model using the same learning algorithm.

TABLE III
TRAINING TIME (S) OF THE IOL-GFMM AND RANDOM HYPERBOXES MODEL ON THE HIGH DIMENSIONAL DATASETS

Dataset	**Algorithm**	$\theta = 0.1$	$\theta = 0.2$	$\theta = 0.3$	$\theta = 0.4$	$\theta = 0.5$	$\theta = 0.6$
PEMS database	IOL-GFMM	51.3784	56.5849	52.6432	52.12905	56.7359	57.1392
	Random hyperboxes	26.2364	26.4292	27.0474	27.3853	29.3139	28.7593
Complex Hydraulic System	IOL-GFMM	2093.5169	2235.3104	2045.8519	1914.7439	1987.5575	1785.5609
	Random hyperboxes	154.9104	125.8966	100.0234	84.0987	75.5298	66.7039

TABLE IV
TESTING TIME (S) OF THE IOL-GFMM AND RANDOM HYPERBOXES MODEL ON THE HIGH DIMENSIONAL DATASETS

Dataset	**Algorithm**	$\theta = 0.1$	$\theta = 0.2$	$\theta = 0.3$	$\theta = 0.4$	$\theta = 0.5$	$\theta = 0.6$
PEMS database	IOL-GFMM	121.2674	126.1965	121.4517	122.0169	126.4106	126.3136
	Random hyperboxes	11.3272	11.3308	12.8158	11.6774	12.3205	10.8228
Complex Hydraulic System	IOL-GFMM	1440.4623	1506.2449	1467.3662	1357.6034	1277.8380	1083.6029
	Random hyperboxes	118.4271	69.8559	44.8562	29.9445	23.3218	17.1749

C. Supplementary Part for Analyzing the Roles of the Number of Base Learners and Maximum Number of Used Features in the Random Hyperboxes models

This part provides some supplementary figures for subsection IV.A.2 from the main paper. This experiment was performed on eight different datasets with diversity in the numbers of samples, features, and classes, i.e., *plant_species_leaves_margin*, *plant_species_leaves_shape*, *movement_libras*, *connectionist_bench_sonar*, *vehicle_sihouettes*, *breast_cancer_wisconsin*, *heart*, and *vowel*. The purpose of this experiment is to study the impacts of the number of base learners and the maximum number of used features on the classification performance of the random hyperboxes model.

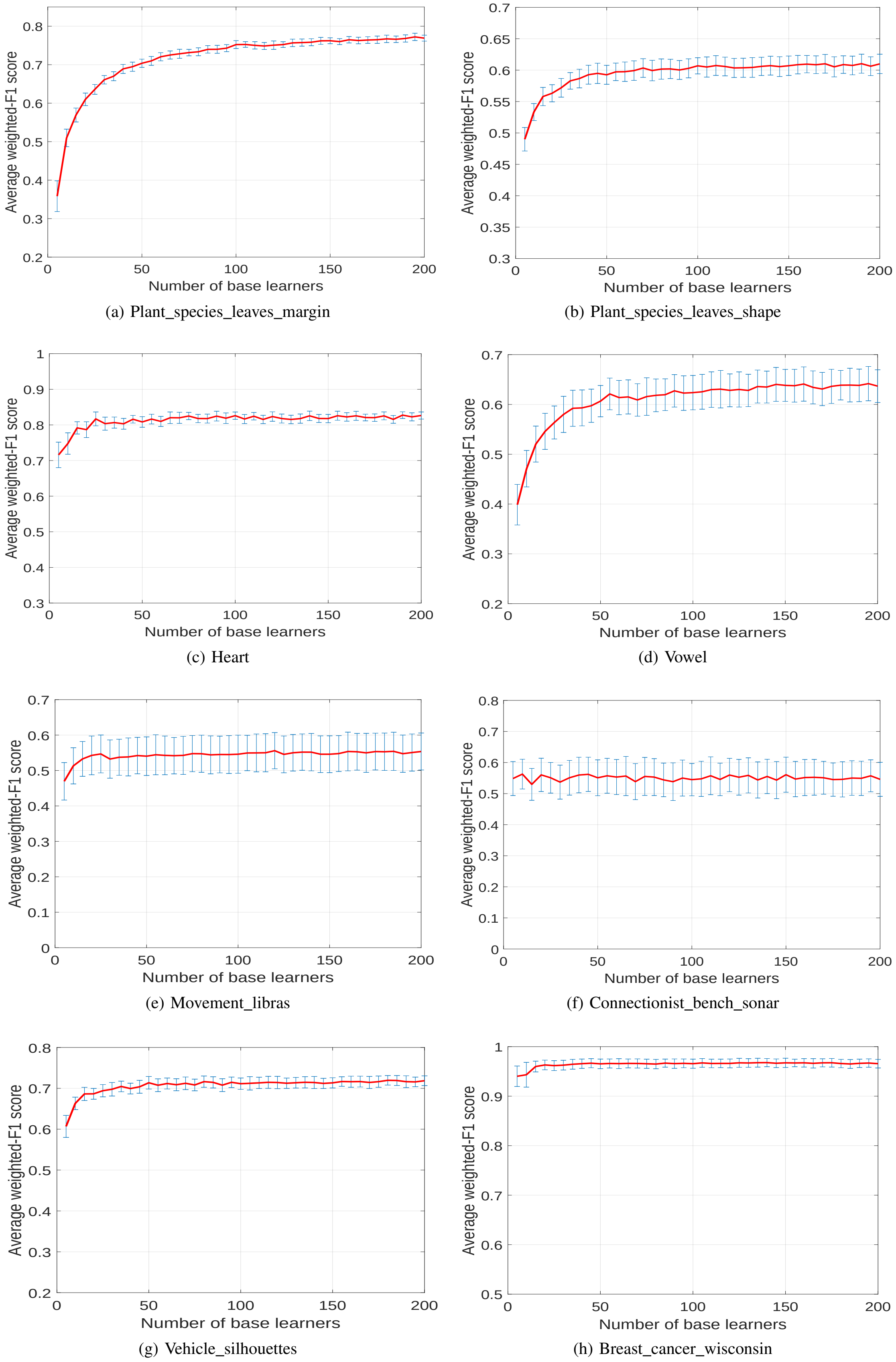
Fig. 17 shows the change in the average weighted-F1 score when we increase the number of base estimators. We can observe a general trend over all experimental datasets which is that the increase in the number of base learners does not lead to the decrease in the classification accuracy. These empirical results are consistent with the statements in the theoretical part (section III.C.1) from the main paper.

Fig. 18 presents the change in the classification performance when the maximum number of used features increases. A general trend can be observed in which the classification accuracy only increases up to a certain value of the maximum number of used features, and then decreases if the maximum number of features available for the base classifiers is increased. The reason for this trend is explained by the correlation between base learners as shown in subsection IV.A.2 from the main paper.

