

# Fuzzy integral-based ELM ensemble for imbalanced big data classification

Junhai Zhai<sup>1</sup> · Sufang Zhang<sup>2</sup> · Mingyang Zhang<sup>1</sup> · Xiaomeng Liu<sup>1</sup>

Published online: 19 February 2018  
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

## Abstract

Big data are data too big to be handled and analyzed by traditional software tools, big data can be characterized by five V's features: volume, velocity, variety, value and veracity. However, in the real world, some big data have another feature, i.e., class imbalanced, such as e-health big data, credit card fraud detection big data and extreme weather forecast big data are all class imbalanced. In order to deal with the problem of classifying binary imbalanced big data, based on MapReduce, non-iterative learning, ensemble learning and oversampling, this paper proposed an promising algorithm which includes three stages. Firstly, for each positive instance, its enemy nearest neighbor is found with MapReduce, and  $p$  positive instances are randomly generated with uniform distribution in its enemy nearest neighbor hypersphere, i.e., oversampling  $p$  positive instances within the hypersphere. Secondly,  $l$  balanced data subsets are constructed and  $l$  classifiers are trained on the constructed data subsets with an non-iterative learning approach. Finally, the trained classifiers are integrated by fuzzy integral to classify unseen instances. We experimentally compared the proposed algorithm with three related algorithms: SMOTE, SMOTE+RF-BigData and MR-V-ELM, and conducted a statistical analysis on the experimental results. The experimental results and the statistical analysis demonstrate that the proposed algorithm outperforms the other three methods.

**Keywords** Imbalanced big data · MapReduce · Non-iterative learning · Oversampling · Fuzzy integral

## 1 Introduction

Binary imbalanced data refer to a type of data, where one class (positive class) is highly under-represented compared to another class (negative class). In the scenario of binary imbalanced classification (He and Garcia 2009; Krawczyk 2016), to which both traditional classification approaches (such as support vector machines, decision trees, neural networks) and

assessment metrics (such as classification accuracy) can't be directly applied. Because there are large requirements for dealing with binary imbalanced data in practice, and many methods have been proposed in the last two decades. These methods can be roughly classified into three categories: data-level methods, algorithm-level methods and hybrid methods (He and Garcia 2009; Krawczyk 2016). For convenience of description, we introduce some notations. Let  $S$  be original set with  $n$  instances, i.e.,  $|S| = n$ ,  $S^+$  and  $S^-$  denote the set of minority class (positive class) instances and majority class (negative class) instances in  $S$ , respectively. Obviously,  $S = S^+ \cup S^-$  and  $|S^+| \ll |S^-|$ .

In the data-level methods, an imbalanced data distribution is modified to a balanced data distribution by random sampling, the random sampling can be categorized into two classes: random undersampling and random oversampling. The random undersampling randomly selects a subset  $S'$  from  $S^-$  and removes  $S'$  from original data set  $S$ . The random undersampling is straightforward, it is usually combined with other approaches for binary imbalanced classification. For example, Liu et al. (2009) combined random undersampling and ensemble method and proposed two ensemble algo-

---

Communicated by X. Wang, A. K. Sangaiah, M. Pelillo.

---

Junhai Zhai and Sufang Zhang have contributed equally to this paper.

---

✉ Junhai Zhai  
mczjh@126.com

✉ Sufang Zhang  
mczsf@126.com

<sup>1</sup> Key Laboratory of Machine Learning and Computational Intelligence, College of Mathematics and Information Science, Hebei University, Baoding 071002, Hebei, China

<sup>2</sup> Hebei Branch of China Meteorological Administration Training Centre, China Meteorological Administration, Baoding 071000, Hebei, China

rithms for class imbalanced learning, the methods in Liu et al. (2009) achieved good classification results. Ofek et al. (2017) combined random undersampling and clustering method and proposed a clustering-based undersampling technique for addressing binary-class imbalance problems, the method in Ofek et al. (2017) demonstrates high predictive performance, while its time complexity is bound by the size of the minority class instances. Bao et al. (2016) combined random undersampling and ensemble of support vector machine and proposed a learning approach for concept detection in large-scale imbalanced data sets. The training data set for each base support vector machine is constructed by a boosted near-miss undersampling technique. The random oversampling increases the size of  $S^+$  by generating new positive instances. The most striking generation-based random oversampling method is the so-called Synthetic Minority Oversampling TEchnique (SMOTE) (Chawla et al. 2002), which generates artificial instances based on  $k$ -nearest neighbors of positive instances. Inspired by the idea of SMOTE, many promising synthetic sampling approaches have been proposed in recent years. For instance, Das et al. (2015) proposed two probabilistic oversampling methods named RACOG and wRACOG, which considered the probability distribution of the minority class in the process of synthetically generating new samples. Rivera (2017) proposed a noise reduction synthetic oversampling algorithm which results in improvement for prediction of minority class. Based on Mahalanobis distance, in the framework of multi-class imbalanced data classification, Abdi and Hashemi (2016) proposed a new oversampling technique which generates synthetic instances which have the same Mahalanobis distance from the considered class mean as other minority class instances. In order to increase classification sensitivity in imbalanced data sets, Rivera and Xanthopoulos (2016) proposed a priori synthetic oversampling methods.

