A Design of Distributed Rule-Based Models in the Presence of Large Data

Hanyu E, Ye Cui, Witold Pedrycz, *Life Fellow*, *IEEE*, Zhiwu Li, *Fellow*, *IEEE*

This work was supported from the Canada Research Chair (CRC) Program and the University of Alberta's Future Energy Systems Research Initiative is gratefully acknowledged. (*Corresponding author:Zhiwu Li*).

H. E is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6R 2V4, Canada (e-mail: hanyu6@ualberta.ca).

Y. Cui is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6R 2V4, Canada (e-mail: ycui7@ualberta.ca).

W. Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6R 2V4, Canada, Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland and also with the Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia (e-mail: wpedrycz@ualberta.ca).

Z. Li is with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau 999078, China and also with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China (e-mail: zhwli@xidian.edu.cn).

***Abstract*—**Generally, fuzzy models, especially rule-based models, are designed in a monolithic manner, meaning that all data are used *en bloc* to design the model. At the same time, there is a visible need to cope with the ever-increasing volumes of data (both in terms of the number of data and their dimensionality) as well as being faced with distributed data located at various locations. The objective of this study is to develop a concept and provide a design framework as well as assess its performance for constructing a collection of rule-based models on a basis of a randomly sampled repository of data and then realize their aggregation. More specifically, for the sampled data, the design of each model is carried out in a standard way as commonly encountered in the case of Takagi-Sugeno (TS) rule-based models and next augmented by gradient boosting. The aggregation is realized by optimizing a weighting scheme applied to the results of the individual models. Our intent is also to carefully demonstrate the performance offered by the mechanisms of machine learning applied in the setting of rule-based models, which is an original task completed before. A number of high-dimensional data are used in the experimental studies to complete a thorough assessment. A comparative performance analysis is reported with respect to the monolithically developed TS models.

Index Terms—distributed rule-based model, gradient boosting, aggregation, data dimensionality, curse of dimensionality

I. Introduction

Since their inception, TS fuzzy rule-based models [1] have attracted much attention. There have been numerous studies devoted to their analysis and design for solving various classification and prediction tasks. The design methodology has been applied to a spectrum of practical problems encountered in diagnosis [2]–[5], decision-making, risk assessment [6]–[8], prediction [9]–[13], and control [14]. For instance, in [14], a new relaxed resilient fuzzy stabilization of discrete time TS system based on the switching-type gain-scheduling control law is proposed. Rule-based models are inherently nonlinear and this helps capture the characteristics of the data (resulting in nonlinear input-output mappings) by changing the number of rules, and by selecting types of membership functions and types of local functions that form the conclusions of the rules. The issues associated with the high dimensionality of data (referred to as the curse of dimensionality) have begun to negatively affect the performance of the resulting models. To alleviate this problem, various well-known methods of dimensionality reduction have been involved in the design of the rules, such as autoencoders [15]–[19], nonnegative matrix factorization [20]–[23], Principal Component Analysis (PCA) [24]–[26], among others. The drawback is the reduced interpretability. Some optimization of the existing methods or porting them to different computing platforms to cope with the increasing data size have been explored [27]–[29]. However, these approaches concern the volumes of data in terms of the number of data as typically witnessed in big data.

The main objective of the study is to develop a comprehensive design of an ensemble of rule-based models producing a distributed architecture to cope with high-dimensional data. This architecture is tractable; the underlying design process is established by engaging ensemble learning. The basic model we used is generic rule-based model, it is refined by forming rules based on the residuals. A suite of experimental studies is covered; with the proposed architecture, a certain level of originality arises. First, the concept of a *distributed model* is introduced and analyzed: the structure of the rule-based model regarded as a base model in the proposed architecture is used in the distributed structure to cope with subsets of data, thus avoiding the curse of dimensionality. Second, an overall design process is established: a standard construction is augmented by gradient boosting and then an aggregation of the individual models in the ensemble (weighting scheme) is developed.

This study is structured into seven sections. First, the related studies are reviewed briefly in Section II. The architecture of the distributed model based on the randomized individual models is covered in Section III. Subsequently, we focus on the detailed development process in Section IV. In Section V, a detailed experimental study is reported. In Section VI, a number of experiments involving high-dimensional data are discussed. The comparative studies are also included to help identify limitations implied by the aspect of high data dimensionality. Conclusions are covered in Section VII.

II. Literature Review

In the context of rule-based models, we have seen a great deal of progress in terms of analysis, design, and deployment of models in specific application domains. In what follows, we elaborate on the main design strategies vis-à-vis the growing modeling challenges encountered with the ongoing demands to cope with the growing dimensionality and complexity of data, as well as the interpretability of the resulting models, among others. The need to collect, store and process large amounts of data is omnipresent

[30]. In light of this situation, machine learning along with its numerous augmentations has become a promising contributor to the algorithmic augmentations of the existing design and analysis practices.

Ensemble learning - The underlying idea is to realize modeling through a series of multiple independent models. The idea was first proposed by Dasarathy and Sheela [31]. Different from traditional machine learning, the ensemble algorithm focuses on the integration of the results obtained by the independent models. The ideas of bagging form a visible tendency in this area. In the realm of fuzzy rules, ensemble learning was applied to assessing a driving style; the fusion of fuzzy rule models led to an increase in the accuracy of the evaluation to 94% [32]. Dieu [33] proposed a tree ensemble algorithm based on a fuzzy rule-based model to predict flash floods. Hu designed the bagging and boosting mechanisms for assembling fuzzy rule-based models and demonstrated that the performance of the ensemble model was superior to the traditional single model for most datasets [34]. The distributed way and hierarchically driven way of rule development was discussed in [35]. Some augmentations of the resulting models with the aid of information granules were investigated [36], [37].

Gradient boosting - Introduced in [38] and originally focusing on the construction of decision trees, gradient boosting is aimed at the successive refinements of some initial models by forming successive models to compensate for errors associated with the initially developed models. Each refinement is realized by considering a model constructed on a basis of input-error pairs of data. The process is repeated and starting from a simple model consisting of a few rules, one can arrive at a large number of rules that as an ensemble can achieve high accuracy. There are active developments in this area with examples of applications, e.g., in problems of prediction solar radiation [39] with the improvement of 40% reported in comparison with random forests. Likewise, improved prediction results were reported in [40]. Chang proposed a model based on a gradient boosting algorithm for risk assessment of financial institutions and improved the accuracy rate from 77% to 90% of traditional algorithms [41]. Wu used a gradient boosting algorithm for solar radiation prediction, effectively reducing the prediction error by 39% [42].