D. Comparing the Performance of the Random Hyperboxes to Other Classifiers

1) Datasets and Parameter Settings:

In this paper, we used 20 datasets with diversity in the numbers of samples, features, and classes taken from the UCI repository [45]. Table V summarizes the information of these datasets. Each dataset is normalized to the range of [0, 1]



(a) Plant_species_leaves_margin

(b) Plant_species_leaves_shape

(c) Heart

(d) Vowel

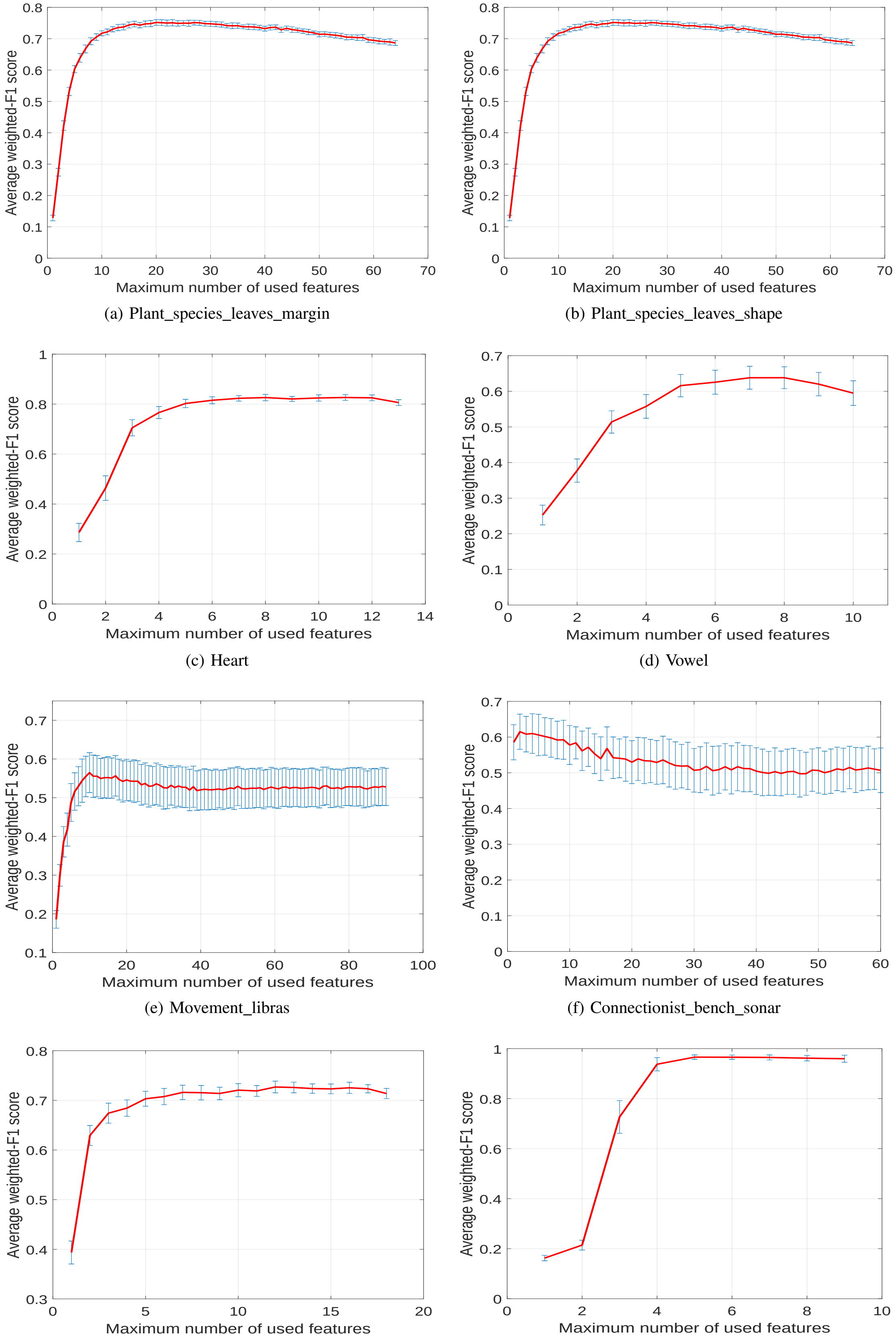
(e) Movement_libras

(f) Connectionist_bench_sonar

(g) Vehicle_silhouettes

(h) Breast_cancer_wisconsin

Fig. 17. The change in the average weighted-F1 scores when increasing the number of base learners for different datasets.



(a) Plant_species_leaves_margin

(b) Plant_species_leaves_shape

(c) Heart

(d) Vowel

(e) Movement_libras

(f) Connectionist_bench_sonar

(g) Vehicle_silhouettes

(h) Breast_cancer_wisconsin

Fig. 18. The change in the average weighted-F1 scores when increasing the maximum number of used dimensions for different datasets.

according to the requirement of the fuzzy min-max neural networks. The experiments were executed on the computer using Red Hat Enterprise Linux 7.5 with Intel Xeon Gold 6150 2.7GHz CPU and 64GB RAM.

TABLE V
THE DESCRIPTIONS OF THE USED DATASETS

ID	Dataset	# samples	# features	# classes
1	Balance_scale	625	4	3
2	banknote_authentication	1372	4	2
3	blood_transfusion	748	4	2
4	breast_cancer_wisconsin	699	9	2
5	BreastCancerCoimbra	116	9	2
6	connectionist_bench_sonar	208	60	2
7	haberman	306	3	2
8	heart	270	13	2
9	movement_libras	360	90	15
10	pima_diabetes	768	8	2
11	plant_species_leaves_margin	1600	64	100
12	plant_species_leaves_shape	1600	64	100
13	ringnorm	7400	20	2
14	landsat_satellite	6435	36	6
15	twonorm	7400	20	2
16	vehicle_silhouettes	846	18	4
17	vertebral_column	310	6	3
18	vowel	990	10	11
19	waveform	5000	21	3
20	wireless_indoor_localization	2000	7	4

For experiments, the maximum hyperbox size of based learners in the random hyperboxes model, as well as different types of FMNNs, is set to $\theta = 0.1$ and the sensitivity parameter of the membership function is fixed at $\gamma = 1$. To compare to other ensemble methods, this study deployed the threshold $2\sqrt{p}$ for the maximum number of used features and 50% of training samples were randomly sampled to train base learners ($r_s = 0.5$). As common settings in the random forest and ensemble classifiers literature, we set the number of base learners $m = 100$. For other parameters of classifiers, we used default settings of libraries such as scikit-learn [46], XGBoost [8], LightGBM [9] apart from the maximum tree depth of decision trees and tree-based ensemble methods is set to the value of 10 to prevent overfitting [47]. For models using a threshold value for nearest neighbors, we used $K = 5$.

