A Micro-Extended Belief Rule-Based System for Big Data Multi-Class Classification Problems

Long-Hao Yang, Jun Liu, *Member*, *IEEE*, Ying-Ming Wang, and Luis Martínez, *Member*, *IEEE*

***Abstract*—Big data classification problems have drawn great attention from diverse fields and many classifiers have been developed. Among those classifiers, the extended belief rule-based system (EBRBS) has shown its potential in both big data and multi-class situations while time complexity and computing efficiency are two challenging issues to be handled in EBRBS. As such, three improvements of EBRBS are proposed firstly in the present paper to decrease the time complexity and computing efficiency of EBRBS for multi-class classification under the assumption of large amount of data, including the strategy to skip rule weight calculation, a simplified evidential reasoning algorithm, and the domain division-based rule reduction method. This turns out to be a micro version of the EBRBS, classed *Micro-EBRBS*. Moreover, one of commonly used cluster computing, named *Apache Spark*, is then applied to implement the parallel rule generation and inference schemes of the Micro-EBRBS for big data multi-class classification problems. The comparative analyses of experimental studies demonstrate that the Micro-EBRBS not only can obtain a desired accuracy, but also has the comparatively better time complexity and computing efficiency than some popular classifiers, especially for multi-class classification problems.**

***Index Terms*—Apache Spark, Big data, Extended belief rule-based system (EBRBS), Multi-class,**

I. Introduction

CLASSIFICATION problems are the common and fundamental ones involved in various real-world applications, such as intrusion detection [1], pattern recognition [19], image processing [2], and DNA sequence classification [5]. Classification becomes much more complex under big data and multi-class situations. The former always implies a high requirement of the computing efficiency of classifiers, while the need of a cluster computing for implementing the classifiers is getting popular. The latter implies many overlaps among the data of different classes, which requires the classifiers to have a powerful ability to differentiate the class boundaries.

Among many methodologies for multi-class classification problems, such as support vector machine (SVM) [34], ensemble learning [21], and others, rule-based systems (RBSs) are one kind of useful tools and have been a popular framework for designing classifiers in past decades. Basically, these RBSs can be divided into two categories depending on the construction methods of rule bases: RBSs based on the optimization model and iterative algorithm to determine the optimal values of parameters involved [33], [36], [39], or RBSs where rules are generated from sample data without the optimal model [8], [22], [28]. While handling big data multi-class classification problems, it is obvious that the RBSs without the optimal model are a better choice because of its high computing efficiency [26].

Some popular RBSs without the optimal model include fuzzy rule-based classification system (FRBCS) based on Chi *et al*. algorithm (Chi-FRBCS) [8] and the extended belief rule-based system (EBRBS) proposed by Liu *et al*. [22], both of them are originated from the work by Wang and Mendel [31]. It is worth noting that Chi-FRBCS has been applied to deal with big data classification problems recently (see details in Section II-B). Although there are some existing works about EBRBS [6], [11], [38], [39], it has not been developed yet as an efficient classifier to deal with big data classification problems. Hence, the goal of this study is to propose a novel big data EBRBS classifier and show its performance, in terms of accuracy and efficiency, in big data multi-class classification problems.

EBRBS has shown its potential to address multi-class classification problems because of its rules with belief structure and the evidential reasoning (ER) algorithm used in the inference scheme to collectively handle multi-class information for classification [20], [23], [24], [25], however, two challenges must be addressed to handle the big data situation:

Manuscript received February 2, 2018; accepted September 11, 2018. This work was supported by the National Natural Science Foundation of China (Nos. 61773123, 71371053, 71501047, 71701050, and 71801050), the Humanities and Social Science Foundation of the Ministry of Education under Grant (No. 14YJC630056), and the Natural Science Foundation of Fujian Province, China (No. 2015J01248). This paper was recommended by Associate Editor XXXX (*Corresponding authors: Ying-Ming Wang*).

L.-H. Yang is with Decision Sciences Institute, Fuzhou University, Fuzhou, PR China, and also with Department of Computer Science, University of Jaén, Jaén, Spain (e-mail: more026@hotmail.com).

J. Liu is with School of Computing, Ulster University at Jordanstown Campus, Northern Ireland, UK (e-mail: j.liu@ulster.ac.uk).

Y.-M. Wang is with Decision Sciences Institute, Fuzhou University, Fuzhou, PR China, and also with Key Laboratory of Spatial Data Mining & Information Sharing of Ministry of Education, Fuzhou University, Fuzhou, RP China (e-mail: msymwang@hotmail.com).

L. Martínez is with Department of Computer Science, University of Jaén, Jaén, Spain (email: martin@ujaen.es)

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

Digital Object Identifier XXXXXXX

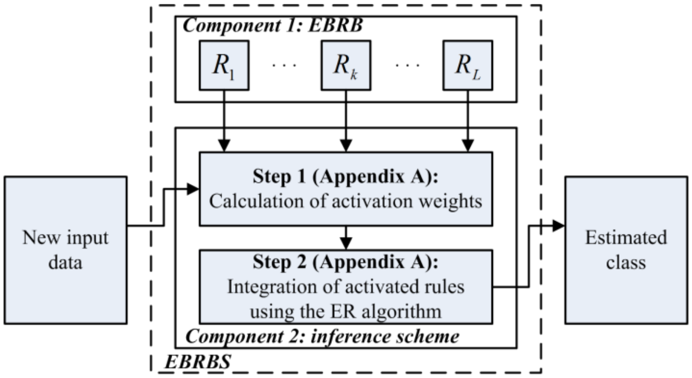

Component 1: EBRB
R_1 · · · R_k · · · R_L
New input data
Step 1 (Appendix A): Calculation of activation weights
Step 2 (Appendix A): Integration of activated rules using the ER algorithm
Component 2: inference scheme
Estimated class
EBRBS

Fig. 1. Methodological framework of EBRBS for classification

(1) The time complexity of the EBRBS must be reduced to ensure the high computing efficiency under big data problems.

The proposed solution in the present paper is to optimize the procedures of the EBRBS so that its time complexity can be reduced, which are mainly focused on three key procedures of the EBRBS: rule weight calculation, the ER algorithm, and the size of rule base. The corresponding improvements are proposed respectively, which forms a micro version of the EBRBS with much higher computing efficiency enables to deal with big data multi-class classification problems, called the *Micro-EBRBS*.

(2) The cluster computing must be applied to implement the parallel computing to improve the computing efficiency.

Chi-FRBCS classifiers usually involve the cluster computing to deal with big data classification problems [10], [26], [29]. Likewise, it is necessary to propose a solution of implementing the EBRBS classifier by using the cluster computing to handle the big data problem. Apache Spark [4] is an open-source framework that supports the processing of large datasets in a distributed computing environment and provides primitives for in-memory cluster computing and APIs in Scala, Java, and Python. As such, Apache Spark-based implementation of the parallel rule generation and inference scheme is proposed to improve the computing efficiency of Micro-EBRBS in big data situation.

To verify the effectiveness and computing efficiency of the Micro-EBRBS, three experiments based on 14 classification datasets, in which 4 of these datasets have relatively large number of data, are carried out to test the performance of the Micro-EBRBS. Two main aspects, namely accuracy and computing time, are used to compare the Micro-EBRBS with the EBRBS, the conventional FRBCS and machine-learning classifiers, and the big data FRBCS classifiers.

The remainder of this paper is organized as follows: Section II briefly reviews the background and challenges of the EBRBS for classification. Section III introduces the Micro-EBRBS for big data multi-classification problems. Section IV discusses experiments to demonstrate the performance of the Micro-EBRBS, and the paper is concluded in Section V.

II. Background and Challenges

In this section, the EBRBS for classification problems is reviewed firstly to provide the basic knowledge of this study.

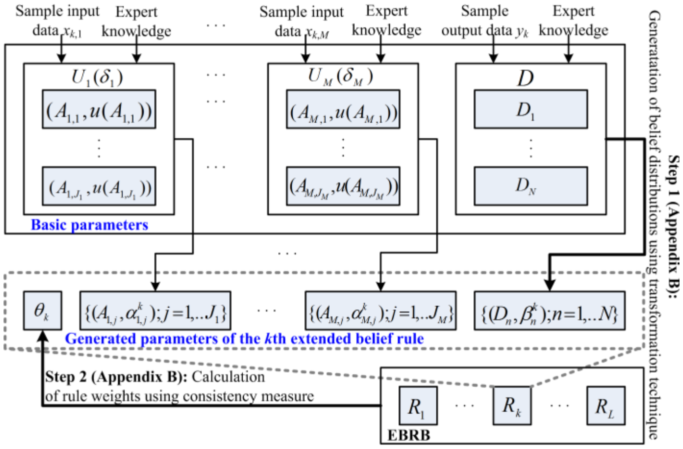

Sample input data $x_{k,1}$
Expert knowledge
Sample input data $x_{k,M}$
Expert knowledge
Sample output data y_k
Expert knowledge
$U_1(\delta_1)$
$(A_{1,1},u(A_{1,1}))$
$(A_{1,J_1},u(A_{1,J_1}))$
$U_M(\delta_M)$
$(A_{M,1},u(A_{M,1}))$
$(A_{M,J_M},u(A_{M,J_M}))$
D
D_1
D_N
Basic parameters
θ_k
$\{(A_{1,j},\alpha^k_{1,j});j=1,..J_1\}$
$\{(A_{M,j},\alpha^k_{M,j});j=1,..J_M\}$
$\{(D_n,\beta^k_n);n=1,..N\}$
Generated parameters of the kth extended belief rule
Step 2 (Appendix B): Calculation of rule weights using consistency measure
R_1 · · · R_k · · · R_L
EBRB
Step 1 (Appendix B): Generatation of belief distributions using transformation technique

Fig. 2. Rule generation scheme of EBRBS

Secondly, the related works of the Chi-FRBCS for big data problems are reviewed for the sake of comparison with the proposed Micro-EBRBS. Finally, the time complexity of the EBRBS is discussed to clarify the challenges to face for the novel classifier under big data and multi-class situation.

A. *EBRBS for classification problems*

An EBRBS consists of two components: the extended belief rule base (EBRB) and the inference scheme. The former can be regarded as a knowledge base to store a set of rules with uncertainty. The latter provides an inference engine to infer new results based on the interaction of test input and the EBRB.

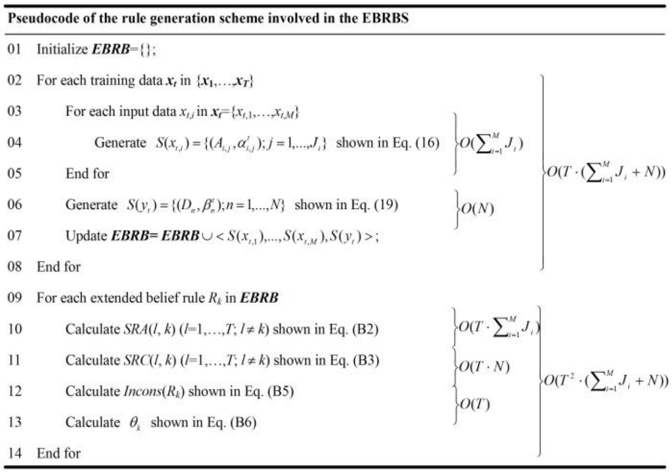
Basically, an EBRB for classification problems includes M antecedent attributes and one consequent attribute, in which each antecedent attribute U_i (i=1,…, M) has an attribute weight δ_i ($0<\delta_i\leq 1$) and J_i reference values $A_{i,j}$ (j=1,…, J_i) which are used as the discrete and representative evaluation grades for describing the ith antecedent attribute; and the consequent attribute D has N classes D_n (n=1,…, N). Thus, the kth (k=1,…, L) extended belief rule (EBR) R_k in the EBRB can be written as:

$$\begin{aligned}R_k : & IF\ U_1\ is\ \{(A_{1,j},\alpha^k_{1,j}); j=1,...,J_1\}\wedge\cdots\wedge U_M\ \ is \\ & \{(A_{M,j},\alpha^k_{M,j}); j=1,...,J_M\}, THEN\ D\ is \\ & \{(D_n,\beta_{n,k}); n=1,...,N\}, with\ \theta_k\ \ and\ \{\delta_1,...,\delta_M\}\end{aligned} \qquad (1)$$

where $\alpha^k_{i,j}$ ($0\leq\alpha^k_{i,j}\leq 1$) and $\beta_{n,k}$ ($0\leq\beta_{n,k}\leq 1$) are the belief degree to the reference value $A_{i,j}$ and the class D_n in the kth rule with $\sum_{j=1}^{J_i}\alpha^k_{i,j}\leq 1$ and $\sum_{n=1}^{N}\beta_{n,k}\leq 1$; {($A_{i,j}$, $\alpha^k_{i,j}$); j=1,…, J_i} and {(D_n, $\beta_{n,k}$); n=1,…, N} are the belief structure embedding in the kth rule; θ_k ($0<\theta_k\leq 1$) is the weight of the kth rule.

Remark 1: By assuming that: (1) $A_{i,j}$ (i=1,.., M; j=1,…, J_i) is a linguistic label modeled by a triangular membership function; (2) $\alpha^k_{i,j}$ =1 (k=1,…, L) and $\alpha^k_{i,t}$ =0(t=1,…, J_i; $t\neq j$); (3) $\beta_{n,k}$ =1 and $\beta_{n,s}$ =0 (s=1,…, N; $s\neq n$); (4) δ_i =1 (i=1,…, M), the kth EBR becomes a fuzzy rule in the Chi-FRBCS [29]:

$$\begin{aligned}R_k : & IF\ U_1\ is\ A_{1,j}\wedge\cdots\wedge U_M\ \ is\ A_{M,j}, \\ & THEN\ D\ is\ D_n, with\ \theta_k\end{aligned} \qquad (2)$$



Pseudocode of the rule generation scheme involved in the EBRBS

```
Initialize EBRB={};
For each training data $\mathbf{x}_t$ in $\{\mathbf{x}_1,\dots,\mathbf{x}_T\}$
    For each input data $x_{t,i}$ in $\mathbf{x}_t=\{x_{t,1},\dots,x_{t,M}\}$
        Generate $S(x_{t,i})=\{(A_{i,j},\alpha_{i,j}^t); j=1,\dots,J_i\}$ shown in Eq. (16)
    End for
    Generate $S(y_t)=\{(D_n,\beta_n^t); n=1,\dots,N\}$ shown in Eq. (19)
    Update EBRB= EBRB $\cup <S(x_{t,1}),\dots,S(x_{t,M}),S(y_t)>$;
End for
For each extended belief rule $R_k$ in EBRB
    Calculate $SRA(l,k)$ ($l$=1,…,$T$; $l \neq k$) shown in Eq. (B2)
    Calculate $SRC(l,k)$ ($l$=1,…,$T$; $l \neq k$) shown in Eq. (B3)
    Calculate $Incons(R_k)$ shown in Eq. (B5)
    Calculate $\theta_k$ shown in Eq. (B6)
End for
```

$O(\sum_{i=1}^{M} J_i)$
$O(N)$
$O(T \cdot (\sum_{i=1}^{M} J_i + N))$
$O(T \cdot \sum_{i=1}^{M} J_i)$
$O(T \cdot N)$
$O(T)$
$O(T^2 \cdot (\sum_{i=1}^{M} J_i + N))$

Fig. 3. Pseudocode and time complexity of the rule generation scheme

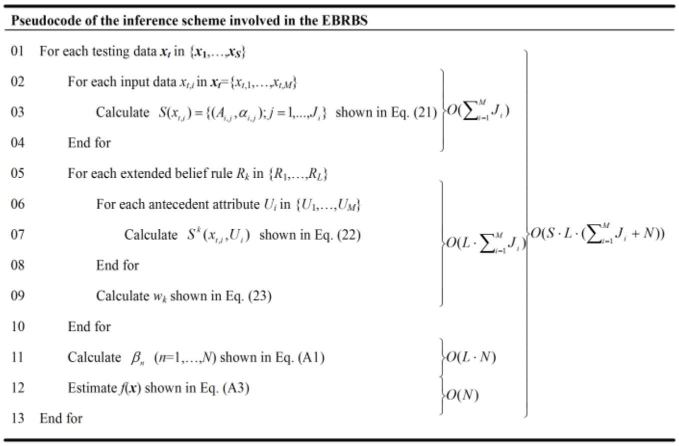
It is clear from the comparison of Eqs. (1) and (2) that the fuzzy rule is a special case of the EBR. Moreover, the EBR is more flexible to express multi-class information under uncertainty and incompleteness, e.g., {(D_1, 0.6), (D_2, 0.4)} means 60% sure that the class is D_1, 40% sure that it is D_2. $\sum_{j=1}^{J_i} \alpha_{i,j}^{k}$ =0.9 means 100% - 90% = 10% ignorance in the *i*th antecedent attribute of the *k*th EBR.

Based on the above EBRB, the ER algorithm based inference scheme is applied to integrate EBRs to produce estimated classes, i.e., the integrated result is belief distribution {(D_1, 0.4008), (D_2, 0.4275), (D_3, 0.1718)} and finally produces the class D_2 as the output. A simple methodological framework of the EBRBS for classification is shown in Fig. 1 and the detailed step procedure can be referred to [22]. Additionally, the rule generation scheme is an indispensable part of the EBRBS and illustrated in Fig. 2.