Adaptive boosting (Adaboost) - First proposed in [43], adaptive boosting uses an adaptive way of focusing the attention on incorrectly classified data by associating them with heavier weighting so that the designed model (classifier) is made more focused on such data. The approach is iterative, and the way of re-weighting data is repeated until some stopping criteria have been satisfied. This design strategy is applied in conjunction with various models including decision tree [44], SVM [45], naïve Bayes [46], *K*-means [47], etc.

In sum, in spite of the intensive body of knowledge established in the area of fuzzy rule-based models, there are several aspects of coping with high-dimensional data and a thorough exploration of mechanisms of machine learning focused on ensemble models that have not been fully explored and yet they deserve attention.

III. An Architecture Of The Model And Its Underlying Processing

The structure of the overall model is composed of a collection of rule-based models that are built on a basis of randomly selected subsets of data.

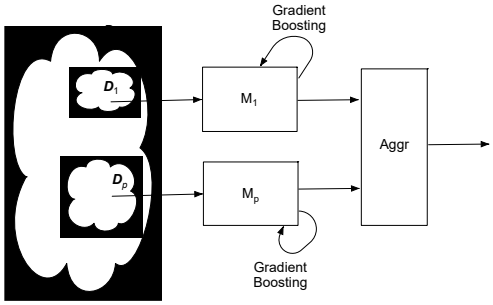


Fig. 1. Overall structure of the model and its functioning.

The main steps of processing completed by each module as outlined in Fig. 1 are carried out as follows:

(i) training of the rule-based models is completed on a basis of randomly selected data $\boldsymbol{D}_j,\ j = 1,2,\ldots,p$. A subset of training data is composed of randomly selected data and randomly selected features. The selection is realized by sampling with replacement so that the same probability of being drawn is ensured for all data instances and features. The feature selection is more important since the selected features compose low-dimensional data which can avoid the concentration effect [48].

(ii) for each subset of data formed above, the standard design process of the TS rule-based model M_j is completed. First, the data are clustered which leads to the condition parts of the rules. Next, the local linear functions forming the conclusions of the rules are optimized by minimizing the sum of squared errors.

(iii) the above design process is augmented by enhancing the performance of the models by engaging its gradient boosting.

(iv) finally, the results of the constructed models are aggregated by the aggregation module; here a linear weighted aggregation scheme is usually considered.

IV. The Design Of Main Modules Of The Model

In what follows, we proceed in detail in accordance with the architecture in Fig.1. As usual, in the overall design process, the available data $\boldsymbol{D}$ is split into the training $\boldsymbol{D}_{train}$ and testing $\boldsymbol{D}_{test}$.

A Generic rule-based model

The rules come in the standard format

$$\text{If } \boldsymbol{x} \text{ is } A_i, \text{then } L_i(\boldsymbol{x}; a_{0i}, \boldsymbol{a}_i) = a_{0i} + \boldsymbol{a}_i^T \boldsymbol{x}, i = 1,2, \dots, c \tag{1.}$$

where A_i is an information granule (fuzzy set) defined in the space of randomly selected datapoints and features. $L_i(\boldsymbol{x}; a_{0i}, \boldsymbol{a}_i)$ is a linear function forming the conclusion part while a_{0i} and a_i are the parameters of the linear function. The parameter c stands for the number of rules. A_i is produced by the FCM clustering. The j-th output of rule-based model is computed on a collection of c rules.

$$y_j = \sum_{i=1}^{c} A_i(\boldsymbol{x}_j) L_i(\boldsymbol{x}_j; a_{0i}, \boldsymbol{a}_i) \tag{2.}$$

where $j = 1,2, \dots, N^*$. The number of rules c varies across the models as it is determined by minimizing the sum of squared errors shown below. Different data in distributed models result in different values of the optimal number of clusters. In addition, the parameters of the local functions are also determined by the minimization of the sum of squared errors

$$error = \frac{1}{N^*} \sum_{j=1}^{N^*} \left(target_j - y_j\right)^2 \tag{3.}$$

where the sum is taken over the corresponding N^* randomly selected data.

In total, we consider p rule-based models, an interesting question arises as to the usage of all features in $\boldsymbol{D}_{train}$ across the subsets of training data $\boldsymbol{D}_{train,j}$, $j = 1, 2, \dots, p$. Note that along with the data, we also randomly pick up a certain proportion (r) of features. Thus, it is of interest to assess how many models (p) have to be constructed to involve all features in the construction of the aggregated model. The probability *prob* of this event (stating that each variable being chosen) is expressed as [49]

$$prob = r + (1-r)r + \cdots + (1-r)^p r = r\left(\frac{1-(1-r)^p}{1-(1-r)}\right) \tag{4.}$$

If we require a certain level of probability (*prob*) to be achieved, for a given value of r, the above relationship helps determine how many models p has to be built.

From the practical perspective, we may request that the value of r should not be too low. If so, each rule-based model cannot capture the input-output dependencies. On the other hand, the excessively high dimensionality r may lead to the deterioration of the rules because of the concentration effect (and this has a detrimental impact on the clustering results). Therefore, a model built for a smaller number of input variables (viz. 2-3) translates into the corresponding value of r. The plot of probability that all the variables have been selected displayed as a function of p for the selected values of r is shown in Fig. 2. This relationship helps determine the number of models once the value of r has been specified and the required minimal probability p has been fixed.

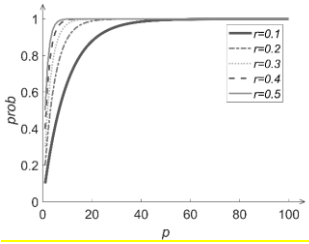


Fig. 2. Probability *prob* versus p for selected values of r.

From Fig. 2, we see that when 10% of the original features are selected each time, it becomes necessary to construct 90 models with randomly selected features to use all of them. If the probability *prob* has been set up as 0.7, we require 12 models with $r = 0.1$ while 2 models in case of $r = 0.5$.

Gradient boosting of the rule-based model. Each rule-based model is further refined by applying gradient boosting. The objective here is to improve the performance of the initially constructed models. Here we follow a well-known scheme of updates for the output of the model, guided by the error values [50].

Consider the j^{th} model M_j is constructed on the basis of $\boldsymbol{D}_{train,j}$. One determines the corresponding errors e_k, $k = 1,2, \dots, card(\boldsymbol{D}_{train,j})$ produced by this model and constructs an auxiliary model $M_j^{\sim}$ on the basis of input-output pairs in the format $(\boldsymbol{x}_k, e_k)$, and then aggregates the result of the model and the auxiliary construct in the additive form $M_j(\boldsymbol{x}_k) + \lambda M_j^{\sim}(\boldsymbol{x}_k)$ such that the sum of errors between the data and the aggregate above is minimized by choosing a suitable value of λ from 0 to 1. The above boosting process is repeated K times by forming successive refinements of the augmented models to obtain the optimized model result $M_{opt,j}$.