We did not adjust the values of hyperparameters for models in these experiments, although we are aware that a thorough experimental comparison among approaches should tune their hyperparameters to their best for every data set. Our reasons for using the standard implementations in libraries are three-fold. First, our goal in this paper was to achieve initial analyses of the effectiveness of the random hyperboxes classifier. If it is worse than other methods with standard implementations, no further studies would be worthwhile to improve and exploit the proposed method. In the opposite case, we need a comparative study to evaluate the impacts of hyperparameters on the predictive performance of methods. Second, standard implementations in libraries are general enough to perform quite well across many problems [35]. The lack of fine-tuning is compensated by the diversity in the number of features, samples, and classes of the used data sets. These datasets are quite common and were randomly chosen without intentionally favoring any learning algorithms. Moreover, the performance of models is also assessed using 10 times repeated 4-fold cross-validation along with statistical testing methods. Third, the use of default parameters without tuning will be easily reproducible by other studies.

2) A Comparison of the Random Hyperboxes With Other FMNNs:

This subsection provides the results of the average weighted-F1 scores of fuzzy min-max classifiers mentioned in subsection IV.B.1 from the main paper. Among different types of fuzzy min-max neural networks, IOL-GFMM and RFMNN have mechanisms to make the decision when there are at least two winning hyperboxes representing different classes (in this case, the sample is located on the decision boundary). Therefore, to make a fair comparison, other fuzzy min-max classifiers used the Manhattan distance from the input pattern to central points of winning hyperboxes to find the predictive class instead of randomly selecting a class. We have implemented all of these fuzzy min-max neural networks in Python.

The average weighted-F1 scores of classifiers using 10 times repeated 4-fold cross-validation are shown in Table VI for the maximum hyperbox size $\theta = 0.1$ and Table VIII for $\theta = 0.7$. To facilitate the process of evaluating the performance and performing statistical testing, the performance of classifiers on each dataset is ranked, in which the best classifier with the highest average weighted-F1 score is ranked first, and the second-best classifier is ranked two and so on. The classifiers with the same average weighted-F1 scores are assigned the average value of their ranks. Table VII shows the ranks of classifiers using $\theta = 0.1$, while Table IX presents the ranks of classifiers with $\theta = 0.7$.

It can be seen that the random hyperboxes classifier achieves the lowest rank for both high and low values of θ. Its average ranks are twice as low as those of the second-best classifiers. In addition, the random hyperboxes classifier obtains the highest average weighted-F1 scores on almost all considered datasets. These figures show the superior performance of the random hyperboxes classifier in comparison to other types of fuzzy min-max neural networks.

TABLE VI
THE AVERAGE WEIGHTED-F1 SCORES AND STANDARD DEVIATION OF THE RANDOM HYPERBOXES (RH) AND OTHER FUZZY MIN-MAX NEURAL NETWORKS ($\theta = 0.1$)

ID	Dataset	IOL-GFMM	Onln-GFMM	RH	FMNN	EFMNN	KNEFMNN	RFMNN	AGGLO-2	Onln-GFMM + AGGLO-2
1	Balance_scale	**0.79383** ± **0.0558**	0.73358 ± 0.0857	0.73821 ± 0.0894	0.73823 ± 0.0635	0.75990 ± 0.0800	0.75990 ± 0.0800	0.75990 ± 0.0800	**0.79383** ± **0.0558**	**0.79383** ± **0.0558**
2	banknote_authentication	0.99782 ± 0.0013	0.99709 ± 0.0021	0.99818 ± 0.0018	0.99854 ± 0.0015	**0.99927** ± **0.0013**	0.99854 ± 0.0015	**0.99927** ± **0.0013**	0.99854 ± 0.0015	0.99854 ± 0.0015
3	blood_transfusion	0.56827 ± 0.1417	0.60796 ± 0.0623	**0.68264** ± **0.0190**	0.62517 ± 0.0596	0.63917 ± 0.0443	0.64600 ± 0.0520	0.63505 ± 0.0639	0.64685 ± 0.0646	0.58389 ± 0.1132
4	breast_cancer_wisconsin	0.94383 ± 0.0329	0.94827 ± 0.0306	**0.96685** ± **0.0180**	0.95840 ± 0.0229	0.95976 ± 0.0244	0.95976 ± 0.0244	0.95976 ± 0.0243	0.94383 ± 0.0329	0.94383 ± 0.0329
5	BreastCancerCoimbra	0.62798 ± 0.0381	0.62798 ± 0.0381	**0.66561** ± **0.0939**	0.64246 ± 0.1193	0.60216 ± 0.0401	0.60216 ± 0.0401	0.60216 ± 0.0401	0.62798 ± 0.0381	0.62798 ± 0.0381
6	connectionist_bench_sonar	0.53013 ± 0.0816	0.53013 ± 0.0816	0.55964 ± 0.1085	**0.58747** ± **0.1217**	0.55768 ± 0.1160	0.55768 ± 0.1160	0.55768 ± 0.1160	0.53013 ± 0.0816	0.53013 ± 0.0816
7	haberman	0.61185 ± 0.0418	0.61711 ± 0.0387	**0.64211** ± **0.0294**	0.60223 ± 0.0168	0.63494 ± 0.0291	0.63076 ± 0.0172	0.62178 ± 0.0450	0.60629 ± 0.0601	0.58827 ± 0.0710
8	heart	0.76101 ± 0.0272	0.75470 ± 0.0125	**0.82643** ± **0.0252**	0.79486 ± 0.0112	0.77308 ± 0.0295	0.77308 ± 0.0295	0.77308 ± 0.0295	0.76101 ± 0.0272	0.76101 ± 0.0272
9	movement_libras	0.53032 ± 0.0796	0.53032 ± 0.0796	**0.54465** ± **0.1064**	0.47268 ± 0.1048	0.49732 ± 0.0879	0.49732 ± 0.0879	0.49732 ± 0.0879	0.53255 ± 0.0796	0.53255 ± 0.0796
10	pima_diabetes	0.70322 ± 0.0113	0.69864 ± 0.0118	**0.72760** ± **0.0339**	0.71264 ± 0.0182	0.71634 ± 0.0356	0.72053 ± 0.0284	0.71634 ± 0.0356	0.70372 ± 0.0164	0.70216 ± 0.0151
11	plant_species_leaves_margin	0.57408 ± 0.0217	0.58113 ± 0.0251	0.74748 ± 0.0195	0.67553 ± 0.0107	**0.76291** ± **0.0090**	**0.76291** ± **0.0090**	**0.76291** ± **0.0090**	0.57408 ± 0.0217	0.57408 ± 0.0217
12	plant_species_leaves_shape	0.53546 ± 0.0218	0.55296 ± 0.0193	**0.60695** ± **0.0306**	0.49222 ± 0.0242	0.48698 ± 0.0324	0.51801 ± 0.0426	0.48154 ± 0.0340	0.55600 ± 0.0395	0.56172 ± 0.0323
13	ringnorm	0.62981 ± 0.0070	0.62981 ± 0.0070	**0.94237** ± **0.0057**	0.79121 ± 0.0067	0.66059 ± 0.0156	0.60626 ± 0.0135	0.66059 ± 0.0156	0.63184 ± 0.0045	0.63184 ± 0.0045
14	landsat_satellite	0.86259 ± 0.0179	0.86207 ± 0.0191	**0.87875** ± **0.0076**	0.80895 ± 0.0405	0.86752 ± 0.0140	0.86749 ± 0.0165	0.86792 ± 0.0117	0.86631 ± 0.0193	0.86464 ± 0.0163
15	twonorm	0.94283 ± 0.0054	0.94284 ± 0.0054	**0.97211** ± **0.0032**	0.94824 ± 0.0060	0.94932 ± 0.0062	0.94932 ± 0.0062	0.94932 ± 0.0062	0.94284 ± 0.0054	0.94284 ± 0.0054
16	vehicle_silhouettes	0.65535 ± 0.0195	0.65993 ± 0.0183	**0.72044** ± **0.0309**	0.66694 ± 0.0112	0.66552 ± 0.0246	0.66715 ± 0.0200	0.66552 ± 0.0246	0.65307 ± 0.0197	0.65432 ± 0.0210
17	vertebral_column	0.74496 ± 0.0405	0.74136 ± 0.0184	**0.76542** ± **0.0411**	0.69182 ± 0.0385	0.74947 ± 0.0528	0.76296 ± 0.0417	0.75850 ± 0.0518	0.73614 ± 0.0372	0.72441 ± 0.0473
18	vowel	0.55279 ± 0.0491	0.55279 ± 0.0491	**0.63059** ± **0.0676**	0.59758 ± 0.0387	0.58589 ± 0.0400	0.58907 ± 0.0387	0.58589 ± 0.0400	0.55507 ± 0.0525	0.55953 ± 0.0547
19	waveform	0.78002 ± 0.0107	0.78002 ± 0.0107	**0.83795** ± **0.0068**	0.76572 ± 0.0136	0.78752 ± 0.0115	0.78752 ± 0.0115	0.78752 ± 0.0115	0.78002 ± 0.0107	0.78002 ± 0.0107
20	wireless_indoor_localization	0.96511 ± 0.0048	0.96561 ± 0.0061	**0.97726** ± **0.0086**	0.96266 ± 0.0181	0.97252 ± 0.0058	0.97403 ± 0.0110	0.97252 ± 0.0058	0.96364 ± 0.0087	0.96565 ± 0.0068