As shown in Fig. 2, there are two kinds of parameters involved in the rule generation scheme of the EBRB. The first one is named as the basic parameters, including attribute weights, reference values and utility values of antecedent attributes, and classes of the consequent attribute. All these basic parameters are always determined by using expert knowledge. The second one is the generated parameters, including rule weights and belief distributions of antecedent and consequent attributes. All these generated parameters have to be initialized according to the sample input-output data and the basic parameters, including two steps: 1) generation of belief distributions for antecedent and consequent attributes using transformation techniques; and 2) calculation of rule weights using consistency measures. The detailed description of those steps can be found in Appendix B.

B. *Chi-FRBCS in big data classification problems*

FRBCSs are popular methods for classification problems with many versions developed so far, e.g., Chi-FRBCS [8], structural learning algorithm on vague environment (SLAVE) [15], fuzzy hybrid genetic-based machine learning algorithm (FH-GBML) [18], fuzzy unordered rule induction algorithm (FURIA) [16], and fuzzy association rule-based classification method for high-dimensional problems (FARC-HD) [3]. However, considering the limitations of standard fuzzy rule base learning approaches for large number of samples, the Chi-FRBCS was recognized by many researchers as the suitable FRBCS to handle big data classification problems [26], [29].



Pseudocode of the inference scheme involved in the EBRBS

```
For each testing data $\mathbf{x}_t$ in $\{\mathbf{x}_1,\dots,\mathbf{x}_S\}$
    For each input data $x_{t,i}$ in $\mathbf{x}_t=\{x_{t,1},\dots,x_{t,M}\}$
        Calculate $S(x_{t,i})=\{(A_{i,j},\alpha_{i,j}); j=1,\dots,J_i\}$ shown in Eq. (21)
    End for
    For each extended belief rule $R_k$ in $\{R_1,\dots,R_L\}$
        For each antecedent attribute $U_i$ in $\{U_1,\dots,U_M\}$
            Calculate $S^k(x_{t,i},U_i)$ shown in Eq. (22)
        End for
        Calculate $w_k$ shown in Eq. (23)
    End for
    Calculate $\beta_n$ ($n$=1,…,$N$) shown in Eq. (A1)
    Estimate $f(\mathbf{x})$ shown in Eq. (A3)
End for
```

$O(\sum_{i=1}^{M} J_i)$
$O(L \cdot \sum_{i=1}^{M} J_i)$
$O(S \cdot L \cdot (\sum_{i=1}^{M} J_i + N))$
$O(L \cdot N)$
$O(N)$

Fig. 4. Pseudocode and time complexity of the inference scheme

In the last few years, several big data classifiers based on the Chi-FRBCS have been proposed and made use of the Apache Hadoop to deploy the distributed system. For example, Lopez *et al*. [26] proposed the first FRBCS capable of addressing big data and imbalance datasets, called *Chi-FRBCS-BigDataCS*, which utilized the Apache Hadoop to distribute the computational operations of Chi-FRBCS and also included cost-sensitive learning techniques to address imbalanced big data. After that, Rio *et al*. [29] developed a more general big data classifier based on the Chi-FRBCS and Apache Hadoop, called *Chi-FRBCS-BigData*, which includes two versions: *Chi-FRBCS-BigData-Max* and *Chi-FRBCS-BigData-Ave*. Both show the ability to deal with big data problems providing competitive results and reasonable computing efficiencies. Fernandez *et al*. [12] studied the relationship between the granularity and data scattering for Chi-FRBCS in big data classification problems and made use of Chi-FRBCS-BigData to accomplish their analysis. Later on, Fernandez *et al*. [13] carried out many experimental studies regarding the use of Chi-FRBCS-BigData to analyze the differences in performance with respect to the lack of data for the learning stage, how rules are distributed among Maps, and their influences on the classification stage. Elkano *et al*. [10] proposed a global version of Chi-FRBCS-BigDataCS in order to address the problem of previous big data classifiers that it would become less accurate when more computing nodes are added into the cluster.

The above literatures review shows many potential applications of Chi-FRBCS in big data classification problems. However, they were mainly focused on two-class classification problems. Although the decomposition strategies [27], such as One-Versus-One (OVO) and One-Versus-All (OVA) schemes, can be used to decompose a multi-class problem into multiple two-class problems, it is unavoidable to cause the increase of the time complexity. Considering that the EBRBS has an effective rule representation scheme better than the one in Chi-FRBCS owning to embedding the belief structure into both

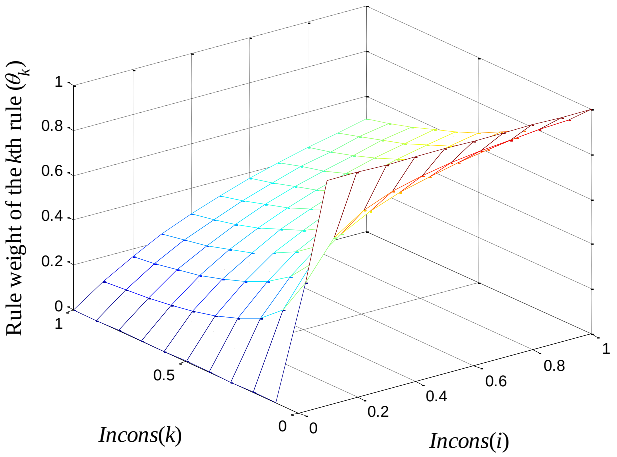


Fig. 5. Relationship between the rule weights and their inconsistency degrees

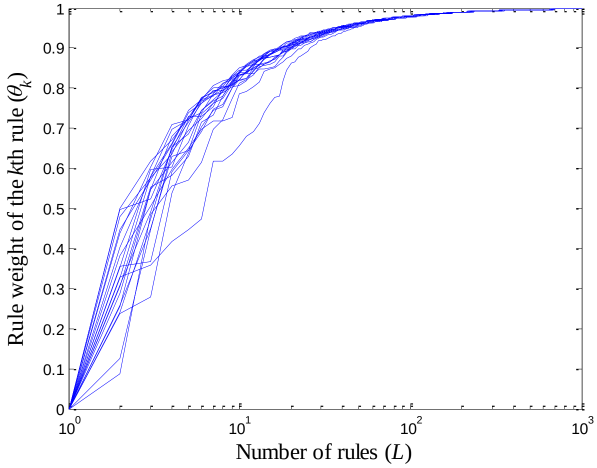


Fig. 6. Relationship between the rule weights of R_k and the number of rules

antecedent and consequent attributes, the present work aims to propose a novel classifier based on the EBRBS for big data problems in comparison with Chi-FRBCS in terms of accuracy and efficiency.

C. Challenges of EBRBS for big data problems

Due to the importance of computing efficiency in big data problems, in the context of the EBRBS, the time complexity of the rule generation and inference scheme are analyzed in this subsection. For the discussion purposes, Figs. 3 and 4 provide the pseudocode and the time complexity of those two schemes involved in the EBRBS, respectively.

From Fig. 3, the time complexity of generating belief distributions shown in the 1st line to the 8th line is $O(T\times(\Sigma_{i=1,\ldots,M}J_i + N)$ and the time complexity of calculating rule weights shown in the 9th line to the 14th line is $O(T^2\times(\Sigma_{i=1,\ldots,M}J_i + N)$, where T is the number of sample data, M is the number of antecedent attributes, J_i is the number of reference values in the ith antecedent attribute, and N is the number of classes. Clearly, for the rule generation scheme of the EBRBS, the calculation of rule weights requires the most computing time and sometimes it would be unacceptable while there are large amounts of sample data. For example, the dataset Poker has 1,025,010 samples (i.e., T=1,025,010), 10 attributes (i.e., M=10), and 10 classes (i.e., N=10). If the number of reference values is assumed to be 3 for each attribute (i.e., J_i=3; i=1,…, M) and the computing time of each operation is 10^{-6} second, then the total computing time of generating rule weights is 11,673.8 hours.

From Fig. 4, the time complexity of the inference scheme is $O(S\times L\times(\Sigma_{i=1,\ldots,M}J_i + N)$, where S is the number of test data, L is the number of EBRs in the EBRB. Considering that one EBR is directly transformed from one sample data [22], so T=L. Therefore the time complexity of the inference scheme can be expressed as $O(S\times T\times(\Sigma_{i=1,\ldots,M}J_i + N)$. Obviously, the computing time would also be unacceptable while there is large amount of sample data involved in the rule generation scheme, i.e., while the 10-fold cross validation is utilized to test the dataset Poker, the number of sample data and test data is therefore 922,509 and 102,501, respectively. Finally, the computing time of the inference scheme is 1,050.6 hours.

The above discussions clearly show that although the EBRBS is the RBS without the optimal model, the time-consuming process found in the calculation of rule weights and the inference scheme would be serious challenges while the EBRBS is applied to address the classification problem with a large amount of data. Therefore, the present work aims to address these challenges.

III. A Novel EBRBS for Big data multi-class classification problems

According to the challenges pointed out in Section II-C, the possible approaches are investigated to reduce the time complexity of the rule generation and the inference schemes, followed by a new rule reduction method to downsize the EBRB. Based on these achievements, a novel EBRBS and its Apache Spark-based implementation are developed to deal with big data multi-class classification problems.

A. Analysis of rule weight calculation and the ER algorithm involved in EBRBS

In order to reduce the time complexity of the EBRBS, the properties of the rule weight calculation and the ER algorithm are investigated as follows.

Theorem 1. The rule weight of each EBR will approximate to 1 while using large number of sample data to generate EBRs.

Proof. Suppose there are L EBRs. Based on the calculation of rule weights in Eq. (B6) in Appendix B, we can get the first order partial derivative of the rule weight θ_k (k=1,…, L) with respect to the inconsistency degree *Incons*(R_k) as follows:

$$\frac{\partial\theta_k}{\partial Incons(R_k)} = -\frac{\sum_{l=1}^{L} Incons(R_l) - Incons(R_k)}{\left[Incons(R_k)+\sum_{l=1,l\neq k}^{L} Incons(R_l)\right]^2} = -\frac{\sum_{l=1,l\neq k}^{L} Incons(R_l)}{\left[Incons(R_k)+\sum_{l=1,l\neq k}^{L} Incons(R_l)\right]^2} < 0 \qquad (3)$$

It follows that θ_k decreases while its *Incons*(R_k) increases. In

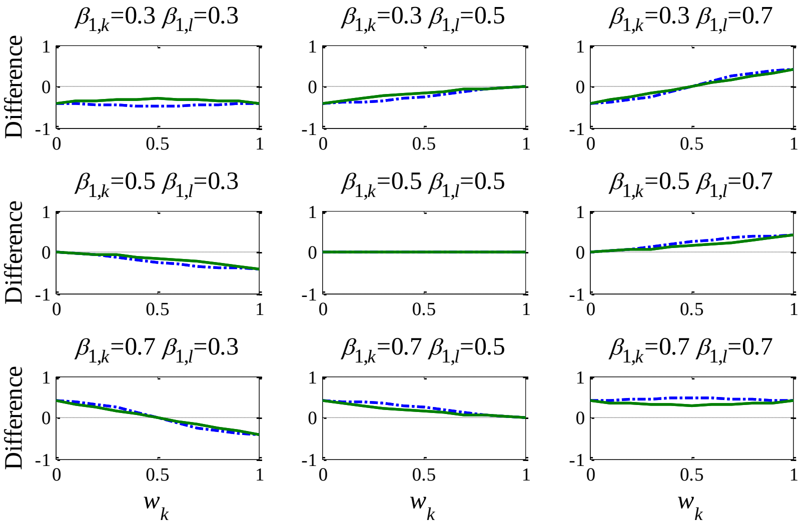


Fig. 7. Comparison of the ER-C and ER algorithms (The full line denotes $d^{ER\text{-}C}$ and the dotted line denotes d^{ER})

other words, when *Incons*(R_k) is equal to 1, R_k has a minimum rule weight. This makes sense that if this rule causes the contradiction, then it will be useless.

Apparently, when the number of rules is increasing, the inconsistency degree of each rule is also increasing. It follows that when L is approaching to ∞, the inconsistency degree of each rule is approaching to 1. Without loss of generality, we consider the rule weight of R_k as follows:

$$\theta_k = \lim_{L\to+\infty}\left[1-\frac{Incons(R_k)}{\sum_{l=1}^{L} Incons(R_l)}\right] = 1-\lim_{L\to+\infty}\frac{1}{L}=1 \qquad (4)$$

Now that L is equal to the number of sample data, this concludes the proof.

Example 1. Suppose there are L EBRs and two different EBRs, namely the kth rule and the ith rule (k, $i\in\{1,\ldots, L\}$; $i\neq k$) with their inconsistency degree *Incons*(R_k) and *Incons*(R_i), respectively. The relationship between rule weights and inconsistency degrees can be shown in Fig. 5: while *Incons*(i) is fixed, θ_k increases with the decrease of *Incons*(k). Moreover, while *Incons*(k)=1, the kth rule has a minimum rule weight.

Secondly, assume that *Incons*(R_k)=1 ($k\in\{1,\ldots, L\}$) and the inconsistency degree of other rules are set by using random values. Then the relationship between θ_k and L (from 1 to 1000) is shown in Fig. 6, in which each line (twenty lines in total) denotes an independent experiment of using random values to determine θ_i (i=1,…, L; $i\neq k$). From Fig. 6, *Incons*(k)=1 leads to the minimum rule weight for the kth rule. However, with the increase of L, it turned out that θ_k approximates to 1.

Remark 2: From Theorem 1, it is unnecessary to calculate each rule weight because its value will approximate to 1 while a large number of data are used to generate EBRs. As a result, the time complexity of the rule generation scheme is reduced from $O(T^2\times(\Sigma_{i=1,\ldots,M}J_i+N)$ to $O(T\times(\Sigma_{i=1,\ldots,M}J_i+N)$.

Theorem 2. For classification problems, the following ER algorithm for classification (denoted as ER-C) is the same as the analytical ER algorithm shown in Eq. (A1) in Appendix A.

$$\beta_n^{ER-C} = \prod_{k=1}^{L}\left(w_k\beta_{n,k}+1-w_k\sum_{i=1}^{N}\beta_{i,k}\right) \qquad (5)$$

Proof. Suppose a classification problem has N classes and the nth (n=1,…, N) class is denoted as D_n. Assume the estimated class of the EBRBS is the nth class for the test input data $\boldsymbol{x}$. Hence, according to Eq. (A3) in Appendix A, we have

$$\beta_n > \beta_t\ ; t=1,\ldots,N; t\neq n \qquad (6)$$

Based on the analytical ER algorithm in Eq. (A1), we assume

$$\chi^1 = \prod_{k=1}^{L}\left(1-w_k\sum_{i=1}^{N}\beta_{i,k}\right) \qquad (7)$$

$$\chi^2 = \prod_{k=1}^{L}(1-w_k) \qquad (8)$$

Hence, we can obtain

$$\beta_n = \frac{\beta_n^{ER-C}-\chi^1}{\sum_{i=1}^{N}\beta_i^{ER-C}-(N-1)\chi^1-\chi^2} > \frac{\beta_t^{ER-C}-\chi^1}{\sum_{i=1}^{N}\beta_i^{ER-C}-(N-1)\chi^1-\chi^2} = \beta_t \qquad (9)$$
$$\Leftrightarrow\ \beta_n^{ER-C} > \beta_t^{ER-C}; t=1,\ldots,N; t\neq n$$

It follows that the ER-C algorithm is the same as the analytical ER algorithm in EBRBS for classification.

Example 2. Suppose there are two EBRs and their belief distributions of the consequent attribute are shown as follows:

$$R_k : \{(D_n,\beta_{n,k}); n=1,2\}\ with \sum_{n=1}^{2}\beta_{n,k}=1 \qquad (10)$$

$$R_l : \{(D_n,\beta_{n,l}); n=1,2\}\ with \sum_{n=1}^{2}\beta_{n,l}=1 \qquad (11)$$

The activation weight of the rule R_k and R_l are assumed to be

w_k and w_l, respectively, and w_k+w_l=1.

According to the analytical ER algorithm in Eq. (A1) and the ER-C algorithm in Eq. (5), the difference of the integrated belief degrees for the D_1 and D_2 can be expressed as follows:

$$d^{ER} = \beta_1 - \beta_2 \tag{12}$$

$$d^{ER-C} = \beta_1^{ER-C} - \beta_2^{ER-C} \tag{13}$$

To illustrate the possible values of d^{ER} and $d^{ER\text{-}C}$, we consider nine illustrative cases under the assumptions that $\beta_{1,l}$>$\beta_{1,k}$, $\beta_{1,l}$= $\beta_{1,k}$, and $\beta_{1,l}$<$\beta_{1,k}$ while the value of w_k lies between 0 and 1. Without loss of generality, the value of both $\beta_{1,l}$and $\beta_{1,k}$ is assumed as 0.3, 0.5, and 0.7, respectively. Hence, the curves of d^{ER} and $d^{ER\text{-}C}$ are shown in Fig. 7. Fig. 7 shows that d^{ER} and $d^{ER\text{-}C}$ have same negative and positive symbols for nine combinations based on different belief degrees and activation weights. Hence, the ER-C algorithm can produce the same estimated class as the ER algorithm.