Aggregation of partial results. The results produced for the already gradient boosted p models $M_{opt,1}, M_{opt,2}, \dots, M_{opt,p}$ are aggregated by taking a weighted average in the form

$$\hat{y} = w_1 M_{opt,1}(\boldsymbol{x}) + w_2 M_{opt,2}(\boldsymbol{x}) + \cdots + w_p M_{opt,p}(\boldsymbol{x}) \tag{5.}$$

where $\boldsymbol{w} = [w_1, w_2, \dots, w_p]^T$ is a vector of adjustable weights used in the aggregation process; the weights are subject to optimization. In this optimization, the performance index is expressed as a sum of squared errors with the sum taken over all data $\boldsymbol{D}_{train}$. As the above optimization problem concerns a standard objective function, there is an analytical solution to the optimal weights $\boldsymbol{w}_{opt}$. The objective is to minimize the distance (sum of squared errors) between the training target ***target*** and the output of the model. With the aid of the LSE (Least square error) minimization algorithm, the optimal weight is:

$$\boldsymbol{w} = (M^T M)^{-1} M^T \boldsymbol{target} \tag{6.}$$

where M is an $N \times p$ dimensional matrix

$$M = \begin{bmatrix} M_{opt,1}(\boldsymbol{x}_1) & M_{opt,2}(\boldsymbol{x}_1) & \dots & M_{opt,p}(\boldsymbol{x}_1) \\ \vdots & \vdots & \ddots & \vdots \\ M_{opt,1}(\boldsymbol{x}_N) & M_{opt,2}(\boldsymbol{x}_N) & \dots & M_{opt,p}(\boldsymbol{x}_N) \end{bmatrix} \tag{7.}$$

where N is the number of data.

The $RMSE$ performance index evaluates the quality of the aggregation of rule-based models:

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}[target_k - \hat{y}_k]^2} \tag{8.}$$

Computing complexity. The computing complexity analysis encounters two main procedures: fuzzy clustering and matrix inversion. We investigate two models, namely the standard TS model and the distributed model with selected features. The complexity for fuzzy clustering is $O(c^2 NnI)$, where n is the number of data features and I is the number of iterations in the fuzzy clustering (in the case of standard TS model). In terms of matrix inversion, the time complexity is $O(p^3)$ (in the case of distributed model).

Statistical tests. To do the comparisons of two models, there are some statistical tests being considered [51]. One is the *Wilcoxon signed-ranks test*. It is a nonparametric test, ranking the difference in performances of two models for each data set. Let d_i be the difference between the performance indexes of two models on the *i*-th of B datasets. Then the differences will be ranked based on their absolute values. The sum of differences ranking is presented as follows

$$R^+ = \sum_{d_i>0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i) \tag{9.}$$

and

$$R^- = \sum_{d_i<0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i) \tag{10.}$$

Then we take the smaller value of the sums, i.e., $T = \min(R^+, R^-)$. The table of Wilcoxon's test indicates that in terms of the confidence level $\alpha = 0.05$ and $B = 10$ datasets, the difference between two models is significant if T is equal to or less than 8. Another one is *Paired t-test*, it checks whether the average difference of the two models is different from zero. The calculation process is presented as follows: d_i is also the difference between two models as mentioned before, the statistics is computed as $\bar{d}/\sigma_{\bar{d}}$, where $\bar{d}$ is the average value of all differences and $\sigma_{\bar{d}}$ is the standard error of the differences. The difference between two models is significant if the *t* statistics is smaller than the one required in the *t*-distribution table (note the confidence level $\alpha = 0.05$).

V. Experimental Studies

In this section, we report on the results obtained for the rule-based model designed as discussed in the previous sections and the standard TS model [1] with gradient boosting optimization (namely, standard gradient boost TS model). Since we apply the gradient boosting algorithm as a part of the proposed method, the gradient boosting is also fused in generic TS model to demonstrate the comparison performance in an objective manner. The experiments were carried out on a PC with AMD Ryzen Threadripper 2990WX 4.1GHz CPU and 64GB RAM running the MATLAB R2022b in the same environment. The performance of the model is reported in terms of its average *RMSE* with 10-fold cross-validation. The data are linearly normalized to [0,1]. In the slew of experiments, we set up the following values of the parameters:

FCM: $m = 2$, the number of iterations = 100. The number of clusters *c* varied from 2 to 10. We optimized the performance of the overall model by choosing the optimal number of clusters for each model. We also tried more values of *c* positioned in the range 2–20; no visible improvement has been reached for the values over 10. Subsequently, the range 2–10 has been selected.

Randomization: the values of *r* were selected as 0.1, 0.2, 0.3, 0.4, and 0.5. With the assumed probability, $prob = 0.999$, the results coming from the theoretical analysis in (4) are *p* = 84, 40, 25, 18, and 13 models, respectively. The random selection of data was governed by a uniform probability distribution. In the experiment, in order to ensure data integrity, we set *p* = 100 for all percentage values of *r*.

The experiments were completed for the UCI machine learning dataset Superconductivity (https://archive.ics.uci.edu/ml/datasets/superconductivty+data), consisting of 21,263 80-dimensional data. Then we separate the data into training and testing sets.

Following the overall design process, we randomly select some data and use them to train the individual fuzzy rule-based model. Then, the models are refined with the use of gradient boosting, see Fig. 3. These plots are reported for the selected percentage r being 0.5 and the number of model p is 100. It is apparent that for successive values of K, the performance index *RMSE* decreases; however, the decline is reported for some initial values of K, say 5–10 and then the values of the index stabilize. It is noticeable that the averaging of the outputs of the models leads to some improvement. Then we also aggregate these results with optimal weights following (6). In Fig. 3, the circle is the average aggregated result and the star stands for the weighted aggregation result. It is evident that the weighted aggregation leads to the better performance.

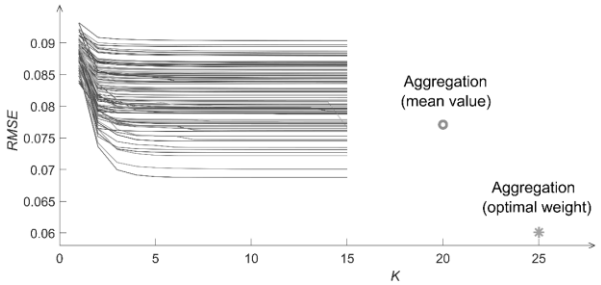


Fig. 3. Performance index *RMSE* obtained for each distributed model and the aggregation results ($r = 0.5$).

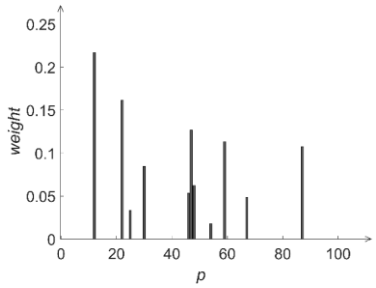


Fig. 4. Optimal values of the weights used in the weighted aggregation of the models.