In the algorithm-level methods, which directly modify existing classification algorithms to adapt to imbalanced classification scenarios. This kind of method mainly includes cost-sensitive methods and instance weight methods. Cost-sensitive methods take into account the variable costs of misclassifications of instances belonging to different classes, in the scenario of binary imbalanced classification, the costs can be modeled by a cost matrix whose element  $c(i, j)$  is the cost of predicting an instance belonging to class  $i$  when in fact it belongs to class  $j$ . The pioneering works of cost-sensitive methods are presented by Sun and her co-researchers (2007). They proposed three cost-sensitive boosting algorithms in Sun et al. (2007), i.e., the well-known AdaC1, AdaC2 and AdaC3, which are developed by introducing cost items into the learning framework of AdaBoost. Based on cost matrix, Krawczyk et al. (2014) proposed an approach of ensemble of cost-sensitive decision tree for imbalanced classification, the cost-sensitive decision trees are constructed according to

a given cost matrix, and are trained on random feature subspaces to ensure sufficient diversity of the base classifiers. In order to overcome the difficulty of determining the precise misclassification costs in cost-sensitive methods, Cao et al. (2013) proposed an optimized cost-sensitive support vector machine (SVM) for imbalanced classification. The proposed method in Cao et al. (2013) incorporates the evaluation measure (AUC and  $G$ -mean) into the objective function of cost-sensitive SVM to improve the performance of binary imbalanced classification. In order to deal with problem of imbalanced big data classification, based on the MapReduce, López et al. (2015) proposed a cost-sensitive fuzzy classification method, which uses the MapReduce framework to distribute the computational operations of the cost-sensitive learning fuzzy model into different cloud computing nodes. Castro and Braga (2013) proposed a novel cost-sensitive approach to improve the performance of MLPs by optimizing a joint cost function which corresponds to the sum of squared errors for positive class and negative class. In the instance weight methods, based on fuzzy rough set theory (Zhao et al. 2015, 2013; Zhai et al. 2017; El-Monsef et al. 2017; Tsang et al. 2017), Ramentol et al. (2015) proposed a classification algorithm for imbalanced data, the proposed algorithm uses fuzzy rough set and aggregation approach to build a weight vector for weighting samples. Inspired by a simple notion that a trained support vector machine provides information regarding bounded and unbounded support vectors, Lee et al. (2017) proposed a method for imbalanced data classification, which introduces a new weight adjustment factor to the proposed algorithm. Zong et al. (2013) proposed an instance weight approach named weighted extreme learning machine (W-ELM). The W-ELM has two merits: (1) the weighted ELM is able to deal with data with imbalanced class distribution while maintain the good performance on well balanced data as unweighted ELM; (2) by assigning different weights for each example according to users' needs, the weighted ELM can be generalized to cost-sensitive learning.

In the hybrid methods, which generally combine the advantages of the two kinds of methods mentioned above by ensemble learning, including cost-sensitive ensemble methods and data preprocessing ensemble methods. The cost-sensitive ensembles keep the general learning framework of AdaBoost, but at the same time introduce cost items into the weight update formula, the predominant cost-sensitive ensembles include AdaCost (Fan et al. 1999), RareBoost (Joshi et al. 2001), and AdaCx ( $x = 1, 2, 3$ ) (Sun et al. 2007). In the data preprocessing ensemble methods, each base classifier is trained with a pre-processed data set, and the data preprocessing techniques are embedded into boosting algorithms. The representative works include SMOTEBoost (Chawla et al. 2003), RUSBoost (Seiffert et al. 2010), and EUSBoost (Galar et al. 2013) etc. Galar et al.

(2012) present an excellent review on the ensembles for the class imbalance problem.

Although many works have addressed the problem of imbalanced data classification, limited studies have been devoted to classifying imbalanced big data, the pioneering works investigated the problem of imbalanced big data classification are reported in literature Ghanavati et al. (2014) and Ríó et al. (2014). By combining metric learning and balancing techniques, Ghanavati et al. (2014) proposed a integrated method to learn from imbalanced big data. Ríó et al. (2014) discussed the extensions of oversampling, undersampling and cost-sensitive learning in imbalanced big data scenario, and gave their implementation frameworks with MapReduce. Aiming at classifying extremely imbalanced big data, Triguero et al. (2016) proposed an approach of evolutionary undersampling and implement the proposed algorithm with Apache Spark. Under the sparsity assumption, Maurya et al. (2016) discussed the problem of online sparse class imbalance learning on big data, and presented an effective solution for the problem. Recently, Fernández et al. (2017) provided an insight into the problem of imbalanced big data classification with researchers including some outcomes and challenges.