Remark 3. From Theorem 2, the ER-C algorithm can be used to replace the ER algorithm in the inference scheme of the EBRBS while facing classification problems. Additionally, the existing studies of using the ER algorithm as inference engine for classification problems, such as [6] and [7], can also use the ER-C algorithm to replace the inference engine.

Remark 4. The advantages of the ER-C algorithm over the ER algorithm can be summarized as follows:

(1) The ER-C algorithm has a much clean and simple formula than the ER algorithm because it derives from the core part of the ER algorithm.

(2) The ER-C algorithm is more efficient than the ER algorithm according to their time complexity, in which the ER-C algorithm is $O(L\times N)$ and the ER algorithm is $O(L\times N^2)$.

(3) In term of independence, the calculation of the integrated belief degree for all classes is independent of each other in the ER-C algorithm so that it is possible to have more solutions of parallelization for the EBRBS.

B. Domain division-based rule reduction method for EBRBS

In order to further reduce the time complexity of the EBRBS, in this subsection, a domain division-based rule reduction method is proposed for the rule generation scheme. Firstly, the main idea, which follows the similar way to the fuzzy partition [17] and the Wang-Mendel model [31], of the proposed rule reduction method are given based on the following definitions.

Definition 1 (Division point). The division point is the intersection between transform functions used to calculate the belief degree to which the input data belongs to the reference value of antecedent attributes. For convenience, $P(A_{i,j}, A_{i,j+1})$ (i=1,…, M; j=1,…, J_i -1) is defined to express the division point between the $A_{i,j}$ and $A_{i,j+1}$ in the ith antecedent attribute.

Definition 2 (Division domain). The division domain is the local input space constructed by the two adjacent division points of each antecedent attribute. For convenience, $D(A_{1,j_1},..., A_{M,j_M})$ (j_i=1,…, J_i; i=1,…, M) is defined to express the division domain constructed by the division point regarding the reference values $A_{1,j_1},..., A_{M,j_M}$.

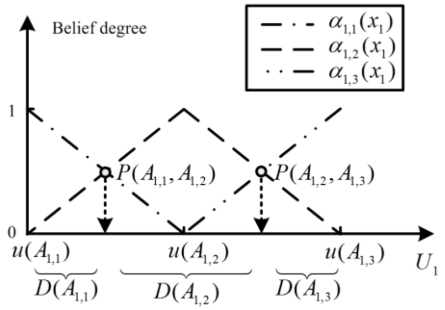


Fig. 8. Example of division point and division domain

Example 3. Suppose that an EBRB includes one antecedent attribute U_1 with three reference values $A_{1,i}$ and their utility values $u(A_{1,i})$ (i=1, 2, 3). Without loss of generality, the order of those utility values is $u(A_{1,1})$<$u(A_{1,2})$<$u(A_{1,3})$. x_1 is assumed to be the input variable of the attribute U_1 and the utility function used to generate belief degrees of the reference value $A_{1,i}$ (i=1, 2, 3), is assumed to be piecewise linear [35], namely $\alpha_{1,i}(x_1)$, shown in Fig. 8. From Fig. 8, there are two division points, namely $P(A_{1,1}, A_{1,2})$ and $P(A_{1,2}, A_{1,3})$, and three division domains, namely $D(A_{1,1})$, $D(A_{1,2})$, and $D(A_{1,3})$.

Remark 5. Based on Definitions 1 and 2 together with Example 3, when an input data falls into a division domain, the relatively maximal belief degree can be generated from the input data for all reference values belonging to the division domain. In other words, each division domain can be regarded as the clustering center of the input data which most likely belong to the reference value of division domain in the form of belief distribution.

Definition 3 (Rule clustering strategy). The rule clustering strategy is the map relationship between EBRs and division domains based on the relatively maximal belief degree in each antecedent attribute so that it can be defined as

$$R_k \rightarrow D(A_{1,j_1},..., A_{M,j_M}) \tag{14}$$

where $j_i = \arg\max_{j=1,...,J_i} \{\alpha_{i,j}^k\}$ (k=1,…, L; i=1,…, M); $\alpha_{i,j}^k$ denotes the belief degree of the reference value $A_{i,j}$ in the kth EBR (j=1,…, J_i;i=1,…, M), M is the number of antecedent attributes, L is the number of EBRs, and J_i is the number of reference values used for the ith antecedent attribute.

Definition 4 (Rule reduction strategy). The rule reduction strategy is the combination strategy for the EBRs which are assigned to the same division domain so that it can be defined as

$$\bar{\alpha}_{i,j}^l = \frac{\sum_{k=1}^{L_l} \alpha_{i,j}^k}{L_l}, \bar{\beta}_{n,l} = \frac{\sum_{k=1}^{L_l} \beta_{n,k}}{L}; while\, L_l > 0 \tag{15}$$

where L_l is the number of the rules gathered at the lth (l=1,…, $\prod_{i=1}^{M} J_i$) division domain and $\beta_{n,k}$ (n=1,…, N; k=1,…, L) denote the belief degrees of the reference value $A_{i,j}$ and D_n in the kth rule, respectively.

Remark 6. As shown in Eq. (15), different denominators are

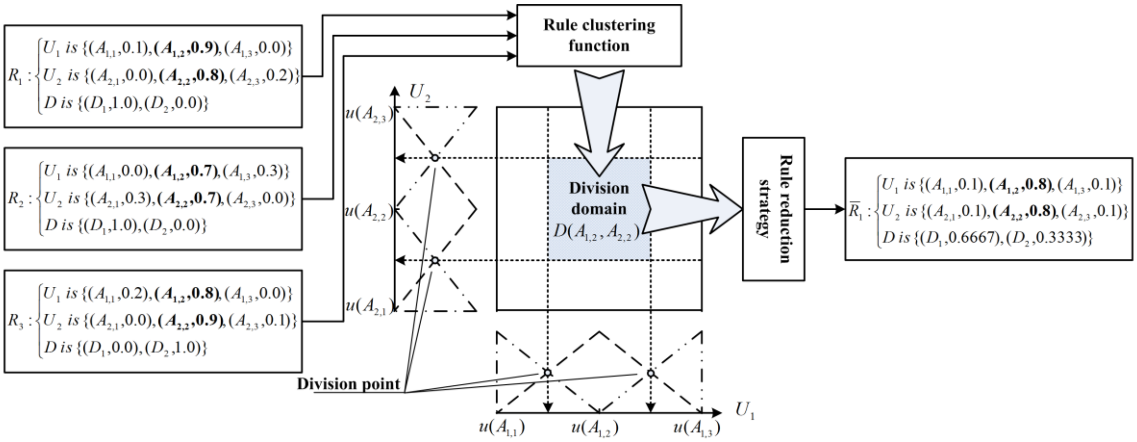

$R_1: \{U_1 \text{ is } \{(A_{1,1},0.1),(\mathbf{A_{1,2}},\mathbf{0.9}),(A_{1,3},0.0)\}; U_2 \text{ is } \{(A_{2,1},0.0),(\mathbf{A_{2,2}},\mathbf{0.8}),(A_{2,3},0.2)\}; D \text{ is } \{(D_1,1.0),(D_2,0.0)\}$
$R_2: \{U_1 \text{ is } \{(A_{1,1},0.0),(\mathbf{A_{1,2}},\mathbf{0.7}),(A_{1,3},0.3)\}; U_2 \text{ is } \{(A_{2,1},0.3),(\mathbf{A_{2,2}},\mathbf{0.7}),(A_{2,3},0.0)\}; D \text{ is } \{(D_1,1.0),(D_2,0.0)\}$
$R_3: \{U_1 \text{ is } \{(A_{1,1},0.2),(\mathbf{A_{1,2}},\mathbf{0.8}),(A_{1,3},0.0)\}; U_2 \text{ is } \{(A_{2,1},0.0),(\mathbf{A_{2,2}},\mathbf{0.9}),(A_{2,3},0.1)\}; D \text{ is } \{(D_1,0.0),(D_2,1.0)\}$
Rule clustering function
U_2
$u(A_{2,3})$
$u(A_{2,2})$
$u(A_{2,1})$
Division domain
$D(A_{1,2}, A_{2,2})$
Rule reduction strategy
$\overline{R}_1: \{U_1 \text{ is } \{(A_{1,1},0.1),(\mathbf{A_{1,2}},\mathbf{0.8}),(A_{1,3},0.1)\}; U_2 \text{ is } \{(A_{2,1},0.1),(\mathbf{A_{2,2}},\mathbf{0.8}),(A_{2,3},0.1)\}; D \text{ is } \{(D_1,0.6667),(D_2,0.3333)\}$
Division point
U_1
$u(A_{1,1})$
$u(A_{1,2})$
$u(A_{1,3})$

Fig. 9. Examples of rule clustering strategy and rule reduction strategy

used to calculate the belief degree of antecedent and consequent attributes. The main reason is that: (1) the belief degree of antecedent attribute reflects the space relationship between different division domains in a sense, so the belief degree is calculated by using L_l; (2) the belief degree of consequent attribute reflects how many rules have the same class, thus the belief degree is calculated by using L.

Example 4. Suppose that an EBRB includes two antecedent attributes U_1 and U_2 with three reference values $A_{i,j}$ and their utility values $u(A_{i,j})$ (i=1, 2; j=1, 2, 3). Without loss of generality, the order of those utility values is $u(A_{i,1})<u(A_{i,2})<u(A_{i,3})$ in the ith antecedent attribute. In addition, the EBRB has three EBRs R_1, R_2, and R_3, respectively, as shown in Fig. 9.

From Fig. 9, four division points and nine division domains can be constructed based on Definitions 1 and 2. After that, according to Definition 3, all three EBRs are assigned to the division domain $D(A_{1,2}, A_{2,2})$. Afterwards, based on Definition 4, the EBRs gathered in $D(A_{1,2}, A_{2,2})$ are all combined as a new rule, denoted as $\overline{R}_1$, which is obtained via Eq. (15), e.g., $\overline{\alpha}_{1,1}^{1}$ =(0.1+0+0.2)/3=0.1, $\overline{\alpha}_{1,2}^{1}$ =(0.9+0.7+0.8)/3 =0.8, and $\overline{\alpha}_{1,3}^{1}$ =(0 +0.3+0)/3=0.1 in the antecedent attribute U_1, and $\overline{\beta}_{1,1}$ = (1+1+ 0)/3=0.6667 and $\overline{\beta}_{2,1}$ =(0+0+1)/3=0.3333 in the attribute D.

Remark 7. Based on Definitions 3 and 4 together with Example 4, when one more EBR is assigned to the same division domain, a new EBR would be generated, in which the information of the new EBR is considered complete because it is generated by using the belief distribution of the antecedent and consequent attribute of all original rules.

Based on the above Definitions 1 to 4 and Examples 3 to 4, the steps of the domain division-based rule reduction method are described as follows:

Step 1: To generate division points for each antecedent attribute by using the transform functions. Suppose that there are M antecedent attributes with J_i reference values for the ith (i=1,…, M) antecedent attribute. Based on Definition 1, J_i -1 division points, namely {$P(A_{i,j}, A_{i,j+1})$; j=1,…, J_i-1}, are generated for the ith antecedent attribute.

Step 2: To generate division domains for the EBRBS by using the division points. Based on Definition 2, $\prod_{i=1}^{M} J_i$ division domains, namely { $D(A_{1,j_1},..., A_{M,j_M})$; j_i=1,…, J_i; i= 1,…, M}, are generated for the EBRBS, in which the division domain is the clustering center according to Remark 5.

Step 3: To assign all EBRs to the division domains based on the rule clustering strategy. Suppose that there are T EBRs transformed from T sample data. Based on Definition 3, these T EBRs are all assigned to the $\prod_{i=1}^{M} J_i$ division domains.

Step 4: To generate new EBRs from the EBRs gathered in each division domain based on the rule reduction strategy. Suppose that there are $\overline{L}$ division domains which include at least one EBR, based on Definition 4, all EBRs in each division domain are used to generate $\overline{L}$ new EBR, respectively.

Remark 8. After utilizing the domain division-based rule reduction method to downsize the EBRB, $\overline{L}$ new EBRs shown in Step 4 can construct a reduced EBRB, in which the number of rules regarding the reduced EBRB should be no more than both the number of sample data T and the number of division domains $\prod_{i=1}^{M} J_i$.

C. Micro-EBRBS: EBRBS with rule reduction and ER-C algorithm but without rule weight calculation

Based on the above analysis and new improvements, a novel EBRBS, called *Micro-EBRBS* which has a simplified rule generation and inference schemes and a downsized EBRB comparing to EBRBS, is developed and its methodological framework is shown in Fig. 10.

It is clear from Fig. 10 that, comparing to the EBRBS shown in Figs. 1 and 2, the Micro-EBRBS includes the process of rule reduction but exclude the process of rule weight calculation. Additionally, the ER-C algorithm is used to replace the ER algorithm in the process of activated rule integration. More specifically, the rule generation and the inference scheme of the Micro-EBRBS are described as follows.

For the rule generation scheme of the Micro-EBRBS, it consists of the following two steps:

Step 1: Generation of belief distributions using the transformation technique.

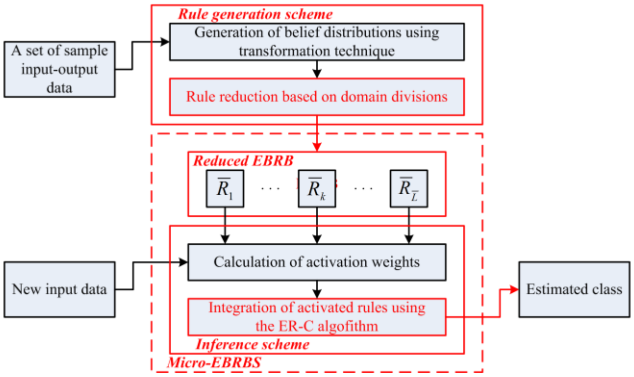

Rule generation scheme
A set of sample input-output data
Generation of belief distributions using transformation technique
Rule reduction based on domain divisions
Reduced EBRB
$\overline{R}_1$
$\overline{R}_k$
$\overline{R}_{\overline{L}}$
New input data
Calculation of activation weights
Integration of activated rules using the ER-C algofithm
Inference scheme
Micro-EBRBS
Estimated class

Fig. 10. Illustration of Micro-EBRBS

After determining the basic parameters based on expert knowledge, including attribute weights, utility values of the reference values used for antecedent and consequent attributes, the belief distributions of the antecedent and consequent attributes can be generated.

Suppose that $\{u(A_{i,j});\ j=1,\dots,J_i\}$ is a set of given utility values used for the ith ($i=1,\dots,M$) antecedent attribute, and $x_{k,i}$ is the kth ($k=1,\dots,L$) sample input data of the ith antecedent attribute. Thus, the belief distribution of the ith antecedent attribute generated using the utility-based equivalence transformation technique [35] is:

$$S(x_{k,i})=\{(A_{i,j},\alpha_{i,j}^{k});\ j=1,\dots,J_i\} \tag{16}$$

where

$$\alpha_{i,j}^{k}=\frac{u(A_{i,j+1})-x_{k,i}}{u(A_{i,j+1})-u(A_{i,j})} \tag{17}$$

$$\alpha_{i,j+1}^{k}=1-\alpha_{i,j}^{k},\ \text{if}\ u(A_{i,j})\le x_{k,i}\le u(A_{i,j+1})$$
$$\alpha_{i,t}^{k}=0,\ \mathit{for}\ t=1,\dots,J_i\ \mathit{and}\ t\ne j,j+1 \tag{18}$$

where $\alpha_{i,j}^{k}$ is the belief degree of $A_{i,j}$ in the kth EBR obtained from the sample input data $x_{k,i}$.

Next, when the kth sample output data is y_k and the given utility values attached to the consequent attribute D are $\{u(D_n);\ n=1,\dots,N\}$, the belief distribution of consequent attribute is:

$$S(y_k)=\{(D_n,\beta_n^{k});n=1,\dots,N\} \tag{19}$$

Finally, all belief distributions shown in Eqs (16) and (19) together with attribute weights determined by experts are used to construct an EBRB.

Step 2: Rule reduction based on domain divisions.

For the EBRB generated by Step 1, the domain division-based rule reduction method shown in Section III-B is then used to reduce the number of rules. Suppose there are M antecedent attributes with J_i reference values $A_{i,j}$ ($j=1,\dots,J_i$) and one consequent attribute D with N classes D_n ($n=1,\dots,N$), the kth ($k=1,\dots,\overline{L}$) rule of the reduced EBRB can be written as

$$\begin{aligned}&\overline{R}_k: \mathit{IF}\ U_1\ \mathit{is}\ \{(A_{1,j},\overline{\alpha}_{1,j}^{k});\ j=1,\dots,J_1\}\wedge\cdots\wedge U_M\ \mathit{is}\\ &\quad\{(A_{M,j},\overline{\alpha}_{M,j}^{k});\ j=1,\dots,J_M\},\ \mathit{THEN}\ D\ \mathit{is}\\ &\quad\{(D_n,\overline{\beta}_{n,k});n=1,\dots,N\};\mathit{with}\ \theta_k=1\ \mathit{and}\ \{\delta_1,\dots,\delta_M\}\end{aligned} \tag{20}$$

where $\overline{\alpha}_{i,j}^{k}$ and $\overline{\beta}_{n,k}$ are the integrated belief degree of the $A_{i,j}$ and the class D_n using the rule reduction strategy.