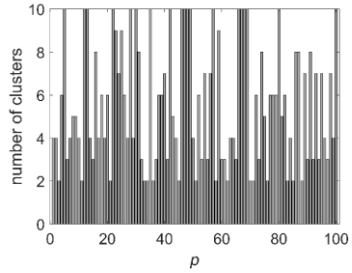
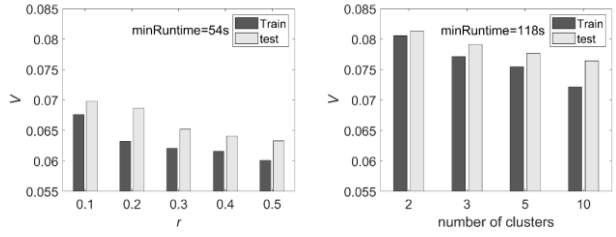
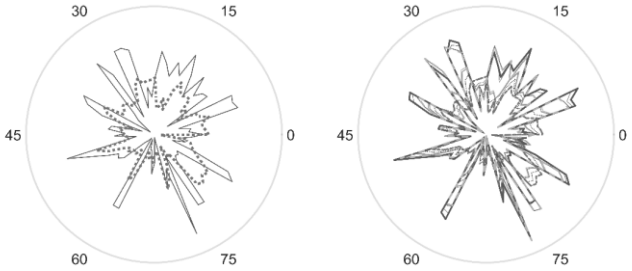


Fig. 5. Optimal numbers of the rules (clusters) in the distributed models.

By inspecting Fig. 4, we conclude that only a handful of models contribute to the aggregation process while most of them exhibit a very limited impact as the values of the corresponding weights are close to zero. In Fig. 5, it is evident that the optimal number of rules (clusters) for most distributed models vary; in most cases, these values are 2,3,4 or 10. In Fig. 6(a), we show the values of the performance index (*RMSE*) for several selected values of r.



(a) (b)



(*c*)

Fig. 6. Experimental results obtained for Super conductivity data set.
(*a*) performance index - distributed model; (*b*) performance index - TS model; (*c*) TS model -prototypes.

For comparative analysis, we consider a standard gradient boosted TS rule-based model as a reference construct, whose structure is the same as in (1). As before, FCM was run for 100 iterations, m was set to be 2 and the number of clusters is optimized in the range 2-10; the results are shown in Fig. 6(*b*). The distributed rule-based model led to the improvement over the TS model; on average (across all experiments) the improvement was around 12.8%. Considering the computational time in Figs. 6(*a*) and (*b*), the performance of the proposed model shows an improvement of 54.2%. In Fig. 6(*c*), we depict the prototypes in the form of radar plot for the TS model when c is 2 (left) and 10 (right). The prototypes are in the range [0, 1] since the data are normalized. It is difficult to distinguish the lines, which means that the prototypes are close to each other. This shows that the TS model cannot effectively cluster the data, thus affecting the performance of the model.

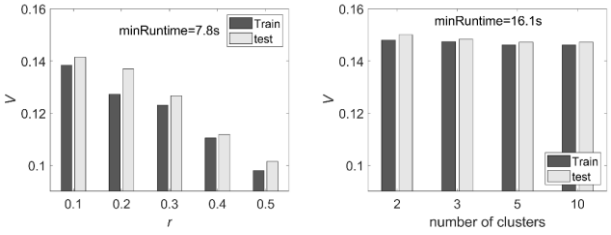
VI. Further Experimental Results With Selected Machine Learning Data

In this section, we report on experimental results obtained for some machine learning datasets coming from UCI machine learning datasets (https://archive.ics.uci.edu/ml/index.php) and Kaggle (https://www.kag-gle.com/). Our intent is to show the impact of the main parameters on the performance of the obtained models as well as to contrast the performance vis-à-vis a TS model constructed for all data. The details of the data are covered in Table 1; it is worth noting that we selected the data of the highest dimensionality of the input space as those are quite challenging in the design of rule-based models. For each dataset, in order to facilitate comparison, we focus on showing the results for the gradient boosting distributed rule-based model and the reference gradient boosted TS model, and we show the prototypes where c is equal to 2 and 10.

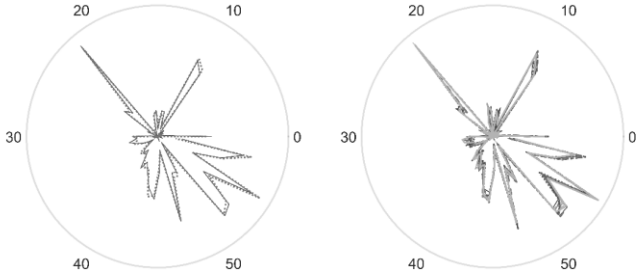
Table 1. Data sets used in experiments.

Data	(number of data, dimensionality of input space)
Online news popularity	(39,644; 58)
Year prediction MSD (first 30K data points)	(30,000; 90)
Parkinson's telemonitoring	(5,875; 17)
Geographical original of music	(1,059; 117)
SML2010	(2764; 24)
Appliance's energy prediction	(19735; 25)
Real Time Bidding (first 30K data points)	(30,000; 89)
LightGBM's regression examples	(6301; 29)
White wine quality	(4898; 12)

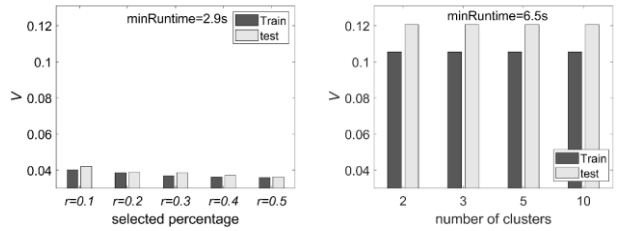
The results are displayed in a series of plots shown in Fig. 7.



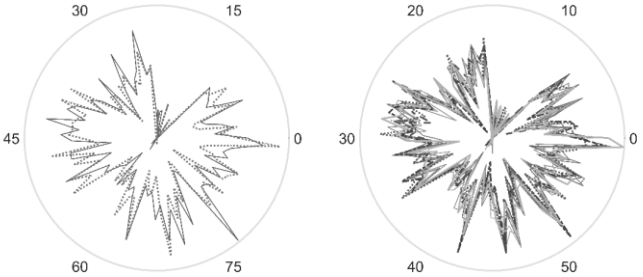
(*a*) i. (*a*) ii.