TABLE VII
THE RANKING OF THE RANDOM HYPERBOXES AND OTHER FUZZY MIN-MAX CLASSFIERS ($\theta = 0.1$)

ID	Dataset	IOL-GFMM	Onln-GFMM	RH	FMNN	EFMNN	KNEFMNN	RFMNN	AGGLO-2	Onln-GFMM + AGGLO-2
1	Balance_scale	**2**	9	8	7	5	5	5	**2**	**2**
2	banknote_authentication	8	9	7	4.5	**1.5**	4.5	**1.5**	4.5	4.5
3	blood_transfusion	9	7	**1**	6	4	3	5	2	8
4	breast_cancer_wisconsin	8	6	**1**	5	3	3	3	8	8
5	BreastCancerCoimbra	4.5	4.5	**1**	2	8	8	8	4.5	4.5
6	connectionist_bench_sonar	7.5	7.5	2	**1**	4	4	4	7.5	7.5
7	haberman	6	5	**1**	8	2	3	4	7	9
8	heart	7	9	**1**	2	4	4	4	7	7
9	movement_libras	4.5	4.5	**1**	9	7	7	7	2.5	2.5
10	pima_diabetes	7	9	**1**	5	3.5	2	3.5	6	8
11	plant_species_leaves_margin	8	6	4	5	**2**	**2**	**2**	8	8
12	plant_species_leaves_shape	5	4	**1**	7	8	6	9	3	2
13	ringnorm	7.5	7.5	**1**	2	3.5	9	3.5	5.5	5.5
14	landsat_satellite	7	8	**1**	9	3	4	2	5	6
15	twonorm	7.5	7.5	**1**	5	3	3	3	7.5	7.5
16	vehicle_silhouettes	7	6	**1**	3	4.5	2	4.5	9	8
17	vertebral_column	5	6	**1**	9	4	2	3	7	8
18	vowel	8.5	8.5	**1**	2	4.5	3	4.5	7	6
19	waveform	6.5	6.5	**1**	9	3	3	3	6.5	6.5
20	wireless_indoor_localization	7	6	**1**	9	3.5	2	3.5	8	5
Average rank		6.625	6.825	**1.85**	5.475	4.05	3.975	4.15	5.875	6.175

3) A Comparison of the Random Hyperboxes With Other Ensemble Classifiers:

This subsection presents the experimental results of the random hyperboxes and other popular ensemble classifiers mentioned in subsection IV.B.2 from the main paper. The base learners of the random hyperboxes model used the threshold $\theta = 0.1$ for the maximum hyperbox size.

TABLE VIII
THE AVERAGE WEIGHTED-F1 SCORES AND STANDARD DEVIATION OF THE RANDOM HYPERBOXES (RH) AND OTHER FUZZY MIN-MAX NEURAL NETWORKS ($\theta = 0.7$)