Remark 9. As shown in the proposed rule reduction method, the most complex step is to generate $\overline{\alpha}_{i,j}^{k}$ ($k=1,\dots,\overline{L}$; $i=1,\dots,M$; $j=1,\dots,J_i$) and $\overline{\beta}_{n,k}$ ($n=1,\dots,N$) by using L rules of EBRB. Furthermore, Section II-C shows that the time complexity of generating belief distributions is $O(L\times(\Sigma_{i=1,\dots,M}J_i+N))$. Hence, the time complexity for the rule generation scheme of the Micro-EBRBS is $O(L\times\overline{L}\times(\Sigma_{i=1,\dots,M}J_i+N))$.

For the inference scheme of the Micro-EBRBS, it consists of the following two steps:

Step 1: Calculation of activation weights using the distance measure.

While a test input data is provided for the Micro-EBRBS, the activation weights can be calculated for each EBR of the reduced EBRB. Suppose that $\boldsymbol{x}=(x_1,\dots,x_M)$ is a test input data, each input x_i ($i=1,\dots,M$) will be firstly transformed into a belief distribution of the reference values of the ith antecedent attribute using Eqs. (17) and (18).

$$S(x_i)=\{(A_{i,j},\alpha_{i,j});\ j=1,\dots,J_i\} \tag{21}$$

Next, the individual matching degree of the ith antecedent attribute in the kth rule, denoted as $S^k(x_i, U_i)$, is calculated by using the Euclidean distance:

$$\begin{aligned}S^k(x_i,U_i)&=1-d^k(x_i,U_i)\\&=1-\min\left\{1,\sqrt{\sum_{j=1}^{Ji}(\alpha_{i,j}-\alpha_{i,j}^{k})^2}\right\}\end{aligned} \tag{22}$$

where $d^k(x_i, U_i)$ is the distance measurement.

Finally, the activation weight of the kth EBR, denoted as w_k, is calculated by

$$w_k=\frac{\theta_k\prod_{i=1}^{M}\left(S^k(x_i,U_i)\right)^{\overline{\delta}_i}}{\sum_{l=1}^{L}\left(\theta_l\prod_{i=1}^{M}\left(S^l(x_i,U_i)\right)^{\overline{\delta}_i}\right)},\ \overline{\delta}_i=\frac{\delta_i}{\max_{i=1,\dots,M}\{\delta_i\}} \tag{23}$$

where θ_k is the weight of the kth rule; δ_i is the weight of the ith antecedent attribute.

Step 2: Integration of activated rules using the ER-C algorithm.

After performing Step 1, all activated rules can be integrated using the ER-C algorithm shown in Eq. (5) and the integrated belief distribution of the test input data $\boldsymbol{x}$ can be represented as follows:

$$f(\boldsymbol{x})=\{(D_n,\beta_n^{ER-C});n=1,\dots,N\} \tag{24}$$

Afterwards, the estimated class for the test input data $\boldsymbol{x}$ can be obtained as follows:

$$f(\boldsymbol{x})=D_n, n=\arg\max_{i=1,\dots,N}\{\beta_i^{ER-C}\} \tag{25}$$

Remark 10. Considering that the inference scheme of the Micro-EBRBS is based on the reduced EBRB, which only has $\overline{L}$ rules, the time complexity of the inference scheme

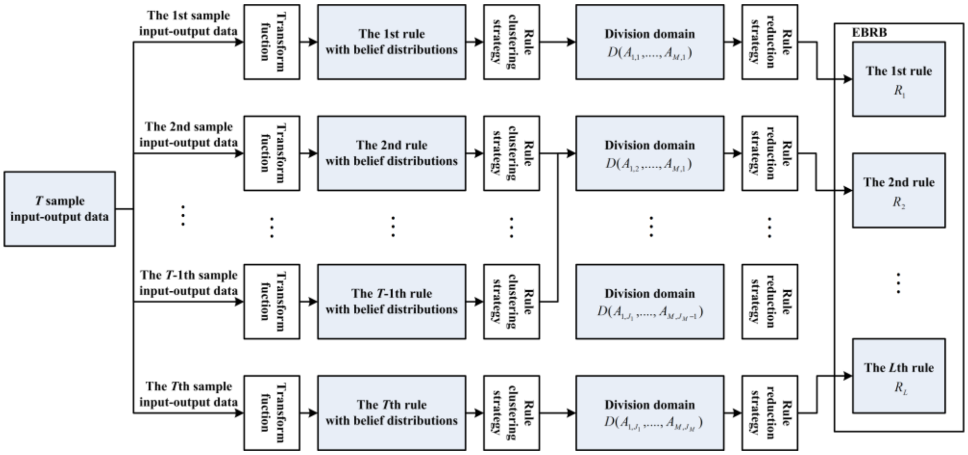

T sample input-output data
The 1st sample input-output data
The 2nd sample input-output data
The T-1th sample input-output data
The Tth sample input-output data
Transform fuction
The 1st rule with belief distributions
The 2nd rule with belief distributions
The T-1th rule with belief distributions
The Tth rule with belief distributions
Rule clustering strategy
Division domain $D(A_{1,1},\ldots,A_{M,1})$
Division domain $D(A_{1,2},\ldots,A_{M,1})$
Division domain $D(A_{1,J_1},\ldots,A_{M,J_M-1})$
Division domain $D(A_{1,J_1},\ldots,A_{M,J_M})$
Rule reduction strategy
EBRB
The 1st rule R_1
The 2nd rule R_2
The Lth rule R_L

Fig. 11. Illustration of the parallel rule generation scheme

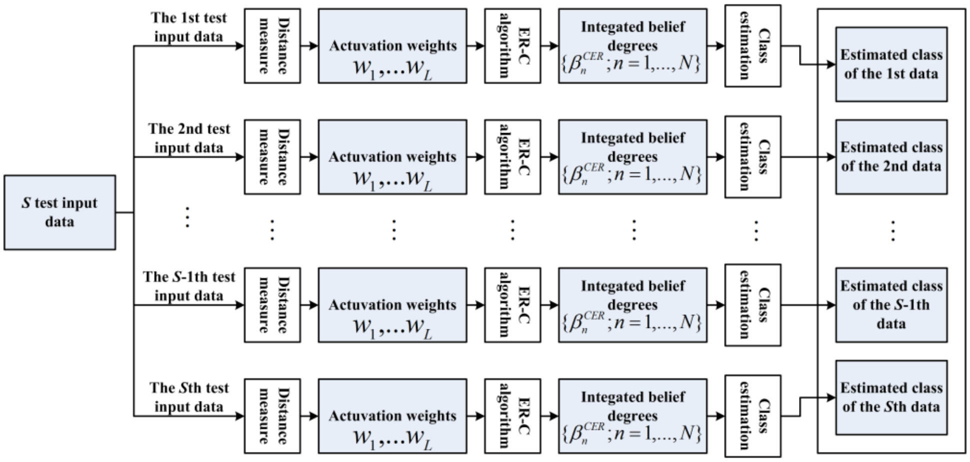

S test input data
The 1st test input data
The 2nd test input data
The S-1th test input data
The Sth test input data
Distance measure
Actuvation weights $w_1,\ldots w_L$
ER-C algorithm
Integated belief degrees $\{\beta_n^{CER};n=1,\ldots,N\}$
Class estimation
Estimated class of the 1st data
Estimated class of the 2nd data
Estimated class of the S-1th data
Estimated class of the Sth data

Fig. 12. Illustration of the parallel inference scheme

regarding the Micro-EBRBS is $O(\overline{L}\times(\Sigma_{i=1,\ldots,M}J_i+N))$ for classifying each test data.

D. Apache Spark-based implementation of Micro-EBRBS for big data multi-class classification

In order to further improve the computing efficiency of the Micro-EBRBS, the Apache Spark is used to achieve the parallel rule generation and inference schemes. As an in-memory big data platform, the Apache Spark has been proven that it supports a much wider range of functionality than the Apache Hadoop [9]. The fundamental data structure of the Apache Spark is the resilient distributed dataset (RDD), which represents a collection of distributed items that can be manipulated across many computing nodes concurrently. Hence, the RDD allows the data cache to be stored in memory and perform computations for the same data directly from memory. After the RDD is constructed, the program can perform the following two operations:

(1) *Transformations*: this kind of operation is to create a new RDD from existing RDD and the concrete function includes *map* (which is to return a new RDD formed by passing each element of the source through a function), *mapToPair* (which is to return a new RDD of key-value pairs by using a function), *reduceByKey* (which is to return a new RDD of key-value pairs where the values for each key are aggregated using the given reduce function), and so on. The detailed description of those functions can be found in [4].

(2) *Actions*: this kind of operation is to return the final results of RDD computations and the concrete function includes *reduce* (which is to aggregate the elements of the RDD using a function), *collect* (which is to return all the elements of the RDD as an array at the driver program), and others. The detailed description of those functions can be found in [4].

Based on the functions of transformation and action, the pseudocodes are provided to illustrate the Apache Spark-based implementation of the parallel rule generation and inference schemes of Micro-EBRBS below respectively.

IV. Experiments

The performance of the Micro-EBRBS is empirically assessed through three different experiments with 14 classification datasets from the well-known UCI repository of machine learning databases [30]. The EBRBS, the conventional FRBCS and machine-learning classifiers, and the big data FRBCS classifiers are used to compare in terms of the accuracy and computing efficiency, respectively.

A. Datasets and experiment conditions

Fourteen classification datasets obtained from UCI are used to evaluate the performance of the Micro-EBRBS. The main

Pseudocode of parallel rule generation scheme of Micro-EBRBS

Input: ***SampleDataSet*** denotes the set of sample input-output data, each sample input-output data of ***SampleDataSet*** is denoted as *sampleData*, *rule* denotes the EBR shown in Eq. (20), *tuple1*, *tuple2*, and *tuple3* denote the 2-tuple composing of *rule* and its division domain.
Output: A set of extended belief rules ***EBRBSet***

```
EBRBSet = new JavaSparkContext().Parallelize (SampleDataSet).mapToPair(sampleData -> {
    Generate rule from sampleData by using Steps 1 to 2 shown in the rule generation scheme of the Micro- EBRBS;
    Generate divisionDomain for rule by using Steps 1 to 3 shown in the domain division-based rule reduction method;
Return new Tuple2<>(divisionDomain, rule)
}).reduceByKey((tuple1, tuple2) -> {
    Generate tuple3 by using Step 4 shown in the domain division-based rule reduction method;
    Return tuple3;
}).map(tuple3 -> {
    Obtain rule from tuple3;
    Return rule;
}).collect();
```

Pseudocode of parallel inference scheme of Micro-EBRBS

Input: ***TestDataSet*** denotes the set of test input data, each test input data of ***TestDataSet*** is denoted as *testData*, *class1* denotes the estimated class of the Micro-EBRBS, and *class2* denotes the actual class of test input data, *a* and *b* denote the integer variable.
Output: The total number of test input data correctly classified by the Micro-EBRBS *totalCorrect*.

```
totalCorrect = new JavaSparkContext().parallelize(TestDataSet)
.map(testData -> {
Generate class1 for testData by using Steps 1 to 2 shown in the inference scheme of the Micro-EBRBS;
Obtain class2 from testData;
Return class1==class2?1:0;
}) .reduce((a, b) -> a+b);
```

TABLE I
STATISTICS ON FOURTEEN CLASSIFICATION DATASETS

No.	Dataset	No. of data	No. of attributes	No. of classes
1	Diabetes	393	8	2
2	Cancer	569	30	2
3	Transfusion	748	4	2
4	Banknote	1,372	4	2
5	Magic	19,020	10	2
6	Wine	178	13	3
7	Waveform	5,000	21	3
8	Glass	214	9	6
9	Red wine	1,599	11	6
10	Satimage	6,435	36	6
11	Census	95,130	40	2
12	Gas sensors	928,991	10	3
13	Covtype	581,012	54	7
14	Poker	1,025,010	10	10

characteristics of these datasets are summarized in Table I. Notice that for the datasets Diabetes, Cancer, and Census, we have removed the data with missing attribute values.

To develop the comparison in multiple aspects, *k*-fold cross-validation (K-CV) is considered in the experiments, where each dataset is divided into *k* blocks, with *k*-1 blocks as training data, namely sample input-output data, and the remaining block as testing data. Additionally, the nonparametric statistical analysis is used to assess if significant differences exist among different classifiers at a level of significance of α=0.1. For conducting multiple statistical comparisons over multiple datasets, as suggested in [14], the Friedman and Holm tests are employed.

For the first and the second experiments (Sections IV-B and IV-C), the datasets with relatively small number of data, including the 1st to the 10th datasets, are used to compare the performance of the Micro-EBRBS with the EBRBS and the conventional FRBCS and machine-learning classifiers. All these classifiers are implemented using Java programming (JDK 1.8.0) and the open source software (Weka and KEEL) on Intel (R) Core (TM) i5-4300U CPU at 1.90GHz and 4GB RAM with Windows 7. For each dataset, the average results of the 10 runs of each classifier are used to compare their performances.

For the third experiment (Section IV-D), the datasets with relatively large number of data, including the 11th to the 14th datasets, are used to compare the performance of the Micro-EBRBS with the big data FRBCS classifiers, and all these classifiers are executed in the 17 nodes cluster connected via 8GT/s Ethernet LAN network, where the master node is composed of 1 Intel Xeon E5-2640 4 cores at 2.5GHz and 16GB RAM and the slave nodes are composed of 2 Intel Xeon E5-2670v2 10 cores at 2.5GHz and 64GB RAM. The entire cluster runs under Red Hat 7.3 and Apache Spark 2.1.0.

B. Comparative analysis between Micro-EBRBS and EBRBS

The first experiment aims to compare the accuracy and computing efficiency of the Micro-EBRBS with the EBRBS, respectively, and the comparisons are based on the 2/4/6/8/10-CV to investigate the influences on the accuracy and computing efficiency of these EBRBSs by using different numbers of training and testing data.

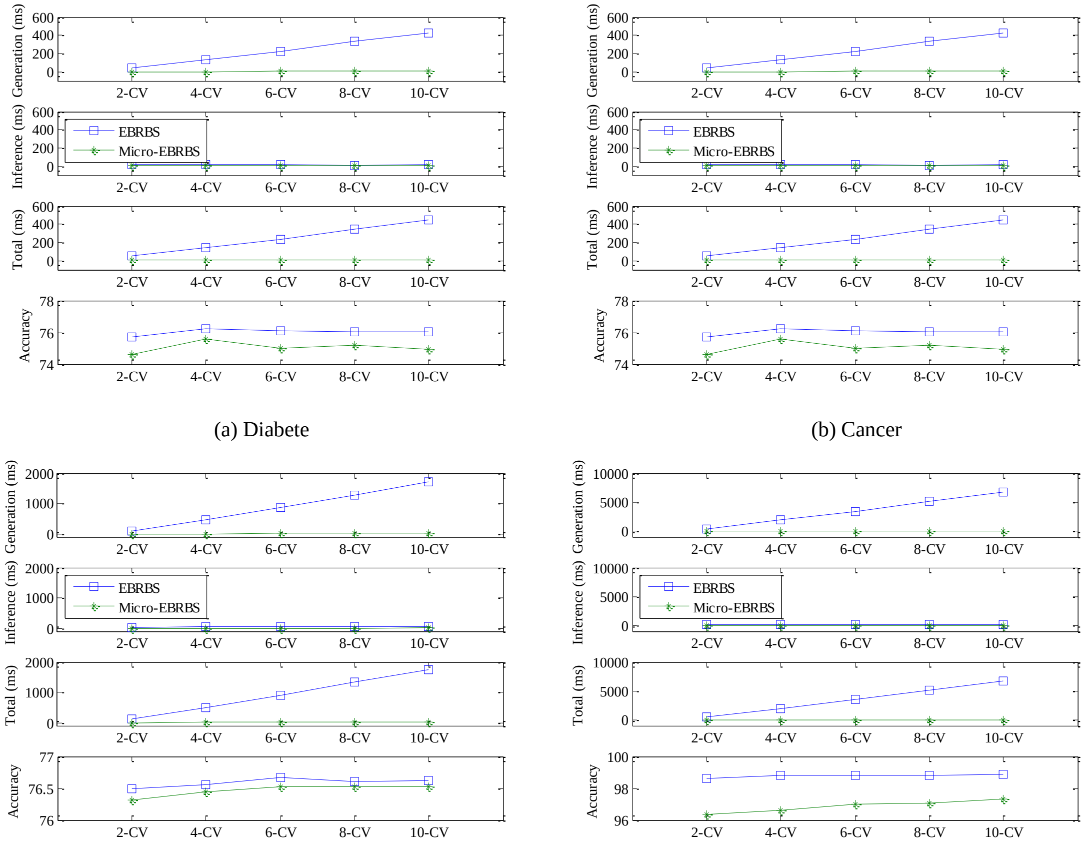
For the basic parameters of the Micro-EBRBS and EBRBS, suppose that all attribute weights are 1, namely

$$\delta_i = 1; i = 1,..., M \tag{26}$$

where *M* is the number of antecedent attributes. The number of reference values is three for each antecedent attribute, and the utility value of these reference values is defined as follows:

$$\{u(A_{i,j}); j = 1,2,3\} = \left\{ lb_i, \frac{lb_i + ub_i}{2}, ub_i \right\}; i = 1,..., M \tag{27}$$

where $u(A_{i,j})$ denotes the utility value of the reference value $A_{i,j}$, lb_i and ub_i denote the lower and upper bounds of the *i*th antecedent attribute, respectively.