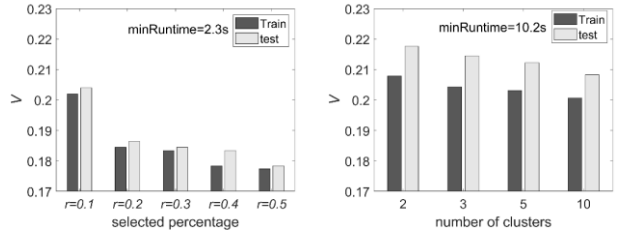
(*a*) iii.



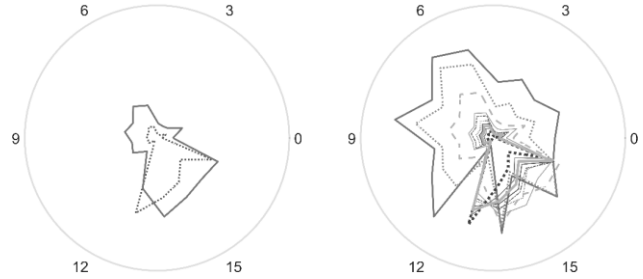
(*b*) i. (*b*) ii.



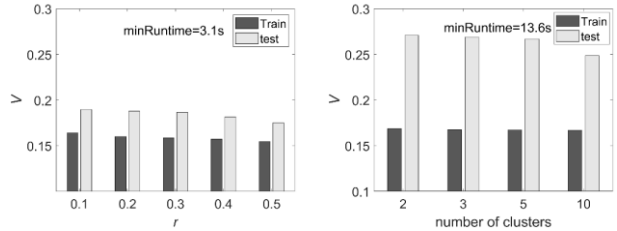
(*b*) iii.



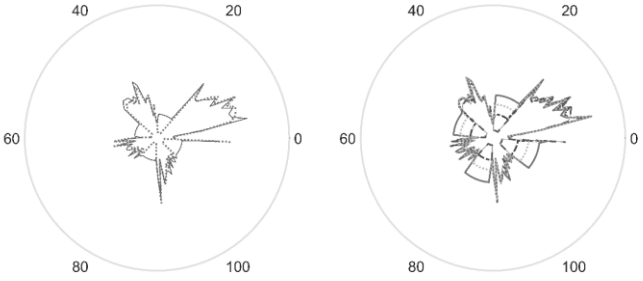
(*c*) i. (*c*) ii.



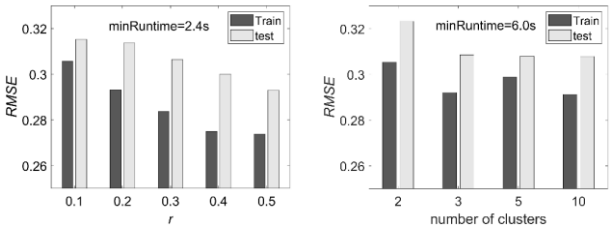
(*c*) iii.



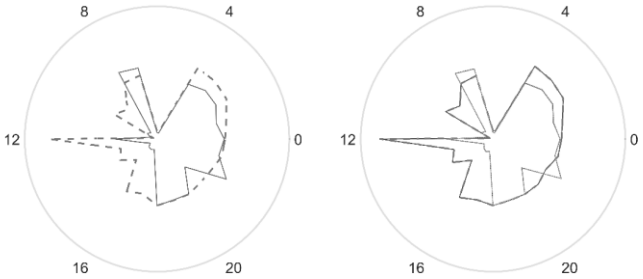
(*d*) i. (*d*) ii.



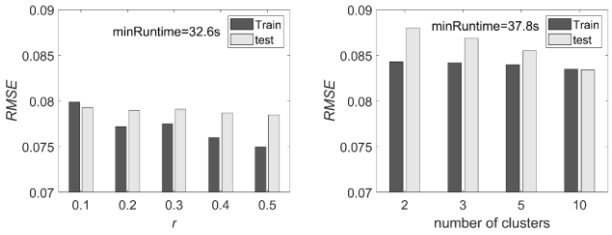
(*d*) iii.



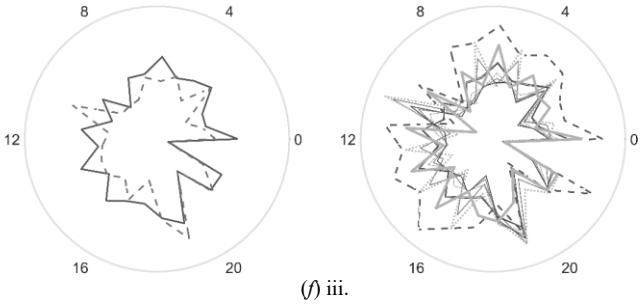
(*e*) i. (*e*) ii.



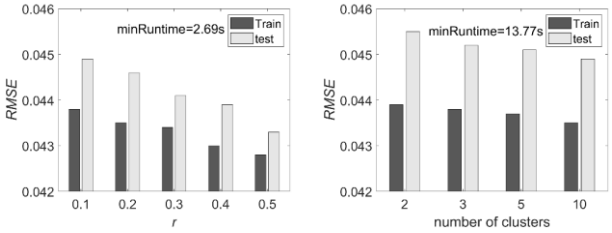
(*e*) iii.



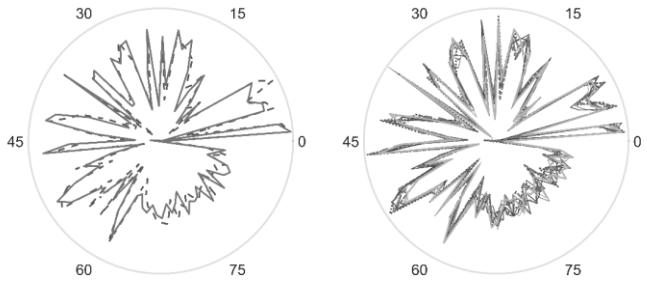
(*f*) i. (*f*) ii.



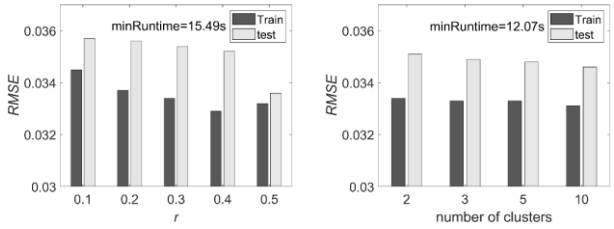
(*f*) iii.



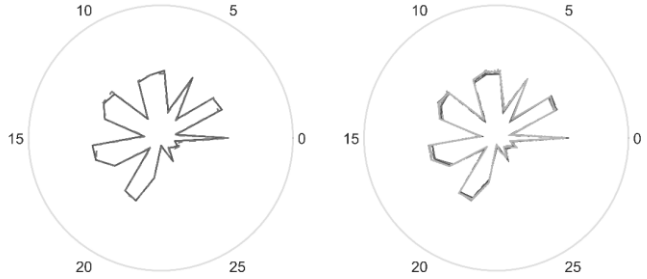
(*g*) i. (*g*) ii.