ID	Dataset	IOL-GFMM	Onln-GFMM	RH	FMNN	EFMNN	KNEFMNN	RFMNN	AGGLO-2	Onln-GFMM + AGGLO-2
1	Balance_scale	0.59351 ± 0.0956	0.54762 ± 0.1334	0.66276 ± 0.0915	0.41057 ± 0.1928	0.60536 ± 0.1306	0.61254 ± 0.0863	**0.70701** ± **0.0942**	0.67825 ± 0.0438	0.67825 ± 0.0438
2	banknote_authentication	0.70285 ± 0.0311	0.80833 ± 0.0184	0.95539 ± 0.0086	0.86049 ± 0.0423	0.76974 ± 0.0901	0.82069 ± 0.0202	0.75250 ± 0.0816	**0.99562** ± **0.0026**	0.9949 ± 0.0013
3	blood_transfusion	0.53269 ± 0.1765	0.46882 ± 0.1652	**0.68779** ± **0.0292**	0.55976 ± 0.0831	0.62575 ± 0.1050	0.60291 ± 0.0905	0.62544 ± 0.0675	0.65971 ± 0.0840	0.62368 ± 0.1017
4	breast_cancer_wisconsin	0.95315 ± 0.0313	0.95995 ± 0.0208	**0.96801** ± **0.0164**	0.88336 ± 0.0623	0.92016 ± 0.0505	0.94232 ± 0.0166	0.95007 ± 0.0307	0.95118 ± 0.0292	0.95111 ± 0.0316
5	BreastCancerCoimbra	0.51162 ± 0.1013	0.59101 ± 0.0692	**0.64862** ± **0.1000**	0.49436 ± 0.1081	0.57430 ± 0.0463	0.61823 ± 0.0204	0.51317 ± 0.0934	0.54535 ± 0.0868	0.54535 ± 0.0868
6	connectionist_bench_sonar	0.49895 ± 0.0870	0.48042 ± 0.1324	**0.56346** ± **0.0870**	0.41787 ± 0.0573	0.51797 ± 0.0904	0.46899 ± 0.1189	0.51250 ± 0.0920	0.48416 ± 0.1419	0.48416 ± 0.1419
7	haberman	0.51121 ± 0.2094	0.49672 ± 0.1993	0.59068 ± 0.1456	0.29147 ± 0.2110	0.58733 ± 0.0958	0.50931 ± 0.1729	0.61214 ± 0.0605	0.62306 ± 0.1450	**0.64127** ± **0.1016**
8	heart	0.77523 ± 0.0512	0.77319 ± 0.0226	**0.81720** ± **0.0253**	0.69788 ± 0.0263	0.78555 ± 0.0494	0.79202 ± 0.0282	0.80062 ± 0.0392	0.76551 ± 0.0317	0.76551 ± 0.0317
9	movement_libras	0.49320 ± 0.0878	0.51492 ± 0.0921	**0.55832** ± **0.0966**	0.31186 ± 0.1207	0.45040 ± 0.1039	0.47365 ± 0.1292	0.36403 ± 0.0916	0.55273 ± 0.1417	0.51082 ± 0.1144
10	pima_diabetes	0.68428 ± 0.0452	0.64318 ± 0.0831	**0.74040** ± **0.0272**	0.58610 ± 0.0589	0.63445 ± 0.0180	0.64443 ± 0.0471	0.63759 ± 0.0174	0.69013 ± 0.0226	0.69448 ± 0.0328
11	plant_species_leaves_margin	0.62682 ± 0.0101	0.63295 ± 0.0097	**0.79507** ± **0.0251**	0.78283 ± 0.0166	0.77767 ± 0.0150	0.78105 ± 0.0186	0.69359 ± 0.0199	0.61894 ± 0.0108	0.61894 ± 0.0108
12	plant_species_leaves_shape	0.52069 ± 0.0166	0.45639 ± 0.0259	**0.61318** ± **0.0251**	0.43395 ± 0.0382	0.43569 ± 0.0355	0.43625 ± 0.0355	0.44809 ± 0.0283	0.56043 ± 0.0257	0.55659 ± 0.0211
13	ringnorm	0.65641 ± 0.1616	0.77946 ± 0.0323	0.86803 ± 0.0250	0.82055 ± 0.0090	0.65747 ± 0.0431	0.67500 ± 0.0705	0.66250 ± 0.0818	**0.87453** ± **0.0018**	**0.87453** ± **0.0018**
14	landsat_satellite	0.83192 ± 0.0269	0.58411 ± 0.0145	0.86226 ± 0.0165	0.52566 ± 0.1310	0.54733 ± 0.1926	0.66287 ± 0.0617	0.75732 ± 0.0439	0.87039 ± 0.0090	**0.87299** ± **0.0148**
15	twonorm	0.91658 ± 0.0088	0.76649 ± 0.0176	**0.97122** ± **0.0029**	0.80136 ± 0.0387	0.79747 ± 0.0167	0.73673 ± 0.0057	0.69279 ± 0.1192	0.95973 ± 0.0016	0.95973 ± 0.0016
16	vehicle_silhouettes	0.64533 ± 0.0184	0.48170 ± 0.0668	**0.70332** ± **0.0255**	0.28571 ± 0.0317	0.48748 ± 0.0196	0.49573 ± 0.0408	0.55892 ± 0.0488	0.65988 ± 0.0328	0.66057 ± 0.0266
17	vertebral_column	0.61791 ± 0.0341	0.74575 ± 0.0193	**0.77698** ± **0.0396**	0.74546 ± 0.0223	0.73233 ± 0.0256	0.75136 ± 0.0213	0.74130 ± 0.0138	0.73627 ± 0.0705	0.75678 ± 0.0547
18	vowel	0.53239 ± 0.0370	0.43198 ± 0.0434	**0.58876** ± **0.0698**	0.33920 ± 0.0476	0.40335 ± 0.0485	0.40713 ± 0.0409	0.46578 ± 0.0521	0.52614 ± 0.0633	0.53066 ± 0.0401
19	waveform	0.79168 ± 0.0150	0.76606 ± 0.0090	**0.83201** ± **0.0098**	0.70826 ± 0.0275	0.73494 ± 0.0188	0.75625 ± 0.0151	0.51829 ± 0.0275	0.81138 ± 0.0126	0.81138 ± 0.0126
20	wireless_indoor_localization	0.85201 ± 0.0450	0.93139 ± 0.0240	**0.97437** ± **0.0093**	0.92680 ± 0.0280	0.86827 ± 0.0470	0.86771 ± 0.0484	0.84572 ± 0.0483	0.97389 ± 0.0053	0.97398 ± 0.0042

TABLE IX
THE RANKING OF THE RANDOM HYPERBOXES AND OTHER FUZZY MIN-MAX CLASSFIERS ($\theta = 0.7$)

ID	Dataset	IOL-GFMM	Onln-GFMM	RH	FMNN	EFMNN	KNEFMNN	RFMNN	AGGLO-2	Onln-GFMM + AGGLO-2
1	Balance_scale	7	8	4	9	6	5	**1**	2.5	2.5
2	banknote_authentication	9	6	3	4	7	5	8	**1**	2
3	blood_transfusion	8	9	**1**	7	3	6	4	2	5
4	breast_cancer_wisconsin	3	2	**1**	9	8	7	6	4	5
5	BreastCancerCoimbra	8	3	**1**	9	4	2	7	5.5	5.5
6	connectionist_bench_sonar	4	7	**1**	9	2	8	3	5.5	5.5
7	haberman	7	8	4	9	5	6	3	2	**1**
8	heart	5	6	**1**	9	4	3	2	7.5	7.5
9	movement_libras	5	3	**1**	9	7	6	8	2	4
10	pima_diabetes	4	6	**1**	9	8	5	7	3	2
11	plant_species_leaves_margin	7	6	**1**	2	4	3	5	8.5	8.5
12	plant_species_leaves_shape	4	5	**1**	9	8	7	6	2	3
13	ringnorm	9	5	3	4	8	6	7	**1.5**	**1.5**
14	landsat_satellite	4	7	3	9	8	6	5	2	**1**
15	twonorm	4	7	**1**	5	6	8	9	2.5	2.5
16	vehicle_silhouettes	4	8	**1**	9	7	6	5	3	2
17	vertebral_column	9	4	**1**	5	8	3	6	7	2
18	vowel	2	6	**1**	9	8	7	5	4	3
19	waveform	4	5	**1**	8	7	6	9	2.5	2.5
20	wireless_indoor_localization	8	4	**1**	5	6	7	9	3	2
	Average rank	5.75	5.75	**1.6**	7.4	6.2	5.6	5.75	3.55	3.4

Table X shows the average weighted-F1 scores of classifiers through 40 iterations with different testing folds (10 times repeated 4-fold cross-validation). The ranks of these classifiers are shown in Table XI.