(a) Diabete

(b) Cancer

(c) Transfusion

(d) Banknote

Fig. 13. Comparison of computing time and accuracy between Micro-EBRBS and EBRBS for two-class datasets

TABLE II
COMPARISON OF COMPUTING EFFICIENCY FOR MICRO-EBRBS AND EBRBS

No. of classes	Dataset	Rule generation (ms)		Ratio	Inference (ms)		Ratio	Total (ms)		Ratio
		Micro-EBRBS	EBRBS		Micro-EBRBS	EBRBS		Micro-EBRBS	EBRBS	
Two-class	Diabetes	**6.1**	429.0	0.014	**3.2**	15.6	0.205	**9.3**	444.6	0.021
	Cancer	**40.8**	1,475.5	0.028	**69.9**	76.7	0.911	**110.7**	1,552.2	0.071
	Transfusion	**3.2**	1,709.6	0.002	**3.0**	37.5	0.080	**6.2**	1,747.1	0.004
	Banknote	**12.4**	6,645.2	0.002	**3.2**	148.3	0.022	**15.6**	6,793.5	0.002
Multi-class	Wine	**4.6**	81.1	0.058	**3.2**	4.7	0.681	**7.8**	85.8	0.091
	Waveform	**750.5**	205,530.3	0.004	**3,623.3**	8,654.2	0.419	**4,373.8**	214,184.5	0.020
	Glass	**3.0**	164.4	0.018	**1.1**	5.3	0.208	**4.1**	169.7	0.024
	Red Wine	**40.7**	13,291.8	0.003	**60.2**	424.9	0.142	**100.9**	13,716.7	0.007

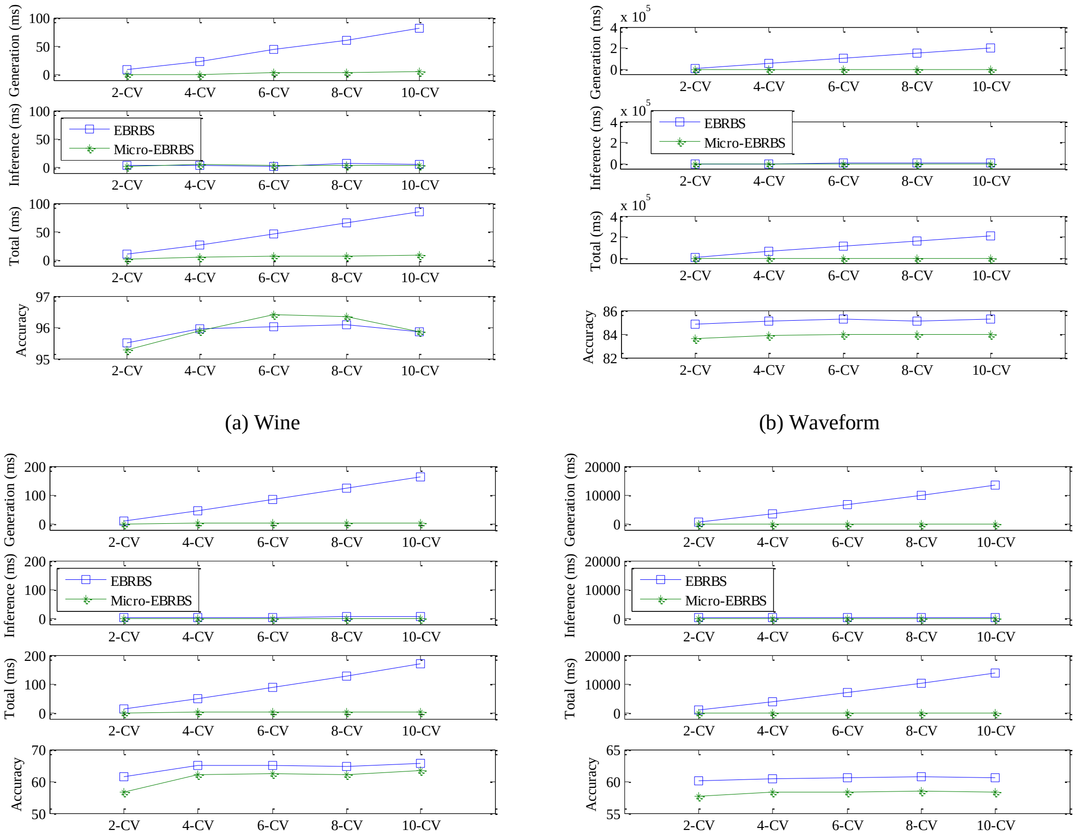
Figs. 13 and 14 show the computing time and the accuracy regarding the Micro-EBRBS in comparison with the EBRBS over two and multi-class datasets, respectively, in which the computing time includes the time of rule generation scheme, inference scheme and total time.

For the two-class datasets, including Diabetes, Cancer, Transfusion, and Banknote, the computing time of rule generation scheme regarding the EBRBS is increasing with the increasing number of training data used to generate EBRs, e.g., 50% data are regarded as the training data in 2-CV and 90% data as the training data in 10-CV. Additionally, there are slight differences between the Micro-EBRBS and the EBRBS in term of the computing time of the inference scheme and the accuracy. As a result, the total computing time of the EBRBS is much more than that of the Micro-EBRBS.

For the multi-class datasets, including Wine, Waveform, Glass, Red wine, the similar conclusions are obtained in terms of the computing time and the accuracy.

In order to show the detailed comparison of the Micro-EBRBS and the EBRBS, Tables II and III provide the results of rule generation time, inference time, total time, number of rules, number of activated rules, and the accuracy, in which the ratio is the result of the Micro-EBRBS divided by the result of the EBRBS. Hence, the Micro-EBRBS with a larger ratio in term of accuracy and a smaller ratio in term of computing time is better than the EBRBS.

Table II shows that the computing time of the Micro-EBRBS is much less than that of the EBRBS for all two and multi-class datasets, where the minimum ratio of computing time is obtained from the dataset Banknote and its ratios are 0.002,



(a) Wine (b) Waveform

(c) Glass (d) Red wine

Fig. 14. Comparison of computing time and accuracy between Micro-EBRBS and EBRBS for multi-class datasets

TABLE III
COMPARISON OF ACCURACY AND NUMBER RULES AND ACTIVATED RULES FOR MICRO-EBRBS AND EBRBS

No. of classes	Dataset	No. of rules		Ratio	No. of activated rules		Ratio	Accuracy (%)		Ratio
		Micro-EBRBS	EBRBS		Micro-EBRBS	EBRBS		Micro-EBRBS	EBRBS	
Two-class	Diabetes	**94.0**	353.7	0.266	**49.6**	208.9	0.237	74.91	**76.01**	0.986
	Cancer	**448.2**	512.1	0.875	**222.7**	256.9	0.867	**96.49**	96.45	1.000
	Transfusion	**12.8**	673.2	0.019	**7.4**	533.1	0.014	76.52	**76.62**	0.999
	Banknote	**30.8**	1,234.8	0.025	**22.3**	955.1	0.023	97.34	**98.86**	0.985
Multi-class	Wine	**126.3**	160.2	0.788	**45.2**	58.6	0.771	**95.84**	**95.84**	1.000
	Waveform	**2,031.7**	4,500.0	0.452	**1,492.9**	3,523.1	0.424	84.00	**85.22**	0.986
	Glass	**41.9**	192.6	0.218	**11.8**	95.4	0.124	63.32	**65.56**	0.967
	Red Wine	**230.8**	1,439.1	0.160	**139.6**	964.1	0.145	58.36	**60.61**	0.963

TABLE IV
FRIEDMAN AND HOLM TESTS TO COMPARE THE ACCURACY OF MICRO-EBRBS AND EBRB (α =0.1)

Indicator	EBRBS (two-class)	EBRBS (Multi-class)
p value	0.3173	0.1336
Critical value	0.1000	0.1000
Hypothesis	Accepted	Accepted

0.022, and 0.002 for the rule generation, inference, and total time, respectively, and the maximum one is obtained from the datasets Cancer and Wine and their ratio are 0.058 and 0.091 for the rule generation time and total time of Wine and 0.911 for the inference time of Cancer.

Table III shows that the number of rules and activated rules involved in the Micro-EBRBS is less than the EBRBS, where the minimum ratio is obtained from the datasets Transfusion, i.e., 0.019 for the number of rules and 0.014 for the number of activated rules, and the maximum one is obtained from the datasets Cancer, 0.875 for the number of rules and 0.867 for the number of activated rules. Furthermore, it is clear from Table III that the accuracy of the Micro-EBRBS closely approximates that of the EBRBS and the range of the ratio can be expressed as [0.963, 1.000] for eight classification datasets.

In order to further compare the accuracy of the Micro-EBRBS and EBRBS, Friedman and Holm tests are applied to provide the statistical analysis based on two and multi-class datasets, respectively. From Table IV, although the EBRBS can obtain the best accuracy at most of two and multi- class datasets, none of hypotheses is in favor of the significant difference between EBRBS and Micro-EBRBS. However, the computing efficiency of Micro-EBRBS is much better than EBRBS.

TABLE V
COMPARISON OF ACCURACY FOR MICRO-EBRBS AND FIVE FRBCS CLASSIFIERS

No. of classes	Dataset	SLAVE	FH-GBML	FURIA	FARC-HD	Chi-FRBCS	Micro-EBRBS
Two-class	Diabetes	**77.10% (1.5)**	70.23% (6)	76.59% (3)	**77.10% (1.5)**	72.80% (5)	74.91% (4)
	Cancer	92.33% (3)	92.26% (5)	90.68% (6)	95.25% (2)	92.32% (2)	**96.49% (1)**
	Transfusion	76.60% (5)	**79.01% (1)**	78.74% (2)	77.27% (3)	76.80% (3)	76.52% (6)
	Banknote	91.33% (6)	98.18% (3)	99.13% (2)	**99.78% (1)**	94.42% (5)	97.34% (4)
	Magic	74.96% (6)	81.30% (3)	**84.63% (1)**	84.51% (2)	80.62% (2)	77.38% (5)
	Average rank	4.3	3.6	2.8	**1.9**	4.4	4.0
Multi-class	Wine	89.47% (6)	92.61% (3)	91.88% (4)	94.35% (2)	90.17% (5)	**95.84% (1)**
	Waveform	81.48% (4)	60.18% (6)	83.10% (3)	83.78% (2)	74.70% (5)	**84.00% (1)**
	Glass	58.05% (4)	57.99% (5)	58.49% (3)	**70.24% (1)**	50.37% (6)	63.32% (2)
	Red Wine	55.60% (6)	**68.67% (1)**	57.72% (4)	59.72% (2)	55.89% (5)	58.36% (3)
	Satimage	81.69% (4)	74.72% (6)	**89.15% (1)**	87.32% (2)	74.79% (5)	85.41% (3)
	Average rank	4.8	4.2	3.0	**1.8**	5.2	2.0

TABLE VI
FRIEDMAN AND HOLM TESTS TO COMPARE THE ACCURACY OF MICRO-EBRBS AND FIVE FRBCS CLASSIFIERS (α =0.1)

Item	Indicator	SLAVE	FH-GBML	FURIA	FARC-HD	Chi-FRBCS
Two-class	p value	0.7999	0.7353	0.3105	0.0759	0.7353
	Critical value	0.1000	0.0333	0.0250	0.0200	0.0333
	Hypothesis	Accepted	Accepted	Accepted	Accepted	Accepted
Multi-class	p value	0.0180	0.0630	0.3980	0.8658	0.0068
	Critical value	0.0250	0.0333	0.0500	0.1000	0.0200
	Hypothesis	**Rejected**	Accepted	Accepted	Accepted	**Rejected**

TABLE VII
COMPARISON OF COMPUTING TIME IN SECOND FOR MICRO-EBRBS AND FIVE FRBCS CLASSIFIERS

No. of classes	Dataset	SLAVE	FH-GBML	FURIA	FARC-HD	Chi-FRBCS	Micro-EBRBS
Two-class	Diabetes	569 s (5)	2s316 s (6)	4.3 s (3)	56 s (4)	**0.0 s (1.5)**	**0.0 s (1.5)**
	Cancer	1200 s (5)	4s947 s (6)	6.8 s (3)	249 s (4)	**0.1 s (1.5)**	**0.1 s (1.5)**
	Transfusion	212 s (5)	2s899 s (6)	4.2 s (3)	31 s (4)	**0.0 s (1.5)**	**0.0 s (1.5)**
	Banknote	502 s (5)	6s256 s (6)	6.0 s (3)	69 s (4)	**0.0 s (1.5)**	**0.0 s (1.5)**
	Magic	31,303 s (5)	387,010 s (6)	715 s (3)	8,655 s (4)	**1.0 s (1)**	1.1 s (2)
	Average rank	5	6	3	4	1.4	1.6
Multi-class	Wine	268 s (5)	1,394 s (6)	2.4 s (3)	75 s (4)	**0.0 s (1.5)**	**0.0 s (1.5)**
	Waveform	73,570 s (6)	45,255 s (5)	184 s (3)	2,490 s (4)	6.5 s (2)	**4.0 s (1)**
	Glass	611 s (5)	1,184 s (6)	3.2 s (3)	100 s (4)	**0.0 s (1.5)**	**0.0 s (1.5)**
	Red Wine	1,049 s (5)	9,434 s (6)	21 s (3)	590 s (4)	0.6 s (2)	**0.1 s (1)**
	Satimage	29,655 s (5)	62,579 s (6)	346 s (3)	8,876 s (4)	87 s (2)	**12 s (1)**
	Average rank	5.2	5.8	3	4	1.8	1.2

In summary, for the comparison of the Micro-EBRBS and EBRBS, the experiment results have shown that the former has much less computing time than the latter. Moreover, the Micro-EBRBS is comparable to the EBRBS in term of accuracy while much less number of rules and activated rules are used to address many classification datasets.

C. Comparative analysis between Micro-EBRBS and conventional classifiers

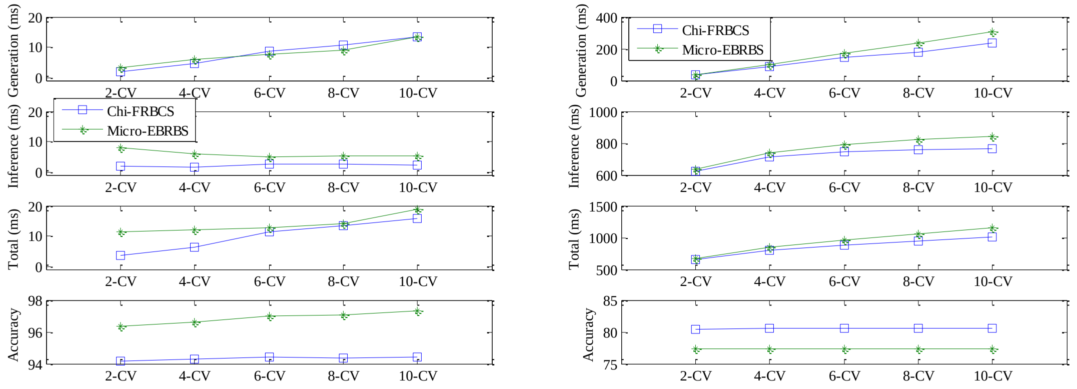
The second experiment aims to compare the accuracy of the Micro-EBRBS with the conventional classifiers, which include the FRBCS and the conventional machine-learning classifiers. Apart from the setting of the Micro-EBRBS introduced in Eqs. (26) and (27), the other classifiers are shown as follows:

(1) Chi-FRBCS [29]: it was proposed by Chi *et al.*, where the Penalized Certainty Factor (PCF) is used to calculate rule weights, the winner rule strategy is used as the fuzzy reasoning method, and the OVO is used to improve the performance of Chi-FRBCS in dealing with multi-class classification problems. Here, assume that the number of fuzzy labels is three for each attribute and these fuzzy labels modeled as triangular membership function.

(2) Other FRBCS classifiers: structural learning algorithm on vague environment (SLAVE), fuzzy hybrid genetic-based machine learning algorithm (FH-GBML), fuzzy unordered rule induction algorithm (FURIA), and fuzzy association rule-based classification method for high-dimensional problems (FARC-HD), they are all obtained from KEEL software. The setting of these FRBCS classifiers follows the previous study in [3].