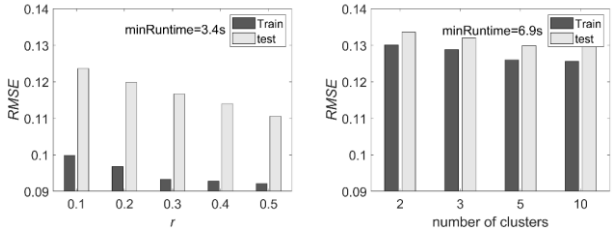
(*g*) iii.



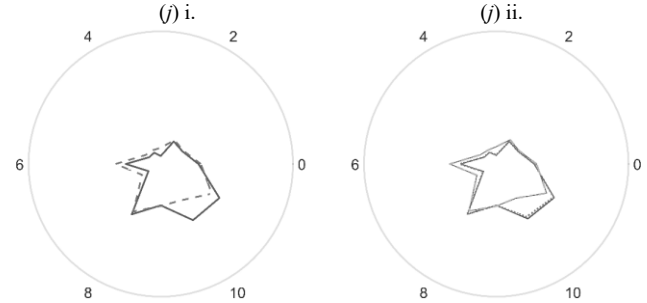
(*h*) i. (*h*) ii.



(*h*) iii.



(*j*) i. (*j*) ii.



(*j*) iii.

Fig. 7. Average results for different datasets: The plots from left to right display: i. performance index - distributed model; ii. performance index - TS model; iii. radar plots present prototypes produced by the TS model.
(*a*) *Online news popularity*; (*b*) *Year prediction MSD*; (*c*) *Parkinson's telemonitoring*; (*d*) *Geographical original of music*. (*e*) *SML2010*; (*f*) *Application energy prediction*; (*g*) *Real Time Bidding*; (*h*) *LightGBM's regression*; (*j*) *White wine quality*.

Comparing the runtime depicted in Fig. 7i and ii, our model performs better regarding most datasets. Table 2 summarizes the performance improvements of the proposed model obtained for each dataset compared with the TS model with the number of clusters set to 10 (for this number of rules, the TS model produces the best results). Compared with the gradient boosted TS model, here we present the minimum, maximum and average improvement values of the experimental results to demonstrate the achieved improvement of the proposed model. Based on the experimental results for the above 10 datasets, we compute the statistical test for our proposed model and the reference TS model. Where the Wilcoxon signed-ranks test result $T = 1$ is less than 8 and the

Paired t-test result t statistics is 2.4242 is greater than $t_{0.05,9}$ (1.8331). Both statistical tests demonstrate that our new model produces significant improvements compared to the standard gradient boosted TS model.

Table 2. Improvement obtained for each dataset

Data	Improvement based on TS model (%)					
	Train			Test		
	min	max	average	min	max	average
Superconductivity	6.4	16.8	12.9	8.8	17.3	13.5
Online news popularity	5.3	33.0	18.3	3.9	31.1	16.0
Year prediction MSD	61.9	65.8	64.4	65.2	69.9	68.0
Parkinson's telemonitoring	1.6	11.6	8.2	2.7	14.4	9.1
Geographical Original of Music	1.8	7.6	4.9	23.8	27.3	25.3
SML2010	-5.0	6.1	2.0	-2.4	4.8	1.0
Application energy prediction	4.3	10.2	7.7	5.2	5.9	5.4
Real Time Bidding	-0.6	1.6	0.5	0	3.6	1.6
LightGBM's regression	-4.2	-0.3	-1.2	-7.8	2.9	-1.4
White wine quality	20.5	26.7	24.4	4.8	14.9	9.2
Average improvement (across all data)	11.5	19. 5	16.1	11.6	20.7	16.6

Table 3. The performance of random forest model

Data	Performance index (*RMSE*)	Run time (s)
Superconductivity	train: 0.065; test: 0.069	2.87
Online news popularity	train: 0.039; test: 0.044	4.65
Year prediction MSD	train: 0.041; test: 0.043	0.5
Parkinson's telemonitoring	train: 0.191; test: 0.192	0.44
Geographical Original of Music	train: 0.189; test: 0.199	0.19
SML2010	train: 0.134; test: 0.141	0.26
Application energy prediction	train: 0.082; test: 0.085	1.75
Real Time Bidding	train: 0.045; test: 0.053	5.72
LightGBM's regression	train: 0.040; test: 0.045	0.66
White wine quality	train: 0.118; test: 0.120	0.37

In addition, we also include the comparison between the random forest regression model [52] and the proposed model. The random forest regression model is a supervised model which uses an ensemble learning method for regression (continuous data). The random forest regression is realized in four steps: 1) pick up 10% of data points and features from the training set randomly; 2) build a decision tree based on the selected data; 3) design a certain number of trees by repeating the above steps; 4) input the testing data to all trees and predict the corresponding results, then average all results to obtain the final predicted output. Regarding the random forest algorithm, we set the parameters as follows: the size of random forest T (the number of trees) is 100 and the maximum depth D of each tree is 50. The experimental results of random forest regression model are shown in Table 3. Compared with the random forest model, our model performs better over regarding most datasets (Superconductivity, Year prediction MSD, Parkinson's telemonitoring, Application energy prediction, Real Time Bidding, LightGBM's regression). Due to the low time complexity of random forest, $O(TD)$, our model has no evident advantages in terms of the running time.

VII. Conclusions

In this study, we have introduced a distributed model developed by integrating ensemble learning ideas and the gradient boosting algorithm. In this process, we demonstrate how to ensure the integrity of the data by randomly sampling. The distributed model obtained through the sampling process and the fuzzy rule-based model are further improved by the gradient boosting algorithm. In this way, we avoid the problem of the curse of dimensionality inherently present in large-dimensional data. Compared with the traditional TS model, the performance of our model has been significantly improved.