It can be observed that the average performance of the random hyperboxes is much better than the results of Random Forest, Rotation Forest, Gradient Boosting, and the ensemble models of IOL-GFMM learners with full features. It is also

TABLE X
THE AVERAGE WEIGHTED-F1 SCORES AND STANDARD DEVIATION OF THE RANDOM HYPERBOX MODEL AND OTHER ENSEMBLE MODELS

ID	Dataset	RH	Random Forest	Rotation Forest	XGBoost	LightGBM	Gradient Boosting	Ens-IOL-GFMM (DL)	Ens-IOL-GFMM (ML)
1	Balance_scale	0.73821 ± 0.0894	0.59325 ± 0.1943	0.71405 ± 0.1039	0.69423 ± 0.1589	0.59721 ± 0.2134	0.58660 ± 0.1947	**0.79987** ± **0.0476**	0.79383 ± 0.0558
2	banknote_authentication	0.99818 ± 0.0018	0.99054 ± 0.0043	0.99052 ± 0.0053	0.99563 ± 0.0025	0.99636 ± 0.0032	0.99709 ± 0.0021	0.9984 ± 0.0017	**0.99869** ± **0.0015**
3	blood_transfusion	0.68264 ± 0.0190	0.66193 ± 0.0331	0.63456 ± 0.0630	0.68826 ± 0.0368	0.66894 ± 0.0418	0.64810 ± 0.0448	**0.68847** ± **0.0268**	0.65257 ± 0.0706
4	breast_cancer_wisconsin	**0.96685** ± **0.0180**	0.95570 ± 0.0253	0.95426 ± 0.0140	0.96139 ± 0.0174	0.94973 ± 0.0181	0.95848 ± 0.0214	0.94721 ± 0.0332	0.94383 ± 0.0329
5	BreastCancerCoimbra	0.66561 ± 0.0939	0.66808 ± 0.0917	0.61706 ± 0.0808	0.70412 ± 0.1285	**0.74284** ± **0.0879**	0.65378 ± 0.1094	0.62096 ± 0.0631	0.62798 ± 0.0381
6	connectionist_bench_sonar	0.55964 ± 0.1085	0.55992 ± 0.0453	0.60441 ± 0.0425	**0.63616** ± **0.0802**	0.61462 ± 0.0408	0.55653 ± 0.0603	0.52737 ± 0.1244	0.53013 ± 0.0816
7	haberman	0.64211 ± 0.0294	0.55050 ± 0.1330	0.53359 ± 0.0789	0.63966 ± 0.0488	0.65125 ± 0.0478	0.63137 ± 0.0214	**0.65373** ± **0.0428**	0.59270 ± 0.0865
8	heart	**0.82643** ± **0.0252**	0.80593 ± 0.0534	0.81462 ± 0.0298	0.81757 ± 0.0430	0.79135 ± 0.0463	0.80947 ± 0.0393	0.75458 ± 0.0185	0.76101 ± 0.0272
9	movement_libras	**0.54465** ± **0.1064**	0.48745 ± 0.1456	0.47701 ± 0.1957	0.42539 ± 0.1558	0.46981 ± 0.1835	0.39863 ± 0.1370	0.5254 ± 0.0858	0.52819 ± 0.0818
10	pima_diabetes	0.72760 ± 0.0339	0.76566 ± 0.0455	**0.77200** ± **0.0474**	0.75014 ± 0.0364	0.72647 ± 0.0228	0.75678 ± 0.0357	0.72359 ± 0.0230	0.70680 ± 0.0148
11	plant_species_leaves_margin	0.74748 ± 0.0195	0.72687 ± 0.0246	0.66113 ± 0.0356	**0.78722** ± **0.0096**	0.78549 ± 0.0139	0.32808 ± 0.0174	0.59427 ± 0.0167	0.57408 ± 0.0217
12	plant_species_leaves_shape	**0.60695** ± **0.0306**	0.55444 ± 0.0146	0.52707 ± 0.0139	0.55204 ± 0.0201	0.50336 ± 0.0148	0.31396 ± 0.0311	0.56949 ± 0.0293	0.58314 ± 0.0103
13	ringnorm	0.94237 ± 0.0057	0.94594 ± 0.0063	0.92605 ± 0.0021	0.97851 ± 0.0030	**0.98094** ± **0.0008**	0.97810 ± 0.0041	0.58982 ± 0.0059	0.59580 ± 0.0058
14	landsat_satellite	0.87875 ± 0.0076	0.88374 ± 0.0082	0.88817 ± 0.0092	0.89463 ± 0.0060	**0.89942** ± **0.0059**	0.89217 ± 0.0059	0.86874 ± 0.0110	0.86412 ± 0.0165
15	twonorm	0.97211 ± 0.0032	0.97081 ± 0.0009	0.96554 ± 0.0033	0.97230 ± 0.0019	0.97311 ± 0.0016	**0.97365** ± **0.0010**	0.96477 ± 0.0039	0.94284 ± 0.0054
16	vehicle_silhouettes	0.72044 ± 0.0309	0.74634 ± 0.0251	0.73402 ± 0.0173	0.76474 ± 0.0136	**0.76918** ± **0.0086**	0.76615 ± 0.0064	0.66757 ± 0.0214	0.65051 ± 0.0206
17	vertebral_column	0.76542 ± 0.0411	**0.85617** ± **0.0329**	0.74431 ± 0.0451	0.80543 ± 0.0398	0.82058 ± 0.0497	0.81368 ± 0.0391	0.73812 ± 0.0572	0.76917 ± 0.0320
18	vowel	**0.63059** ± **0.0676**	0.59992 ± 0.0470	0.57678 ± 0.1026	0.60286 ± 0.0685	0.60048 ± 0.0465	0.60462 ± 0.0529	0.54363 ± 0.0684	0.56073 ± 0.0485
19	waveform	0.83795 ± 0.0068	0.84593 ± 0.0120	0.85010 ± 0.0093	**0.85878** ± **0.0065**	0.85587 ± 0.0037	0.85855 ± 0.0086	0.81709 ± 0.0082	0.78002 ± 0.0107
20	wireless_indoor_localization	**0.97726** ± **0.0086**	0.97348 ± 0.0159	0.95229 ± 0.0112	0.97599 ± 0.0112	0.97554 ± 0.0088	0.97344 ± 0.0185	0.97566 ± 0.0037	0.97112 ± 0.0058