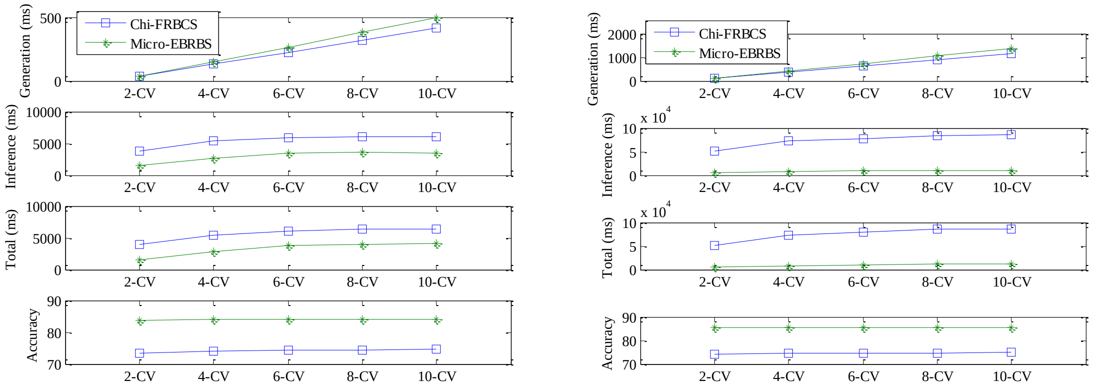
(3) Machine-learning classifiers: *k* nearest neighbor (KNN), Naïve Bayes (NB), Decision Tree (DT), Random Forest (RF), Artificial Neural Network (ANN), and Support Vector Machine (SVM), are all obtained from WEKA software. Apart from the default setting, 20% number of training data is set as neighbors for the KNN, 5% number of training data is set as the minimum number of data per leaf for the DT, the number of random trees is set as 5 for the RF, and the number of iterations is set as 10 for the ANN.

Table V shows the accuracy of Micro-EBRBS in comparison with five FRBCS classifiers, including SLAVE, FH-GBML, FURIA, FARC-HD, and Chi-FRBCS, over two and multi-class datasets, respectively, where the result of the best accuracy is highlighted in bold-face and the number in brackets denotes the rank of each classifier. For the two-class datasets, the accuracy



(a) Banknote (b) Magic

Fig. 15. Comparison of computing efficiency and accuracy between Chi-FRBCS and Micro-EBRBS for two-class datasets



(a) Waveform (b) Satimage

Fig. 16. Comparison of computing efficiency and accuracy between Chi-FRBCS and Micro-EBRBS for multi-class datasets

of FARC-HD is better than other FRBCS classifiers as well as Micro-EBRBS ranked at the 3rd place in term of average rank. For the multi-class datasets, the FARC-HD remains its advantages in dealing with classification problems over other classifiers. Despite the fact that the FARC-HD outperforms the Micro-EBRBS, it is still possible to see a considerable decrease in the average rank of Micro-EBRBS, namely from 4.0 to 2.0. This is so because the distributed belief degree is used in each EBR to express multiple classes and Micro-EBRBS therefore has excellent abilities to deal with multi-class problems.

In addition to the accuracy and average rank for each dataset and each classifier shown in Table V, Table VI provides the statistical analysis of accuracy while the Micro-EBRBS is selected as the control method for the Friedman and Holm tests. As shown in Table VI, apart from the SLAVE and Chi-FRBCS in the case of multi-class datasets, all hypotheses regarding five FRBCS classifiers are accepted, which means that although some of FRBCS classifiers, such as the FARCH-HD, FURIA, and FH-GBML, are better than Micro-EBRBS, none of hypotheses is in favor of the significant differences between the FRBCS classifiers and the Micro-EBRBS. For the SLAVE and Chi-FRBCS in the case of multi-class datasets, the hypothesis is rejected, which means that the significant differences can be found to show a better accuracy of Micro-EBRBS comparing to the SLAVE and Chi-FRBCS.

Table VII shows the computing time of Micro-EBRBS in comparison with the SLAVE, FH-GBML, FURIA, FARC-HD, and Chi-FRBCS, where the result of the best computing time is highlighted in bold-face and the number in brackets denotes the rank of each classifier. In both of two and multi-class datasets, the computing time of Micro-EBRBS is close to that of Chi-FRBCS, and is significantly faster than the other FRBCS classifiers. This is because some additional methodologies were used to improve the FRBCS classifiers, e.g. the genetic algorithm, which is an iterative optimization algorithm, and is one of the components of SLAVE and FARC-HD, leading to a time-consuming process while using those FRBCS classifiers.

Hence, in the application of FRBCS classifiers for addressing big data classification problems, the related works introduced in Section II-B were all based on Chi-FRBCS owning to its high efficient process of dealing with data.

In order to further compare with the Micro-EBRB and Chi-FRBCS, Figs. 15 and 16 show their time of rule generation scheme, inference scheme, and total as well as the accuracy over two and multi-class relatively large datasets, respectively.

For the two-class datasets Banknote and Magic, the computing time of rule generation scheme regarding the Chi-FRBCS closely approximates to that regarding the Micro-

TABLE VIII
COMPARISON OF COMPUTING EFFICIENCY FOR MICRO-EBRBS AND CHI-FRBCS

No. of classes	Dataset	Rule generation (ms)		Ratio	Inference (ms)		Ratio	Total (ms)		Ratio
		Micro-EBRBS	Chi-FRBCS		Micro-EBRBS	Chi-FRBCS		Micro-EBRBS	Chi-FRBCS	
Two-class	Banknote	**13.5**	**13.5**	1.000	5.2	**2.3**	2.261	18.7	**15.8**	1.184
	Magic	303.5	**237.2**	1.280	843.6	**767.0**	1.100	1,147.1	**1,004.5**	1.142
Multi-class	Waveform	496.7	**414.8**	1.197	**3,542.1**	6,034.2	0.587	**4,038.8**	6,449.0	0.626
	Satimage	1,386.8	**1,162.2**	1.193	**10,864.9**	85,759.8	0.127	**12,251.7**	86,922.0	0.141

TABLE IX
COMPARISON OF ACCURACY AND NUMBER OF RULES AND ACTIVATED RULES FOR MICRO-EBRBS AND CHI-FRBC

No. of classes	Dataset	No. of rules		Ratio	No. of activated rules		Ratio	Accuracy (%)		Ratio
		Micro-EBRBS	Chi-FRBCS		Micro-EBRBS	Chi-FRBCS		Micro-EBRBS	Chi-FRBCS	
Two-class	Banknote	**30.8**	**30.8**	1.000	22.3	**12.2**	1.828	**97.34**	94.42	1.031
	Magic	**354.1**	**354.1**	1.000	140.9	**99.3**	1.419	77.38	**80.62**	0.960
Multi-class	Waveform	**2,032.7**	**2,031.9**	1.000	1,494.2	**451.9**	3.307	**83.97**	74.66	1.125
	Satimage	**3,182.0**	**3,182.0**	1.000	945.6	**485.6**	1.947	**85.41**	74.79	1.142

TABLE X
COMPARISON OF ACCURACY AND NUMBER OF RULES AND ACTIVATED RULES FOR MICRO-EBRBS AND CHI-FRBC

No. of classes	Dataset	KNN	NB	DT	RF	ANN	SVM	Micro-EBRBS
Two-class	Diabetes	74.30% (6)	76.84% (4)	77.61% (3)	77.86% (2)	**79.13% (1)**	66.92% (7)	74.91% (5)
	Cancer	94.02% (4)	93.15% (6)	93.32% (5)	95.61% (3)	96.31% (2)	62.74% (7)	**96.49% (1)**
	Transfusion	76.20% (4)	75.40% (5)	**78.34% (1)**	72.33% (7)	76.34% (3)	75.27% (6)	76.52% (2)
	Banknote	92.93% (5)	84.26% (7)	90.31% (6)	99.20% (2)	98.25% (3)	**100.00% (1)**	97.34% (4)
	Magic	74.79% (5)	72.69% (6)	81.17% (3)	**86.14% (1)**	83.73% (2)	65.88% (7)	77.38% (4)
	Average rank	4.8	5.6	3.6	3.0	2.2	5.6	3.2
Multi-class	Wine	**97.19% (1.5)**	96.63% (3)	92.13% (6)	94.38% (5)	**97.19% (1.5)**	44.38% (7)	95.84% (4)
	Waveform	82.88% (4)	81.02% (5)	73.26% (7)	80.20% (6)	85.78% (2)	**86.10% (1)**	84.00% (3)
	Glass	61.21% (5)	47.66% (6)	67.76% (3)	**71.50% (1)**	39.72% (7)	69.16% (2)	63.32% (4)
	Red Wine	57.22% (5)	54.97% (7)	58.16% (3)	**64.35% (1)**	57.04% (6)	57.85% (4)	58.36% (2)
	Satimage	74.34% (6)	79.70% (4)	79.63% (5)	**88.94% (1)**	86.28% (2)	26.08% (7)	85.41% (3)
	Average rank	4.3	5.0	4.8	**2.8**	3.7	4.2	3.2

TABLE XI
FRIEDMAN AND HOLM TESTS TO COMPARE THE ACCURACY OF MICRO-EBRBS AND SIX MACHINE-LEARNING CLASSIFIERS (α =0.1)

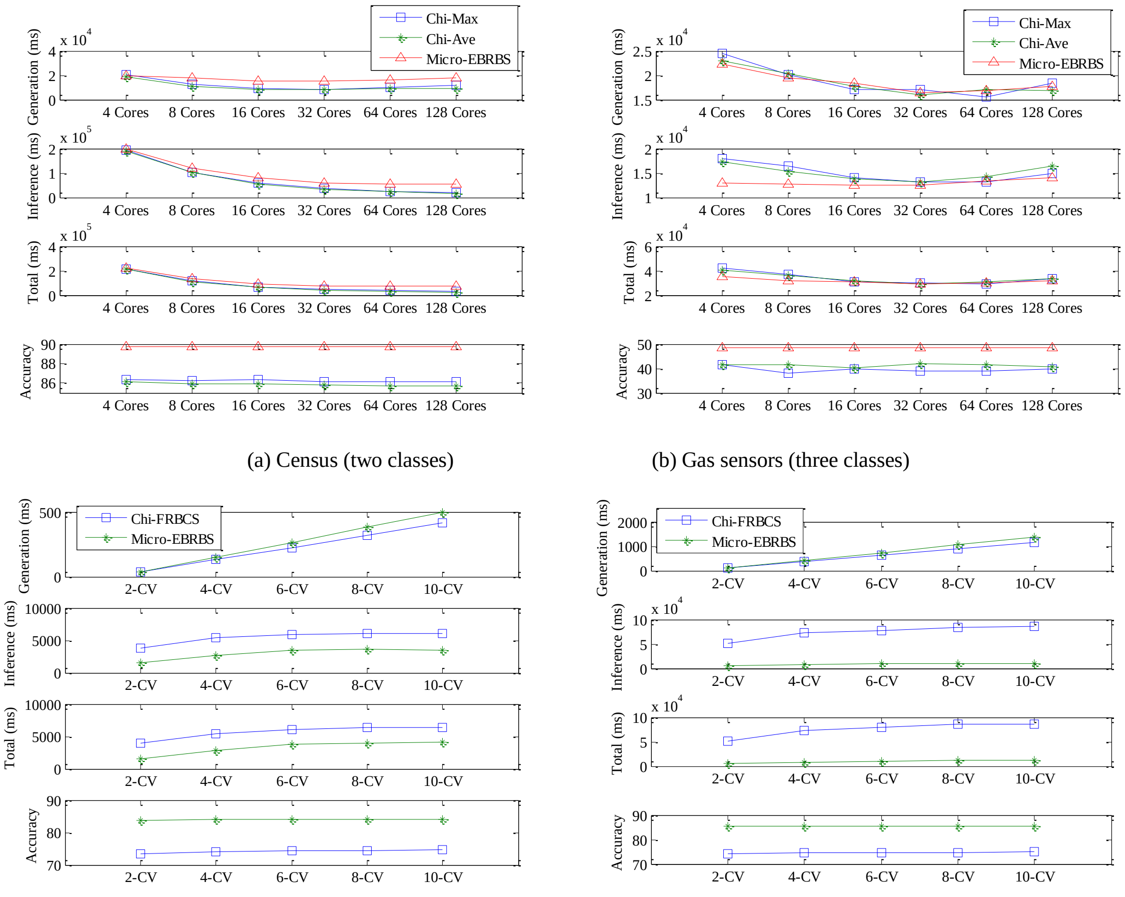
Item	Indicator	KNN	NB	DT	RF	ANN	SVM
Two-class	p value	0.2416	0.0790	0.7697	0.8836	0.4642	0.0790
	Critical value	0.0250	0.0167	0.0500	0.1000	0.0333	0.0167
	Hypothesis	Accepted	Accepted	Accepted	Accepted	Accepted	Accepted
Multi-class	p value	0.4208	0.1877	0.2416	0.7697	0.7144	0.4642
	Critical value	0.0250	0.0167	0.020	0.1000	0.0500	0.0333
	Hypothesis	Accepted	Accepted	Accepted	Accepted	Accepted	Accepted

EBRBS from 2-CV to 10-CV. Meanwhile, for the dataset Banknote, the Micro-EBRBS is slightly better than the Chi-FRBCS in terms of the computing time of inference scheme and the accuracy. But for the dataset Magic, the Chi-FRBCS is slightly better than the Micro-EBRBS regarding the computing time and accuracy. For the multi-class datasets Waveform and Satimage, apart from the computing time of rule generation scheme, the Micro-EBRBS is much better than the Chi-FRBCS in term of the computing time of the inference scheme and the accuracy.

In order to show the detailed comparison of Micro-EBRBS and Chi-FRBCS, Tables VIII and IX provide the results and their ratio obtained from 10-CV, such as the rule generation time, inference time, total time, number of rules, number of activated rules, and accuracy. Table VIII shows that the Chi-FRBCS has less computing time than the Micro-EBRBS in terms of the rule generation scheme for both two and multi-class datasets and the inference scheme for two-class datasets. However, the Micro-EBRBS has less computing time of the inference scheme than the Chi-FRBCS in the multi-class datasets. As a result, the Chi-FRBCS has better total time and their ratio are 1.184 and 1.142 for the two-class datasets Banknote and Magic, and the Micro-EBRBS has better total time and their ratio are 0.626 and 0.141 for the multi-class datasets Waveform and Satimage.

Table IX shows that the number of rules in the Chi-FRBCS is the same as the Micro-EBRBS, but the number of activated rules in the Chi-FRBCS is smaller than the Micro-EBRBS, where the maximum ratio is obtained from the multi-class dataset Waveform and its ratio is 3.307, and except for the dataset Magic, the accuracy of the Chi-FRBCS is worse than the Micro-EBRBS for all two and multi-class datasets, where the maximum ratio is obtained from the multi-class dataset Satimage and its ratio is 1.142.

In order to compare the accuracy of the Micro-EBRBS with the conventional machine-learning classifiers, Table X shows the accuracy of seven classifiers for ten classification datasets. The number in brackets denotes the rank of each classifier and the best result is marked as bold in Table X. For the two-class datasets, the accuracy of the RF and ANN are better than the Micro-EBRBS ranked at the 3rd place, and are further better than the KNN, NB, DT, and SVM. For the multi-class dataset,



(a) Census (two classes)

(b) Gas sensors (three classes)

(c) Covtype (seven classes)

(d) Poker (ten classes)

Fig. 17. Comparison of computing efficiency and accuracy for Chi-Max, Chi-Ave, and Micro-EBRBS

the Micro-EBRBS can reach the second best average rank better than the KNN, DT, NB, ANN, and SVM. From Table X, RF obtains the best average rank in both two and multi-class datasets. This is partly because RF is an ensemble learning method that operates by constructing a multitude of DTs. Hence, in one sense, the ensemble learning used in the RF can also use to improve the Micro-EBRB.

Table XI shows the statistical analysis of accuracy while the Micro-EBRBS is selected as the control method for the Friedman and Holm tests. From Table XI, all hypotheses regarding the six machine-learning classifiers are accepted, namely the Micro-EBRBS and the machine-learning classifiers have a similar accuracy for two and multi-class datasets without significant differences.

In summary, for the comparison of the Micro-EBRBS and the conventional classifiers, the experiment results have proven that the Micro-EBRBS has the similar accuracies with the conventional FRBCS and the conventional machine-learning classifiers but its computing time is much less than the conventional FRBCS classifiers except for the Chi-FRBCS.

It is worth noting that the Micro-EBRBS has the higher accuracy but less computing time than the Chi-FRBCS for multi-class datasets.

D. Comparative analysis between Micro-EBRBS and big data FRBCS classifiers

The third experiment aims to compare the accuracy and the computing efficiency of the Micro-EBRBS with the big data FRBCS classifiers, where the core supporting theory of the big data FRBCS classifiers is shown in Section IV-C and they include the following two versions [29]:

(1) Chi-FRBCS-BigData-Max (Chi-Max for short): In this big data classifier, the rule generation scheme searches for the fuzzy rules with the same fuzzy label. Among these fuzzy rules, only the fuzzy rule with the highest rule weight is maintained in the fuzzy rule base.

(2) Chi-FRBCS-BigData-Ave (Chi-Ave for short): In this big data classifier, the rule generation scheme also searches for the fuzzy rules with the same fuzzy label. Afterwards, the average rule weight of the fuzzy rules that have the same class is computed. Finally, the fuzzy rules with the greatest average rule weight is kept in the fuzzy rule base.