Although the feasibility of the approach is demonstrated, there are still some directions worth exploring. For example, in the random sampling stage, under the premise of ensuring data integrity, we intend to reduce the number of repetitions and find an optimal parameter combination. At the same time, one can envision applying the data with similar features to a distributed model

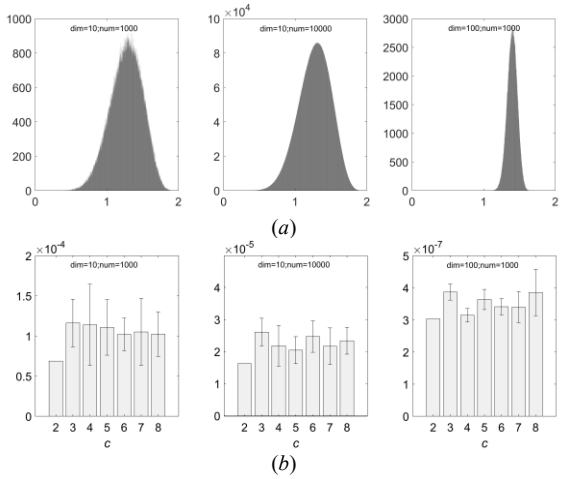
through similarity analysis where each model could be optimized with extreme gradient boosting method. In addition, in the process of optimizing the aggregation, we can consider further optimizing the model by incorporating information granularity into the parameters of the constructed model.

VIII. APPENDIX

The concentration effect states that in a high-dimensional space, the difference in distance between data points tends to become smaller as outlined as follows. To make it easy to follow, we elaborate upon an experiment with the data chosen uniformly at random from a unit sphere, where the unit sphere in $\boldsymbol{R}^n$ is defined as follows

$$\{\boldsymbol{x} \in \boldsymbol{R}^n | \|\boldsymbol{x}\| < 1\}$$

In the experiment, we compute the distances between any two data points and the distance between any two clustering centers (prototypes) which are obtained by running FCM algorithm, as shown in the following figure. In Fig. 8(a), the x-coordinate is the value of distance and y-coordinate shows the number of the data pairs generating the corresponding distance. It is obvious that with the increase of the dimensionality, the spread of distances between any two data points decreases and loses the diversity, resulting in high data similarity. However, the increase in the number of data instances has no big influence on the distribution of the distance between any two data points. In addition, we run FCM based on the chosen data and obtain several centers, then the distance between any two centers has been depicted in Fig. 8(b), the x-coordinate is the number of clusters and y-coordinate shows the mean value and standard deviation of the distances. With the increase of dimensionality, the distances between prototypes become close to each other.



(a)

(b)

Fig. 8. The distance distribution as a function of the number of data features and data instances.

REFERENCES

[1] T. Takagi and M. Sugeno, "Fuzzy identification of systems and Its applications to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985.

[2] Mohd. A. Khan and A. S. Jalal, "A fuzzy rule based multimodal framework for face sketch-to-photo retrieval," *Expert Systems with Applications*, vol. 134, pp. 138–152, Nov. 2019.

[3] P. P. Angelov and X. Gu, "Deep rule-based classifier with human-level performance and characteristics," *Information Sciences*, vol. 463–464, pp. 196–213, Oct. 2018.

[4] A. K. Paul, P. C. Shill, Md. R. I. Rabin, and K. Murase, "Adaptive weighted fuzzy rule-based system for the risk level assessment of heart disease," *Applied Intelligence*, vol. 48, no. 7, pp. 1739–1756, Jul. 2018.

[5] A. Jindal, A. Dua, N. Kumar, A. Das, A. Vasilakos, and J. Rodrigues, "Providing healthcare-as-a-service using fuzzy rule based big data analytics in cloud computing," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 5, pp. 1605–1618, Sep. 2018.

[6] H. Karunathilake, K. Hewage, W. Mérida, and R. Sadiq, "Renewable energy selection for net-zero energy communities: Life cycle based decision making under uncertainty," *Renewable Energy*, vol. 130, pp. 558–573, Jan. 2019.

[7] I. ben Ali, M. Turki, J. Belhadj, and X. Roboam, "Optimized fuzzy rule-based energy management for a battery-less PV/wind-BWRO desalination system," *Energy*, vol. 159, pp. 216–228, Sep. 2018.

[8] R. G. G. Caiado, L. F. Scavarda, L. O. Gavião, P. Ivson, D. L. de M. Nascimento, and J. A. Garza-Reyes, "A fuzzy rule-based industry 4.0 maturity model for operations and supply chain management," *International Journal of Production Economics*, vol. 231, Jan. 2021.

[9] M. Hasanipanah and H. Bakhshandeh Amnieh, "A fuzzy rule-based approach to address uncertainty in risk assessment and prediction of blast-Induced flyrock in a quarry," *Natural Resources Research*, vol. 29, no. 2, pp. 669–689, Apr. 2020.

[10] I. Škrjanc, J. A. Iglesias, A. Sanchis, D. Leite, E. Lughofer, and F. Gomide, "Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A Survey," *Information Sciences*, vol. 490, pp. 344–368, Jul. 2019.

[11] N. L. Tsakiridis, J. B. Theocharis, P. Panagos, and G. C. Zalidis, "An evolutionary fuzzy rule-based system applied to the prediction of soil organic carbon from soil spectral libraries," *Applied Soft Computing*, vol. 81, Aug. 2019, doi: https://doi.org/10.1016/j.asoc.2019.105504.

[12] M. A. Jallal, A. González-Vidal, A. F. Skarmeta, S. Chabaa, and A. Zeroual, "A hybrid neuro-fuzzy inference system-based algorithm for time series forecasting applied to energy consumption prediction," *Applied Energy*, vol. 268, Jun. 2020, doi: 10.1016/j.apenergy.2020.114977.

[13] W. Dong, Q. Yang, X. Fang, and W. Ruan, "Adaptive optimal fuzzy logic based energy management in multi-energy microgrid considering operational uncertainties," *Applied Soft Computing*, vol. 98, Jan. 2021, doi: 10.1016/j.asoc.2020.106882.

[14] X. Xie, C. Wei, Z. Gu, and K. Shi, "Relaxed resilient fuzzy stabilization of discrete-time Takagi-Sugeno systems via a higher order Time-Variant balanced matrix method," *IEEE Transactions on Fuzzy Systems*, 2022, doi: 10.1109/TFUZZ.2022.3145809.

[15] D. Wang and J. Gu, "VASC: Dimension reduction and visualization of single-cell RNA-seq data by deep variational autoencoder," *Genomics, Proteomics and Bioinformatics*, vol. 16, no. 5, pp. 320–331, Oct. 2018.

[16] F. J. Pulgar, F. Charte, A. J. Rivera, and M. J. del Jesus, "Choosing the proper autoencoder for feature fusion based on data complexity and classifiers: Analysis, tips and guidelines," *Information Fusion*, vol. 54, pp. 44–60, Feb. 2020.

[17] G. Boquet, A. Morell, J. Serrano, and J. L. Vicario, "A variational autoencoder solution for road traffic forecasting systems: Missing data imputation, dimension reduction, model selection and anomaly detection," *Transportation Research Part C: Emerging Technologies*, vol. 115, Jun. 2020, doi: 10.1016/j.trc.2020.102622.

[18] J. Zhao *et al.*, "Attribute mapping and autoencoder neural network based matrix factorization initialization for recommendation systems," *Knowledge-Based Systems*, vol. 166, pp. 132–139, Feb. 2019.