TABLE XI
THE RANKING OF THE RANDOM HYPERBOX MODEL AND OTHER ENSEMBLE MODELS

ID	Dataset	RH	Random Forest	Rotation Forest	XGBoost	LightGBM	Gradient Boosting	Ens-IOL-GFMM (DL)	Ens-IOL-GFMM (ML)
1	Balance_scale	3	7	4	5	6	8	**1**	2
2	banknote_authentication	3	7	8	6	5	4	2	**1**
3	blood_transfusion	3	5	8	2	4	7	**1**	6
4	breast_cancer_wisconsin	**1**	4	5	2	6	3	7	8
5	BreastCancerCoimbra	4	3	8	2	**1**	5	7	6
6	connectionist_bench_sonar	5	4	3	**1**	2	6	8	7
7	haberman	3	7	8	4	2	5	**1**	6
8	heart	**1**	5	3	2	6	4	8	7
9	movement_libras	**1**	4	5	7	6	8	3	2
10	pima_diabetes	5	2	**1**	4	6	3	7	8
11	plant_species_leaves_margin	3	4	5	**1**	2	8	6	7
12	plant_species_leaves_shape	**1**	4	6	5	7	8	3	2
13	ringnorm	5	4	6	2	**1**	3	8	7
14	landsat_satellite	6	5	4	2	**1**	3	7	8
15	twonorm	4	5	6	3	2	**1**	7	8
16	vehicle_silhouettes	6	4	5	3	**1**	2	7	8
17	vertebral_column	6	**1**	7	4	2	3	8	5
18	vowel	**1**	5	6	3	4	2	8	7
19	waveform	6	5	4	**1**	3	2	7	8
20	wireless_indoor_localization	**1**	5	8	2	4	6	3	7
Average rank		3.4	4.5	5.5	**3.05**	3.55	4.55	5.45	6

slightly better than LightGBM, but the random hyperboxes classifier cannot outperform the XGBoost model on 20 considered datasets. In spite of using the same base learners and sampling method, the random hyperboxes classifier is much better than the Ens-IOL-GFMM with decision and model combination levels. It is due to the fact that the random hyperboxes classifier uses only a subset of features to train each base learner. This method reduces the correlation between base learners, and so it leads to the reduction of generalization errors. These empirical results are consistent with the theoretical results presented in the main paper. However, it is also noted that the correlation is linked with variance, so achieving a low correlation but high variance will not decrease the prediction error. In addition, when reducing correlation by using a smaller number of features, it will also increase the variance of each base learner. Therefore, to achieve the reduction of prediction error, the correlation between base learners has to decrease faster than the growth of the variance. This issue needs to be analyzed in more details in the future study, especially the relationship between the maximum number of used features and the number of base learners.

4) A Comparison of the Random Hyperboxes With Other Machine Learning Algorithms:

This part presents the empirical results of the random hyperboxes classifier and other popular machine learning algorithms shown in subsection IV.B.3 from the main paper.

Table XII shows the average weighted-F1 scores of the random hyperboxes and other classifiers for the 20 datasets. The ranks of these models are presented in Table XIII. In this experiment, the random hyperboxes achieved the best performance among considered classifiers.

TABLE XII
THE AVERAGE WEIGHTED-F1 SCORES AND STANDARD DEVIATION OF THE RANDOM HYPERBOXES AND OTHER MACHINE LEARNING ALGORITHMS

ID	Dataset	Random hyperboxes	Decision tree	SVM	KNN	Naive Bayes	LDA
1	Balance_scale	0.73821 ± 0.0894	0.57709 ± 0.1822	**0.80213 ± 0.0505**	0.74637 ± 0.0566	0.64495 ± 0.1950	0.76648 ± 0.0743
2	banknote_authentication	0.99818 ± 0.0018	0.98471 ± 0.0052	0.97891 ± 0.0082	**0.99854 ± 0.0015**	0.83836 ± 0.0232	0.97674 ± 0.0080
3	blood_transfusion	0.68264 ± 0.0190	0.64101 ± 0.0670	0.65913 ± 0.0019	0.64196 ± 0.1086	0.70667 ± 0.0386	**0.72411 ± 0.0748**
4	breast_cancer_wisconsin	**0.96685 ± 0.0180**	0.92186 ± 0.0306	0.95989 ± 0.0267	0.96287 ± 0.0222	0.96035 ± 0.0154	0.95521 ± 0.0356
5	BreastCancerCoimbra	0.66561 ± 0.0939	0.63368 ± 0.1102	0.38407 ± 0.0145	**0.68939 ± 0.1169**	0.56916 ± 0.1305	0.58265 ± 0.1505
6	connectionist_bench_sonar	0.55964 ± 0.1085	0.53636 ± 0.0475	**0.56390 ± 0.1128**	0.47834 ± 0.1083	0.53318 ± 0.1559	0.55007 ± 0.0890
7	haberman	0.64211 ± 0.0294	0.56815 ± 0.0639	0.62317 ± 0.0036	0.68248 ± 0.0334	0.69737 ± 0.0264	**0.70811 ± 0.0171**
8	heart	0.82643 ± 0.0252	0.76268 ± 0.0588	0.83253 ± 0.0339	0.80648 ± 0.0171	**0.84803 ± 0.0122**	0.83265 ± 0.0335
9	movement_libras	**0.54465 ± 0.1064**	0.34075 ± 0.1197	0.41919 ± 0.1311	0.50787 ± 0.1029	0.42762 ± 0.1902	0.50048 ± 0.0515
10	pima_diabetes	0.72760 ± 0.0339	0.70859 ± 0.0347	0.73475 ± 0.0259	0.74955 ± 0.0224	0.74796 ± 0.0232	**0.76027 ± 0.0327**
11	plant_species_leaves_margin	0.74748 ± 0.0195	0.17446 ± 0.0127	0.72261 ± 0.0118	0.74603 ± 0.0092	0.71911 ± 0.0191	**0.78220 ± 0.0189**
12	plant_species_leaves_shape	**0.60695 ± 0.0306**	0.32775 ± 0.0372	0.40432 ± 0.0226	0.54375 ± 0.0349	0.51460 ± 0.0173	0.47546 ± 0.0369
13	ringnorm	0.94237 ± 0.0057	0.86909 ± 0.0040	0.83666 ± 0.0038	0.66874 ± 0.0097	**0.98662 ± 0.0016**	0.77029 ± 0.0015
14	landsat_satellite	0.87875 ± 0.0076	0.83136 ± 0.0183	0.82357 ± 0.0107	**0.88190 ± 0.0130**	0.79565 ± 0.0370	0.82012 ± 0.0116
15	twonorm	0.97211 ± 0.0032	0.84553 ± 0.0066	0.97824 ± 0.0017	0.97284 ± 0.0022	**0.97892 ± 0.0020**	0.97838 ± 0.0025
16	vehicle_silhouettes	0.72044 ± 0.0309	0.71707 ± 0.0262	0.55849 ± 0.0273	0.69223 ± 0.0067	0.44138 ± 0.0210	**0.78081 ± 0.0304**
17	vertebral_column	0.76542 ± 0.0411	0.81931 ± 0.0454	0.61393 ± 0.0375	0.73993 ± 0.0351	**0.82177 ± 0.0346**	0.80953 ± 0.0673
18	vowel	**0.63059 ± 0.0676**	0.43943 ± 0.0552	0.36361 ± 0.0883	0.57885 ± 0.0586	0.52917 ± 0.0814	0.46100 ± 0.0763
19	waveform	0.83795 ± 0.0068	0.76834 ± 0.0049	**0.86992 ± 0.0025**	0.84700 ± 0.0080	0.79820 ± 0.0045	0.86255 ± 0.0074
20	wireless_indoor_localization	0.97726 ± 0.0086	0.96088 ± 0.0175	0.97405 ± 0.0052	0.97556 ± 0.0040	**0.98055 ± 0.0096**	0.97074 ± 0.0062