Fig. 17 shows the computing time and the accuracy of the Micro-EBRBS in comparison with the big data FRBCS classifiers, including Chi-Max and Chi-Ave, over two-class and multi-class datasets, respectively.

From Fig. 17, the Micro-EBRBS has the significantly better computing efficiency for the seven-class dataset Covtype and the ten-class dataset Poker, similar one for the three-class

TABLE XII
COMPARISON OF COMPUTING TIME AND ACCURACY FOR CHI-MAX, CHI-AVE AND MICRO-EBRBS UNDER FOUR CORES

Indicator	Classifier	Census		Gas sensors		Covtype		Poker	
		Value	Ratio	Value	Ratio	Value	Ratio	Value	Ratio
Rule generation (ms)	Micro-EBRBS	19,817	-	**22,269**	-	**50,062**	-	**36,099**	-
	Chi-Max	20,007	0.991	24,349	0.915	68,246	0.734	388,942	0.093
	Chi-Ave	**18,759**	1.056	23,000	0.968	67,656	0.740	376,419	0.096
Inference (ms)	Micro-EBRBS	199,324	-	**12,879**	-	**129,858**	-	**1,818,235**	-
	Chi-Max	191,375	1.042	17,845	0.722	8,310,398	0.016	141,730,759	0.013
	Chi-Ave	**190,816**	1.045	17,223	0.748	8,307,228	0.016	139,971,396	0.013
Total (ms)	Micro-EBRBS	219,141	-	**35,148**	-	**179,920**	-	**1,854,334**	-
	Chi-Max	211,382	1.037	42,194	0.833	8,378,644	0.022	142,119,701	0.013
	Chi-Ave	**209,575**	1.046	40,223	0.874	8,374,884	0.022	140,347,815	0.013
Accuracy (%)	Micro-EBRBS	**89.69**	-	**48.60**	-	**70.24**	-	**57.20**	-
	Chi-Max	86.34	1.039	41.34	1.176	65.19	1.078	52.79	1.084
	Chi-Ave	86.14	1.041	41.68	1.166	67.08	1.047	53.53	1.069

TABLE XIII
FRIEDMAN AND HOLM TESTS TO COMPARE THE ACCURACY OF MICRO-EBRBS, CHI-MAX, AND CHI-AVE (α =0.1)

Indicator	Chi-Max	Chi-Ave
p value	0.0133	0.0771
Critical value	0.05	0.1
Hypothesis	Rejected	Rejected

dataset Gas sensors and worse one for the two-class dataset Census comparing to big data FRBCS classifiers. For the ten, seven, and three-class datasets, the computing efficiency differences between these classifiers diminish gradually from 4 cores to 128 cores. Additionally, from the datasets Census and Gas sensors, while 128 cores are used in the cluster computing, the computing time is more than the result obtained from 64 cores mainly because of the increasing cluster costs. So, determination of the number of cores is important to improve the computing efficiency of the Micro-EBRBS.

From the comparison of accuracy, the Micro-EBRBS reflects a better robustness than the big data FRBCS classifiers because the accuracy is consistent for all two and multi-class datasets when varying the number of cores used in the cluster computing, but the accuracy of the big data FRBCS classifiers are changeable. Therefore, comparing to the big data FRBCS classifiers, the Micro-EBRBS is able to provide exactly the same classifier while implementing by the Apache Spark. More specifically, Table XII shows the value and ratio of the Micro-EBRBS, Chi-Max, and Chi-Ave under 4 cores, where the best result is marked as bold.

In term of the computing time, the big data FRBCS classifiers have a slight advantage while addressing the two-class dataset Census and all ratios of rule generation, inference, and total time are less than 1.056. However, the computing time of the Micro-EBRBS is much better than both Chi-Max and Chi-Ave while increasing number of classes, i.e. ten-class dataset Poker whose ratios are all smaller than 0.1 for the rule generation, inference, and total time. In term of the accuracy, the Micro-EBRBS is better than the big data FRBCS classifiers and the range of the ratio can be expressed as [1.039, 1.176] for four datasets.

Additionally, in order to detect significant differences among the accuracy of the Micro-EBRBS, Chi-Max, and Chi-Ave, the Friedman and Holm tests are carried out. Table XIII shows that two hypotheses are rejected because there are significant differences among the obtained results with a level of significance of α =0.1. Hence, in the datasets Census, Gas sensors, Covtype, and Poker, the accuracy of the Micro-EBRBS is better than that of the Chi-Max and Chi-Ave.

In summary, according to the comparison of the Micro-EBRBS and the big data FRBCS classifiers, it is evident that the Micro-EBRBS has the advantage of using less computing time and obtaining better accuracy and robustness for the big data multi-class datasets.

E. *Time complexity Comparison*

In this subsection, a comparison of the Micro-EBRBS, EBRBS, and Chi-FRBCS is provided to show which one has a better time complexity to deal with big data multi-class classification problems.

Suppose that there are L rules in EBRB (or sample data), $\overline{L}$ rules in the reduced EBRB, S testing data, M antecedent attribute with J_i reference values, and N classes. The time complexity of different schemes in the EBRBS, Micro-EBRBS, and Chi-FRBCS is shown in Table XIV based on discussions in Section II-B, Section III-C, and [29]. Additionally, in order to clearly compare three classifiers, their time complexity can be simplified under the assumptions: (1) the number of sample data L is much bigger than the square of the number of classes N^2; and (2) the total number of reference values for antecedent attributes $\sum_{i=1}^{M} J_i$ is much bigger than the number of classes N.

Remark 11. It is clear from Table XIV that the time complexity of the Micro-EBRBS is better than the EBRBS in both the rule generation and the inference schemes due to the following three reasons:

(1) The Micro-EBRBS has a simple process in the rule generation scheme because of excluding the calculation of rule weights comparing to the EBRBS.

(2) The Micro-EBRBS has less number of rules in the reduced EBRB owning to using the proposed rule reduction method to downsize the EBRB comparing to the EBRBS.

(3) The Micro-EBRBS can be implemented by using the Apache Spark to generate rules and classify test input data in parallel thanks to the better performance of independence in the rule generation and the inference schemes

Remark 12. It is clear from Table XIV that the Chi-FRBCS is more efficient than the Micro-EBRBS in the rule generation

TABLE XIV
TIME COMPLEXITY OF EBRBS, MICRO-EBRBS, AND CHI-FRBCS

	Rule generation scheme	Inference scheme	Rule generation scheme $(L >> N^2)$	Inference scheme $(\sum_{i=1}^{M} J_i >> N)$
EBRBS	$O(L^2 \cdot \sum_{i=1}^{M} J_i + L^2 \cdot N)$	$O(S \cdot L \cdot (\sum_{i=1}^{M} J_i + N))$	$O(L^2 \cdot \sum_{i=1}^{M} J_i + L^2 \cdot N)$	$O(S \cdot L \cdot \sum_{i=1}^{M} J_i)$
Micro-EBRBS	$O(L \cdot \bar{L} \cdot \sum_{i=1}^{M} J_i + L \cdot \bar{L} \cdot N)$	$O(S \cdot \bar{L} \cdot (\sum_{i=1}^{M} J_i + N))$	$O(L \cdot \bar{L} \cdot \sum_{i=1}^{M} J_i + L \cdot \bar{L} \cdot N)$	$O(S \cdot \bar{L} \cdot \sum_{i=1}^{M} J_i)$
Chi-FRBCS	$O(L \cdot \sum_{i=1}^{M} J_i + L \cdot \bar{L} + N^2 \cdot \bar{L})$	$O(S \cdot \bar{L} \cdot \sum_{i=1}^{M} N^2)$	$O(L \cdot \sum_{i=1}^{M} J_i + L \cdot \bar{L})$	$O(S \cdot \bar{L} \cdot \sum_{i=1}^{M} N^2)$

scheme. However, the Micro-EBRBS is more efficient than the Chi-FRBCS in the inference scheme, especially for the multi-class classification problems. Additionally, the computing efficiency of both the two big data classifiers can be further improved by using the Apache Spark.

V. CONCLUSIONS

In this study, the analysis of the rule weight calculation and the ER algorithm were carried out to investigate the approach of reducing the time complexity of the EBRBS, a popular advanced rule-based system, followed by a ER-C algorithm and a domain division-based rule reduction method proposed for developing a micro version of EBRBS with high computing efficiency, called Micro-EBRBS. Furthermore, the Apache Spark was introduced to implement the Micro-EBRBS for better dealing with big data multi-class classification problems. 14 classification datasets were used to validate the accuracy and computing efficiency of the Micro-EBRBS in comparison with the EBRBS, the conventional FRBCS and machine-learning classifiers, and the big data FRBCS classifiers. The detailed contributions are summarized as follows:

(1) The non-necessity of the rule weight calculation and the ER algorithm involved in the EBRBS were investigated, in which the former demonstrates that it is unnecessary to calculate rule weights for each EBR under the assumption of large amount of data, the latter prove that the ER-C algorithm has the same functionality as the ER algorithm under the assumption of classification problems.

(2) The division point and division domain were defined to divide the input space of the EBRBS into multiple local input spaces. Accordingly, the rule clustering strategy and rule reduction strategy were further defined to propose a domain division-based rule reduction method to downsize EBRB.

(3) The Micro-EBRBS, which includes the rule reduction and the ER-C algorithm but excludes the rule weight calculation in comparison with the EBRBS, and its implementation based on the Apache Spark were developed to deal with big data multi-class classification problems, which were then validated through the detailed case studies. The results have shown advantages of the Micro-EBRBS over the existing methods in terms of computing efficiency and classification accuracy.

For the future research, the application of Micro-EBRBS and further improvement to make it more effective to deal with the practical problem with uncertain and imbalance data.

APPENDIX A. INFERENCE SCHEME OF EBRBS FOR CLASSIFICATION PROBLEMS

The inference scheme of the EBRBS mainly includes two steps [22]: (1) calculation of activation weights for each EBR using distance measure and (2) integration of activated rules for estimating classes using the ER algorithm.

One thing to note is that the procedure of the first step is the same as the Micro-EBRBS. Hence, the activation weight w_k (k=1,…, L) can be obtained by using Eq. (23), which shows a positive correlation between the w_k and rule weight θ_k. After calculating activation weights, all activated rules should be integrated using the analytical ER algorithm [32], [37]:

$$\begin{aligned}\beta_n = (&\prod_{k=1}^{L}(w_k\beta_{n,k} + 1 - w_k\sum\nolimits_{i=1}^{N}\beta_{i,k}) - \\ &\prod_{k=1}^{L}(1 - w_k\sum\nolimits_{i=1}^{N}\beta_{i,k}))/ \\ &(\sum_{i=1}^{N}\prod_{k=1}^{L}(w_k\beta_{i,k} + 1 - w_k\sum\nolimits_{j=1}^{N}\beta_{j,k}) - \\ &(N-1)\prod_{k=1}^{L}(1 - w_k\sum\nolimits_{j=1}^{N}\beta_{j,k}) - \prod_{k=1}^{L}(1 - w_k))\end{aligned} \quad \text{(A1)}$$

The integrated belief distribution of the test input data $\boldsymbol{x}$ is:

$$f(\boldsymbol{x}) = \{(D_n, \beta_n); n = 1,..., N\} \quad \text{(A2)}$$

For classification problems, suppose D_n denotes the nth class, the estimated class of the EBRBS can be obtained by seeking the greatest belief degree.

$$f(\boldsymbol{x}) = D_n, n = \arg\max_{i=1,..,N}\{\beta_i\} \quad \text{(A3)}$$

APPENDIX B. BELIEF DISTRIBUTION GENERATION AND RULE WEIGHT CALCULATION OF EBRBS

The belief distribution generation and the rule weight calculation are important processes in the rule generation scheme of the EBRBS and their details can be refer to [22].

One thing to note is that the detailed procedure of the belief distribution generation is the same as the Micro-EBRBS. Hence, for the kth (k=1,…, L) EBR, the belief distributions of the ith antecedent attribute S_i^k={($A_{i,j}$, $\alpha_{i,j}^k$); j= 1,…, J_i} (i=1,…, M) and the consequent attribute S^k={(D_n, $\beta_{n,k}$); n=1,…, N} can be obtained by using Eqs. (16) and (19).

Definition B.1 (Similarity of two belief distributions). Suppose two belief distribution P=(p_1,…, p_T) and Q=(q_1,…, q_3), then the similarity of P and Q can be calculated as follows:

Pseudocode of rule generation scheme of the Chi-FRBCS

```
Input: $\mu_{y_t}(\boldsymbol{x}_t)$ and $R_{y_t}(\boldsymbol{x}_t)$ denote the membership degree and the fuzzy label set of the fuzzy rule which is transformed from the sample input data $\boldsymbol{x}_t$ and the class $y_t$, $y_t \in \{D_1,\ldots, D_N\}$; $Class_{i,j}(R_k)$ and $w_{i,j}(R_k)$ denote the class and the rule weight of the fuzzy rule $R_k$ while considering the $i$th and the $j$th class as a two-class classification problem.
Output: the set of fuzzy rules $\boldsymbol{FRB}$
01 Initialize $\boldsymbol{FRB}=\{\}$;
02 For each sample input data $\boldsymbol{x}_t$ in $\{x_1,\ldots,x_L\}$
03   Initialize $\mu_{y_t}(\boldsymbol{x}_t)=1$ and $R_{y_t}(\boldsymbol{x}_t)=\{\}$;
04   For each input data $x_{t,i}$ in $\boldsymbol{x}_t=\{x_{t,1},\ldots,x_{t,M}\}$
05     Calculate $\mu_{y_t}(\boldsymbol{x}_t)=\mu_{y_i}(\boldsymbol{x}_t)*\max\{\mu_{A_{i,j}}(x_{t,i}); j=1,\ldots,J_i\}$;
06     Add $R_{y_t}(\boldsymbol{x}_t)=R_{y_t}(\boldsymbol{x}_t)\cup\{A_{i,s}\}; s=\arg\max_{s=1,\ldots,J_i}\{\mu_{A_{i,j}}(x_{t,i})\}$;
07   End for
08   If $\neg R_k \in \boldsymbol{FRB}$ and $R_k = R_{y_t}(\boldsymbol{x}_t)$ then
09     Update $\mu_{y_t}(R_k)=\mu_{y_t}(R_k)+\mu_{y_t}(\boldsymbol{x}_t)$ ;
10   Else if $R_{y_t}(\boldsymbol{x}_t) \notin \boldsymbol{FRB}$ then
11     Add $\boldsymbol{FRB}=\boldsymbol{FRB}\cup\{R_{y_t}(\boldsymbol{x}_t)\}$;
12     Initialize $\mu_{y_t}(R_k)=\mu_{y_t}(\boldsymbol{x}_t)$;
13   End if
14 End for
15 For each fuzzy rule $R_k$ in $\boldsymbol{FRB}$
16   For each class $D_i$ and $D_j$ (i < j) in $\{D_1,\ldots,D_N\}$
17     Initialize $Class_{i,j}(R_k)=D_s$, $s$=arg max$_{n=i,j}$\{ $\mu_n(R_k)$ \};
18     Initialize $w_{i,j}(R_k)=\left|\mu_i(R_k)-\mu_j(R_k)\right|/(\mu_i(R_k)+\mu_j(R_k))$;
19   End for
20 End for
```

Pseudocode of inference scheme of the Chi-FRBCS

```
Input: $S_{i,j}(\boldsymbol{x}_t)$ denotes the score of test input data $\boldsymbol{x}_t$ while considering the $i$th and the $j$th class as a two-class classification problem; $\mu(\boldsymbol{x}_t)$ denotes the membership degree of the test input data $\boldsymbol{x}_t$; $w_{i,j}(R_k)$ denotes the rule weight of the fuzzy rule $R_k$ while considering the $i$th and the $j$th class as a two-class classification problem.
Output: The estimated class $Class(\boldsymbol{x}_t)$ ($t$=1,…, $S$)
01 For each test input data $\boldsymbol{x}_t$ in $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_S\}$
02   Initialize $S_{i,j}(\boldsymbol{x}_t)$= 0 ($i$, $j$ = 1,…,$N$)
03     For each $D_i$ and $D_j$ ($i$ < $j$) in $\{D_1,\ldots,D_N\}$
04       For each fuzzy rule $R_k$ in $\boldsymbol{FRB}$
05         Initialize $\mu(\boldsymbol{x}_t)=1$;
06           For each input data $x_{t,m}$ in $\boldsymbol{x}_t=\{x_{t,1},\ldots,x_{t,M}\}$
07             Calculate $\mu(\boldsymbol{x}_t)=\mu(\boldsymbol{x}_t)*\mu_{A_m^k}(x_{t,m})$;
08           End for
09           If $Class_{i,j}(R_k)=D_i$ and $S_{i,j}(\boldsymbol{x}_t)$< $\mu(\boldsymbol{x}_t)*w_{i,j}(R_k)$ then
10             Calculate $S_{i,j}(\boldsymbol{x}_t)$= $\mu_{y_t}(\boldsymbol{x}_t)*w_{i,j}(R_k)$;
11           Else if $Class_{i,j}(R_k)=D_j$ and $S_{j,i}(\boldsymbol{x}_t)$< $\mu(\boldsymbol{x}_t)*w_{i,j}(R_k)$ then
12             Calculate $S_{j,i}(\boldsymbol{x}_t)$= $\mu(\boldsymbol{x}_t)*w_{i,j}(R_k)$;
13           End if
14       End for
15   End for
16   Calculate $Class(\boldsymbol{x}_t)=D_n$; $n$=arg max$_{i=1,\ldots,N}$\{ $\sum_{j=1}^{N} S_{i,j}(\boldsymbol{x}_t)$ \}
17 End for
```

$$
\begin{aligned}
Sim(P,Q) &= 1-d(P,Q) \\
&= 1-\min\left\{1, \sqrt{\sum_{t=1}^{T}(p_t - q_t)^2}\right\}
\end{aligned} \tag{B1}
$$

where $d(P, Q)$ denotes the distance between P and Q.