[19] S. Ryu, H. Choi, H. Lee, and H. Kim, "Convolutional autoencoder based feature extraction and clustering for customer load analysis," *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 1048–1060, Mar. 2020.

[20] H. E, Y. Cui, W. Pedrycz, and Z. Li, "Fuzzy relational matrix factorization and its granular characterization in data description," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 3, pp. 794–804, Mar. 2022, [Online]. Available: https://ieeexplore.ieee.org/document/9311791/

[21] Y. Yang, A. Ming, Y. Zhang, and Y. Zhu, "Discriminative non-negative matrix factorization (DNMF) and its application to the fault diagnosis of diesel engine," *Mechanical Systems and Signal Processing*, vol. 95, pp. 158–171, Oct. 2017.

[22] L. Zhang, L. Zhang, B. Du, J. You, and D. Tao, "Hyperspectral image unsupervised classification by robust manifold matrix factorization," *Information Sciences*, vol. 485, pp. 154–169, Jun. 2019.

[23] Z. Li, J. Tang, and X. He, "Robust structured nonnegative matrix factorization for image representation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1947–1960, May 2018.

[24] X. Kang, X. Xiang, S. Li, and J. A. Benediktsson, "PCA-based edge-preserving features for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7140–7151, Dec. 2017.

[25] C. Happ and S. Greven, "Multivariate functional principal component analysis for data observed on different (dimensional) domains," *J Am Stat Assoc*, vol. 113, no. 522, pp. 649–659, Apr. 2018.

[26] S. Yi, Z. Lai, Z. He, Y. ming Cheung, and Y. Liu, "Joint sparse principal component analysis," *Pattern Recognition*, vol. 61, pp. 524–536, Jan. 2017.

[27] J. Wu, Z. Wu, J. Cao, H. Liu, G. Chen, and Y. Zhang, "Fuzzy consensus clustering with applications on big Data," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 6, pp. 1430–1445, Dec. 2017.

[28] A. Segatori, F. Marcelloni, and W. Pedrycz, "On distributed fuzzy decision trees for big data," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 1, pp. 174–192, Feb. 2018.

[29] S. Salloum, Z. Huang, and Y. He, "Random sample partition: A distributed data model for big data analysis," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 11, pp. 5846–5854, Nov. 2019.

[30] G. Tanaka *et al.*, "Recent advances in physical reservoir computing: A review," *Neural Networks*, vol. 115, pp. 100–123, Jul. 2019.

[31] B. V. Dasarathy and B. V. Sheela, "A composite classifier system design: Concepts and methodology," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 708–713, 1979.

[32] M. M. Bejani and M. Ghatee, "A context aware system for driving style evaluation by an ensemble learning on smartphone sensors data," *Transportation Research Part C: Emerging Technologies*, vol. 89, pp. 303–320, Apr. 2018.

[33] D. T. Bui, P. Tsangaratos, P. T. T. Ngo, T. D. Pham, and B. T. Pham, "Flash flood susceptibility modeling using an optimized fuzzy rule based feature selection technique and tree based ensemble methods," *Science of the Total Environment*, vol. 668, pp. 1038–1054, Jun. 2019.

[34] X. Hu, W. Pedrycz, and X. Wang, "Random ensemble of fuzzy rule-based models," *Knowledge-Based Systems*, vol. 181, p. 104768, Oct. 2019.

[35] J. Kerr-Wilson and W. Pedrycz, "Some new qualitative insights into quality of fuzzy rule-based models," *Fuzzy Sets and Systems*, vol. 307, pp. 29–49, Jan. 2017.

[36] J. Kerr-Wilson and W. Pedrycz, "Generating a hierarchical fuzzy rule-based model," *Fuzzy Sets and Systems*, vol. 381, pp. 124–139, Feb. 2020.

[37] J. Kerr-Wilson and W. Pedrycz, "Design of rule-based models through information granulation," *Expert Systems with Applications*, vol. 46, pp. 274–285, Mar. 2016.

[38] L. Breiman, "Arcing classifiers," *The Annals of Statistics*, vol. 26, no. 3, pp. 801–849, Jun. 1998.

[39] P. Kumari and D. Toshniwal, "Extreme gradient boosting and deep neural network based ensemble learning approach to forecast hourly solar irradiance," *Journal of Cleaner Production*, vol. 279, Jan. 2021, doi: 10.1016/j.jclepro.2020.123285.

[40] T. Zhang, W. He, H. Zheng, Y. Cui, H. Song, and S. Fu, "Satellite-based ground PM2.5 estimation using a gradient boosting decision tree," *Chemosphere*, vol. 268, Apr. 2021, doi: 10.1016/j.chemosphere.2020.128801.

[41] Y.-C. Chang, K.-H. Chang, and G.-J. Wu, "Application of extreme gradient boosting trees in the construction of credit risk assessment models for financial institutions," *Applied Soft Computing*, vol. 73, pp. 914–920, Dec. 2018.

[42] H. Verbois, A. Rusydi, and A. Thiery, "Probabilistic forecasting of day-ahead solar irradiance using quantile gradient boosting," *Solar Energy*, vol. 173, pp. 313–327, Oct. 2018.

[43] 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, Aug. 1997.

[44] D. C. Feng *et al.*, "Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach," *Construction and Building Materials*, vol. 230, Jan. 2020, doi: 10.1016/j.conbuildmat.2019.117000.

[45] L. Wang, Y. Ngan, and H. Yung, "Automatic incident classification for large-scale traffic data by adaptive boosting SVM," *Information Sciences*, vol. 467, pp. 59–73, Oct. 2018.

[46] Q. Zhang, W. Hu, Z. Liu, and J. Tan, "TBM performance prediction with bayesian optimization and automated machine learning," *Tunnelling and Underground Space Technology*, vol. 103, Sep. 2020.

[47] G. Hu, C. Yin, M. Wan, Y. Zhang, and Y. Fang, "Recognition of diseased Pinus trees in UAV images using deep learning and AdaBoost classifier," *Biosystems Engineering*, vol. 194, pp. 138–151, Jun. 2020.

[48] S. W. Knox, *Machine learning: a concise introduction*, vol. 285. Hoboken, NJ, USA: Wiley & Sons, 2018, pp. 180–184.

[49] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How many trees in a random forest?" in *Machine Learning and Data Mining in Pattern Recognition*, 2012, pp. 154–168.

[50] J. H. Friedman, "Greedy function approximation: a gradient boosting machines," *Ann Stat*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[51] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, Aug. 2006.

[52] U. Grömping, "Variable importance assessment in regression: linear regression versus random forest," *The American Statistician*, vol. 63, no. 4, pp. 308–319, Sep. 2008.