TABLE XIII
THE RANKING OF THE RANDOM HYPERBOXES AND OTHER MACHINE LEARNING ALGORITHMS

ID	Dataset	Random hyperboxes	Decision tree	SVM	KNN	Naive Bayes	LDA
1	Balance_scale	4	6	**1**	3	5	2
2	banknote_authentication	2	3	4	**1**	6	5
3	blood_transfusion	3	6	4	5	2	**1**
4	breast_cancer_wisconsin	**1**	6	4	2	3	5
5	BreastCancerCoimbra	2	3	6	**1**	5	4
6	connectionist_bench_sonar	2	4	**1**	6	5	3
7	haberman	4	6	5	3	2	**1**
8	heart	4	6	3	5	**1**	2
9	movement_libras	**1**	6	5	2	4	3
10	pima_diabetes	5	6	4	2	3	**1**
11	plant_species_leaves_margin	2	6	4	3	5	**1**
12	plant_species_leaves_shape	**1**	6	5	2	3	4
13	ringnorm	2	3	4	6	**1**	5
14	landsat_satellite	2	3	4	**1**	6	5
15	twonorm	5	6	3	4	**1**	2
16	vehicle_silhouettes	2	3	5	4	6	**1**
17	vertebral_column	4	2	6	5	**1**	3
18	vowel	**1**	5	6	2	3	4
19	waveform	4	6	**1**	3	5	2
20	wireless_indoor_localization	2	6	4	3	**1**	5
Average rank		**2.65**	4.9	3.95	3.15	3.4	2.95

References

[1] B. Gabrys and A. Bargiela, "General fuzzy min-max neural network for clustering and classification," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 769–783, 2000.

[2] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, pp. 206–215, 2019.

[3] G. Biau, L. Devroye, and G. Lugosi, "Consistency of random forests and other averaging classifiers," *Journal of Machine Learning Research*, vol. 9, pp. 2015–2033, 2008.

[4] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[5] ——, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[6] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119 – 139, 1997.

[7] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[8] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[9] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 3146–3154.

[10] T. T. Khuat, D. Ruta, and B. Gabrys, "Hyperbox based machine learning algorithms: A comprehensive survey," *arXiv e-prints*, p. arXiv:1901.11303, 2019.

[11] B. Gabrys, "Combining neuro-fuzzy classifiers for improved generalisation and reliability," in *Proceedings of the 2002 International Joint Conference on Neural Networks*, 2002a, pp. 2410–2415.

[12] B. Gabrys, "Learning hybrid neuro-fuzzy classifier models from data: to combine or not to combine?" *Fuzzy Sets and Systems*, vol. 147, no. 1, pp. 39–56, 2004.

[13] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Information fusion*, vol. 6, no. 1, pp. 63–81, 2005.

[14] T. Hastie, T. Robert, and J. Friedman, *The Elements of Statistical Learning: Data mining, Inference and Prediction*, 2nd ed. New York: Springer, 2009.

[15] O. N. Al Sayaydeh, M. F. Mohammed, and C. P. Lim, "Survey of fuzzy min–max neural network for pattern classification variants and applications," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 4, pp. 635–645, 2019.

[16] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Computation*, vol. 9, no. 7, pp. 1545–1588, 1997.

[17] Tin Kam Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[18] R. J. Durrant and A. Kabán, "Random projections as regularizers: learning a linear discriminant from fewer observations than dimensions," *Machine Learning*, vol. 99, no. 2, pp. 257–286, 2015.

[19] P. Andras, "Random projection neural network approximation," in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 2380–2387.

[20] T. I. Cannings and R. J. Samworth, "Random-projection ensemble classification," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 79, no. 4, pp. 959–1035, 2017.

[21] P. K. Simpson, "Fuzzy min-max neural networks. i. classification," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 776–786, 1992.

[22] ——, "Fuzzy min-max neural networks - part 2: Clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 1, p. 32, 1993.

[23] B. Gabrys, "Agglomerative learning algorithms for general fuzzy min-max neural network," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 32, no. 1, pp. 67–82, 2002b.

[24] A. Bargiela, W. Pedrycz, and M. Tanaka, "An inclusion/exclusion fuzzy hyperbox classifier," *International Journal of Knowledge-based and Intelligent Engineering Systems*, vol. 8, no. 2, pp. 91–98, 2004.

[25] T. T. Khuat, F. Chen, and B. Gabrys, "An improved online learning algorithm for general fuzzy min-max neural network," in *Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–9.

[26] D. N. Politis, J. P. Romano, and M. Wolf, *Subsampling*. Springer Science & Business Media, 1999.

[27] T. T. Khuat and B. Gabrys, "Accelerated learning algorithms of general fuzzy min-max neural network using a novel hyperbox selection rule," *arXiv preprint*, p. arXiv:2003.11333, 2020.

[28] C. Mallah, J. Cope, and J. Orwell, "Plant leaf classification using probabilistic integration of shape, texture and margin features," *Signal Processing, Pattern Recognition and Applications*, vol. 5, no. 1, pp. 45–54, 2013.

[29] M. F. Mohammed and C. P. Lim, "An enhanced fuzzy min–max neural network for pattern classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 3, pp. 417–429, 2015.

[30] M. F. Mohammed and C. P. Lim, "Improving the fuzzy min-max neural network with a k-nearest hyperbox expansion rule for pattern classification," *Applied Soft Computing*, vol. 52, pp. 135 – 145, 2017.

[31] O. N. Al Sayaydeh, M. F. Mohammed, E. Alhroob, H. Tao, and C. P. Lim, "A refined fuzzy min-max neural network with new learning procedures for pattern classification," *IEEE Transactions on Fuzzy Systems*, vol. Early Access, 2020.

[32] T. T. Khuat and B. Gabrys, "A comparative study of general fuzzy min-max neural networks for pattern classification problems," *Neurocomputing*, vol. 386, pp. 110 – 125, 2020.

[33] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.

[34] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[35] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1619–1630, 2006.

[36] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[37] H. Zhang, "The optimality of naive bayes," in *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference*, 2004, p. 562–567.

[38] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[39] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[40] J. Ye, "Least squares linear discriminant analysis," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 1087–1093.

[41] M. Budka and B. Gabrys, "Density-preserving sampling: Robust and efficient alternative to cross-validation for error estimation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 1, pp. 22–34, 2013.

[42] J. H. Friedman, "On bias, variance, 0/1—loss, and the curse-of-dimensionality," *Data mining and knowledge discovery*, vol. 1, no. 1, pp. 55–77, 1997.

[43] M. Cuturi, "Fast global alignment kernels," in *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011, pp. 929–936.

[44] N. Helwig, E. Pignanelli, and A. Schütze, "Condition monitoring of a complex hydraulic system using multivariate statistics," in *The Proc. of IEEE International Instrumentation and Measurement Technology Conference*, 2015, pp. 210–215.

[45] D. Dua and C. Graff, "UCI machine learning repository," 2019. [Online]. Available: http://archive.ics.uci.edu/ml

[46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[47] D. Bertsimas and J. Dunn, "Optimal classification trees," *Machine Learning*, vol. 106, no. 7, pp. 1039–1082, 2017.