Based on Definition B.1, for the kth (k=1,…, L) EBR, the similarity of rule antecedent (SRA) and the similarity of rule consequent (SRC) can be calculated as follows:

$$
\begin{aligned}
SRA(R_l,R_k) &= \min_{t=1,\ldots,M}\left\{Sim(S_t^l,S_t^k)\right\} \\
&= \min_{t=1,\ldots,M}\left\{1-\min\left\{1,\sqrt{\sum_{j=1}^{J_t}(\alpha_{t,j}^l-\alpha_{t,j}^k)^2}\right\}\right\};
\end{aligned} \tag{B2}
$$

$$
\begin{aligned}
SRC(R_l,R_k) &= Sim(S^l,S^k) \\
&= 1-\min\left\{1,\sqrt{\sum_{n=1}^{N}(\beta_{n,l}-\beta_{n,k})^2}\right\};
\end{aligned} \tag{B3}
$$

where l=1,…, L and $l \neq k$; S_t^k denotes the belief distribution of the tth antecedent attribute in the kth EBR; S^k denotes the belief distribution of the consequent attribute in the kth EBR.

Definition B.2 (Consistency of EBRs). Suppose the SRA and the SRC of the lth (l=1,…, L) and the kth (k=1,…, L; $k \neq l$) EBRs are $SRA(R_l, R_k)$ and $SRC(R_l, R_k)$, respectively, then the consistency of the rules R_l and R_k can be calculated as follows:

$$
Cons(R_l,R_k)=\exp\left\{-\frac{\left(\dfrac{SRA(R_l,R_k)}{SRC(R_l,R_k)}-1\right)^2}{\left(\dfrac{1}{SRA(R_l,R_k)}\right)^2}\right\} \tag{B4}
$$

Based on Definition B.2, the inconsistency degree of the kth EBR can be calculated as follows:

$$
\begin{aligned}
Incons(R_k) &= \sum_{l=1,l\neq k}^{L}\left(1-Cons(R_l,R_k)\right) \\
&= \sum_{l=1,l\neq k}^{L}\left(1-\exp\left\{-\frac{\left(\dfrac{SRA(R_l,R_k)}{SRC(R_l,R_k)}-1\right)^2}{\left(\dfrac{1}{SRA(R_l,R_k)}\right)^2}\right\}\right)
\end{aligned} \tag{B5}
$$

Finally, the rule weight is calculated as follows:

$$
\theta_k = 1-\frac{Incons(R_k)}{\sum_{j=1}^{L}Incons(R_j)} \tag{B6}
$$

Appendix C. Rule generation and inference schemes of Chi-FRBCS

A. *Time complexity of the rule generation scheme*

The rule generation scheme of the Chi-FRBCS, which consists of the PCF and OVO, is introduced as follows:

As shown in the pseudocode of the rule generation scheme, the time complexity of calculating $\mu_{y_t}(\boldsymbol{x}_t)$ shown in the 5th line and updating the set ***FRB*** shown in the 8th to the 13th lines is $O(\sum_{i-1}^{M} J_i)$ and $O(\overline{L})$, respectively, for each sample data, where $\overline{L}$ is the number of fuzzy rules. In addition, from the 15th to the 20th lines, its time complexity is $O(\overline{L} \times N^2)$. As a result, the time complexity of the rule generation scheme involved in the Chi-FRBCS is shown as follows:

$$O(L \times (\sum_{i=1}^{M} J_i + \overline{L}) + \overline{L} \times N^2) \tag{C1}$$

B. Time complexity of the inference scheme

The inference scheme of the Chi-FRBCS, which consists of the winning rule strategy and OVO, is introduced as follows:

From the pseudocode of the rule generation scheme, the time complexity of calculating $\mu(\boldsymbol{x_t})$ shown in the 7th line is $O(M)$ for each test input data, fuzzy rule, and two-class classification problem. Hence, the time complexity of the inference scheme involved in the Chi-FRBCS is shown as follows:

$$O(S \times \overline{L} \times M \times N^2) \tag{C2}$$

References

[1] Akashdeep, I. Manzoor, and N. Kumar, “A feature reduced intrusion detection system using ANN classifier,” *Expert Syst. Appl.*, vol. 88, pp. 249-459, Dec., 2017.

[2] N. Al-Najdawi, M. Biltawi, and S. Tedmori, “Mammogram image visual enhancement, mass segmentation and classification,” *Appl. Soft Comput.*, vol. 35, pp. 175-185, Oct., 2015.

[3] J. Alcala-Fdez, R. Alcala, and F. Herrera, “A Fuzzy Association Rule-Based Classification Model for High-Dimensional Problems With Genetic Rule Selection and Lateral Tuning,” *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 5, pp. 857-872, Oct. 2011.

[4] A. Kattt, “Spark: Lightning-fast Cluster Computing,” Apache spark, Nov. 21, 2011. [Online]. Available: https://amplab.cs.berkeley.edu/projects/spark-lightning-fast-cluster-computing/

[5] W. Ashlock and S. Datta, “Evolved Features for DNA Sequence Classification and Their Fitness Landscapes,” *IEEE Trans. Evolut. Comput.*, vol. 17, no. 2, pp. 185-197, Apr., 2013.

[6] A. Calzada, J. Liu, H. Wang, and A. Kashyap, “A New Dynamic Rule Activation Method for Extended Belief Rule-Based Systems,” *IEEE Trans. Knowl. Data Eng.*, vol. 27 no. 4, pp. 880-894, Apr., 2015.

[7] L. L. Chang, Z. J. Zhou, Y. You, L. H. Yang, and Z. G. Zhou, “Belief rule based expert system for classification problems with new rule activation and weight calculation procedures,” *Inf. Sci.*, vol. 336, pp. 75-91, 2016.

[8] Z. Chi, H. Yan, and T. Pham, “Fuzzy Algorithms with Applications to Image Processing and Pattern Recognition,” World Scientific, 1996, pp. 139-187.

[9] J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters,” *Commun. ACM*, vol. 51, no. 1, pp. 107-113, 2008.

[10] M. Elkano, M. Galar, J. Sanz, and H. Bustince, “CHI-BD: A Fuzzy Rule-Based Classification System for Big Data classification problems,” *Fuzzy Sets Syst.*, vol. 348, no. 1, pp. 75-101, Oct. 2018.

[11] M. Espinilla, J. Medina, A. Calzada, J. Liu, L. Martinez, and C. Nugent, “Optimizing the configuration of a heterogeneous architecture of sensors for activity recognition, using the extended belief rule-based inference methodology,” *Microprocess Microsy.*, vol. 52, pp. 381-390, Jul., 2017.

[12] A. Fernández, S. D. Rio, A. Bawakid, and F. Herrera, “Fuzzy rule based classification systems for big data with MapReduce: granularty analysis,” *Adv. Data Anal. Classi.*, vol. 11, no. 4, pp. 1-20, Dec., 2017.

[13] A. Fernández, A. Altalhi, S. Alshomrani, and F. Herrera, “Why linguistic fuzzy rule based classification systems perform well in big data applications?” *Int. J. Comput. Intell. Syst.*, vol. 10, pp. 1211-1225, 2017.

[14] S. García, A. Fernández, J. Luengo, and F. Herrera, “Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining-experimental analysis of power,”*Inf. Sci.*, vol. 180, pp. 2044-2064, 2010.

[15] A. González and R. Pérez, “SLAVE: A Genetic Learning System Based on an Iterative Approach,” *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 2, pp. 176-191, Apr., 1999.

[16] J. Hühn and E. Hüllermeier, “FURIA: an algorithm for unordered fuzzy rule induction,” *Data Min. Knowl. Disc.*, vol. 19, no. 3, pp. 293-319, Dec., 2009.

[17] H. Ishibuchi, T. Nakashima, and T. Muratam, “A fuzzy classifier system that generates fuzzy if-then rules for pattern classification problems,” *IEEE Int. Conf. Evolu. Comput.*, vol. 2, pp. 759-764, Nov./ Dec., 1995.

[18] H. Ishibuchi, T. Yamamoto, and T. Nakashima, “Hybridization of Fuzzy GBML Approaches for Pattern Classification Problems,” *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 35, no. 2, pp. 359-365, Apr., 2005.

[19] L. M. Jiao, T. Denoeux, and Q. Pan, “A Hybrid Belief Rule-Based Classification System Based on Uncertain Training Data and Expert Knowledge,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 12, pp. 1711-1723, Dec., 2016.

[20] L. M. Jiao, Q. Pan, T. Denœux, Y. Liang, and X. X. Feng, “Belief rule-based classification system: Extension of FRBCS in belief functions framework,” *Inf. Sci.*, vol. 309, pp. 26-49, Jul., 2015.

[21] B. Krawczyk, M. Galar, M. Wozniak, H. Bustince, and F. Herrera, “Dynamic ensemble selection for multi-class classification with one-class classifiers,” *Pattern Recog.*, vol. 83, pp. 34-51, Nov., 2018.

[22] J. Liu, L. Martinez, A. Calzada, and H. Wang, “A novel belief rule base representation, generation and its inference methodology,” *Knowl. Based Syst.*, vol. 53, pp. 129-141, Nov., 2013.

[23] Z. G. Liu, Q. Pan, J. Dezert, J. W. Han, and Y. He, “Classifier Fusion with Contextual Reliability Evaluation,” *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1605-1618, May, 2018.

[24] Z. G. Liu, Q. Pan, J. Dezert, and A. Martin, “Combination of Classifiers with Optimal Weight Based on Evidential Reasoning,” *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 3, pp. 1217-1230, Jun., 2018.

[25] Z. G. Liu, Q. Pan, J. Dezert, and G. Mercier, “Hybrid Classification System for Uncertain Data,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 10, pp. 2783-2790, Otc., 2017.

[26] V. Lopez, S. D. Rio, J. M. Benitez, and F. Herrera, “Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data,” *Fuzzy Sets Syst.*, vol. 258, pp. 5-38, Jan., 2015.

[27] A. Lorena, A. Carvalho, and J. Gama, “A review on the combination of binary classifiers in multiclass problems,” *Artif. Intell. Rev.*, vol. 30, no. 1-4, pp. 19-37, Dec., 2008.

[28] M. I. Rey, M. Galende, M. J. Fuente, and G. I. Sainz-Palmero, “Multi-objective based Fuzzy Rule Based Systems (FBRSs) for trade-off improvement in accuracy and interpretability: a rule relevance point of view,” *Knowl. Based Syst.*, vol. 127, pp. 67-84, Jul., 2017.

[29] S. D. Rio, V. Lopez, J. M. Benitez, and F. Herrera, “A MapReduce Approach to Address Big Data Classification Problems Based on the Fusion of Linguistic Fuzzy Rules,” *Int. J. Comput. Intell. Syst.*, vol. 8, no. 3, pp. 422-437, Feb., 2015.

[30] UCI repository of machine learning database, Sep. 14, 2018. [Online]. Available: http://archive.ics.uci.edu/ml/index.php.

[31] L. X. Wang and J.M. Mendel, “Generating fuzzy rules by learning from examples,” *IEEE Trans. Syst., Man, and Cybern.*, vol. 22, no. 6, pp. 1414–1427, Nov./Dec., 1992.

[32] Y. M. Wang, J. B. Yang, and D. L. Xu, “Environmental impact assessment using the evidential reasoning approach,” *Eur. J. Oper. Res.*, vol. 174, no. 3, pp. 1885-1913, Nov., 2006.

[33] X. J. Xu, C. X. Sheng, C. Q. Yuan, D. L. Xu, and J. B. Yang, “A belief rule-based expert system for fault diagnosis of marine diesel engines,” *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published. DOI: 10.1109/TSMC.2017.2759026.

[34] C. X. Yan, M. N. Luo, H. Liu, Z. H. Li, and Q. H. Zheng, “Top-k multi-class SVM using multiple features,” *Inf. Sci.*, vol. 432, pp. 479-494, Mar., 2018.

[35] J. B. Yang, “Rule and utility based evidential reasoning approach for multiattribute decision analysis under uncertainties,” *Eur. J. Oper. Res.*, vol. 131, no. 1, pp. 31-61, May, 2001.

[36] J. B. Yang, J. Liu, D. L. Xu, J. Wang, and H. W. Wang, “Optimization models for training belief-rule-based systems,” *IEEE Trans. Syst., Man, Cybern., A, Syst. Humans*, vol. 37, no. 4, pp. 569-585, Jul., 2007.

[37] J. B. Yang, J. Liu, J. Wang, H. S. Sii, and H. W. Wang, “Belief rule-base

inference methodology using the evidential reasoning approach-RIMER," *IEEE Trans. Syst., Man, Cybern., A, Syst. Humans*, vol. 36, no. 2, pp. 266-285, Mar., 2006.

[38] L. H. Yang, Y. M. Wang, Y. X. Lan, L. Chen, and Y. G. Fu, "A data envelopment analysis (DEA)-based method for rule reduction in extended belief-rule-based systems," *Knowl. Based Syst.*, vol. 123, pp. 174-187, May, 2017.

[39] L. H. Yang, Y. M. Wang, Q. Su, Y. G. Fu, and K. S. Chin, "Multi-attribute search framework for optimizing extended belief rule-based systems," *Inf. Sci.*, vol. 370-371, pp. 159-183, Nov., 2016.

[40] Z. J. Zhou, G. Y. Hu, B. C. Zhang, C. H. Hu, Z. G. Zhou, and P. L. Qiao, "A model for hidden behavior prediction of complex systems based on belief rule base and power set," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 99, pp. 1-7, Sep., 2017.

Long-Hao Yang received the B.Eng. and M.Eng. degrees from Fuzhou University, China, in 2012 and 2015, respectively, and he is currently pursuing the Ph.D. degree at Fuzhou University, China.

He has published over 10 articles in BRB-related studies. His current research interests include BRB learning, machine learning and artificial intelligence, and BRB applications in complex systems modeling.

Jun Liu (M'09) received the BSc and MSc degrees in applied mathematics, and the PhD degree in information engineering from Southwest Jiaotong University, Chengdu, China, in 1993, 1996, and 1999, respectively.

He is currently a Reader in Computer Science at Ulster University, Northern Ireland, UK. His current research interests include logic and reasoning methods for intelligent systems and formal verification; intelligent DSSs and information management, with applications in health care, engineering, and industry field, etc.; applied computational intelligence for uncertainty analysis and optimisation. He has published more than 80 papers in journals indexed by the SCI and more than 80 contributions in International Conferences.

Ying-Ming Wang received his MSc in Systems Engineering from Huazhong University of Science and Technology, China, in 1987, and Ph.D. in Automatic Control Theory and Application from Southeast University, China, in 1991.

He is currently a full distinguished professor of Changjiang Scholars Program at Fuzhou University, Fuzhou, China. He has published over 123 SCI and 28 SSCI-indexed journal papers and has been the most cited Chinese researchers since 2014. His research interests include multiple criteria decision analysis, data envelopment analysis, rule-based inference, and quality function deployment.

Luis Martínez (M'10) received the M.Sc. and Ph.D. degrees in computer sciences from the University of Granada, Granada, Spain, in 1993 and 1999, respectively.

He is currently a Full Professor with the Computer Science Department, University of Jaén, Jaén, Spain. He is also Visiting Professor in University of Technology Sydney, University of Portsmouth (Isambard Kingdom Brunel Fellowship Scheme), and in the Wuhan University of Technology (Chutian Scholar), Guest Professor in the Southwest Jiaotong University and Honourable professor in Xihua University both in Chengdu (China). He has co-edited eleven journal special issues on fuzzy preference modelling, soft computing, linguistic decision making and fuzzy sets theory and has been main researcher in 14 R&D projects, also has published more than 100 papers in journals indexed by the SCI and more than 150 contributions in International Conferences related to his areas. His current research interests include decision making, fuzzy logic-based systems, computing with words and recommender systems. He was a recipient of the IEEE Transactions on fuzzy systems Outstanding Paper Award 2008 and 2012 (bestowed in 2011 and 2015, respectively). He is a Co-Editor-in-Chief of the International Journal of Computational Intelligence Systems and an Associate Editor of the journals, including the IEEE Transactions on fuzzy systems, Information Fusion, the International Journal of Fuzzy Systems, the Journal of Intelligent and Fuzzy Systems. He is a member of IEEE and of the European Society for Fuzzy Logic and Technology. Eventually, he has been appointed as *Highly Cited Researcher* 2017 in Computer Science