Regarding the training approaches for basic classifiers of neural networks in ensemble learning, compared with iterative approaches, such as gradient-based learning approaches, non-iterative approaches are much more efficient and effective, because that non-iterative approaches have fast learning speed and good generalization performance. In the past decade, non-iterative approaches have attracted wide attention of the researchers. Non-iterative approaches can be classified into two categories: deterministic non-iterative approaches and random non-iterative approaches. In the first category, the first deterministic non-iterative method was proposed by Schmidt et al. (1991) for training feed-forward networks, which was motivated by the computation of concave piecewise linear discriminant functions. Based on the memory desaturation technique, Reznik (1999) proposed a non-iterative method for training Hopfield neural networks. Recently, Oscar et al. (2017) proposed an incremental non-iterative method for training one-layer feed-forward neural networks without hidden layers. As pointed out by Oscar et al. (2017), the deterministic non-iterative approaches are quite scarce, actually only a few salient methods mentioned above are proposed. On the contrary, the random non-iterative approaches are very plentiful, many researchers (Schmidt et al. 1992; Pao et al. 1992, 1994; Igel'nik and Pao 1995) have contributed their efforts to this field. Recently, Wang et al. (2017) introduced non-iterative approach into deep learning, and proposed a new training algorithm for training deep neural networks, Cao et al. (2018) presented a comprehensive survey on neural networks with random weights. Extreme learning machines (ELMs) (Huang et al. 2006, 2011, 2015)

were initially proposed by Huang and his co-researchers for training SLFNs. Huang (2015) presented the relationships and the differences between the ELMs and some earlier similar works by clarifying the essential elements of ELMs. The key superiority of ELM is that it has fast learning speed and good generalization capacity (Huang et al. 2006). With respect to generalization capacity of classifiers, Wang et al. (2017, 2015) and Wang and Dong (2009) studied the relationship between generalization capacity and uncertainties of classifiers, and proposed corresponding technique to enhance the generalization capacity of classifiers. Recently, on the study of the relationship between generalization and complexity of classification problem, Wang et al. (2018) obtained an important conclusion that the generalization ability of a classifier is statistically becoming better with the increase in uncertainty when the complexity of the classification problem is relatively high, and the generalization ability is statistically becoming worse with the increase in uncertainty when the complexity is relatively low. Furthermore, they applied this conclusion to multiple-instance active learning to enhance the generalization ability of the learning system (Wang et al. 2017). ELM not only can be used to solve many classical problems, such as function approximation (Huang et al. 2012), decision making (Ren and Wei 2017; Ye 2017; Mao et al. 2017), data reduction (Meng et al. 2017; Cai et al. 2017; Li et al. 2017; Ding et al. 2017) and so on, but also can be used to solve some new problems, for example, Zhang and Zhang (2016); Zhang et al. (2016) and Zhang and Zhang (2015) addressed the problem of visual knowledge adaptation by ELM, and proposed ELM-based cross-domain network learning framework called ELM-based domain adaptation (EDA), extensive experiments on benchmark visual data sets demonstrate that the EDA outperforms the existing cross-domain learning methods. The latest advances in extreme learning machine can be found in Huang et al. (2011).

In general framework of classifier ensemble or classifier fusion, there is an acquiescent assumption that the basic classifiers used for fusion are independent or noninteractive (Ralescu and Adams 1980; Abdallah et al. 2012; Kuncheva 2001; Zhan et al. 2012). However, this assumption is not realistic in many applications. Actually, there are some inherent interactions among different basic classifiers, the interactions may be positive synergy, and in this case the basic classifiers cooperate and enhance each other. The ensemble system can take advantages of the strengths of the individual classifiers, and can overcome their weaknesses, finally achieve a higher accuracy than any individuals. The interaction also may be negative synergy, i.e., the corresponding basic classifiers restrain each other. Fuzzy integral (Ralescu and Adams 1980) is distinguished from other ensemble methods by this desirable property. Furthermore, fuzzy integral can also characterize the significance of basic classifiers by fuzzy measure. Motivated by these merits of fuzzy integral, in this paper we

proposed an algorithm for imbalanced big data classification, which combines sampling technique, non-iterative learning and fuzzy integral-based ensemble learning, and we implement the proposed algorithm with MapReduce. The paper is structured as follows: the preliminaries used in this paper are briefly reviewed in Sect. 2. The proposed algorithm is given in Sect. 3. The experimental results and analysis are presented in Sect. 4 and Sect. 5 concludes this paper.

## 2 Preliminaries

In the proposed approach, the base classifiers used in ensemble for imbalanced big data classification are SLFNs which are trained with an efficient non-iterative learning approach ELM. The reason why we select ELM is that it is effective and efficient, in other words, it has fast learning speed with very good generalization capacity. The selected ensemble method is fuzzy integral, we employ fuzzy integral as ensemble tool because that it can well model the interactions among the base classifiers. In the following, we will briefly review the ELM and fuzzy integral.

### 2.1 Extreme learning machine (ELM)

ELM is an efficient random non-iterative algorithm for training SLFNs, it randomly assigns the weights between input layer and hidden layer, and analytically determined the weights between hidden layer and output layer. Given a training set,  $D = \{(x_i, y_i) | x_i \in R^d, y_i \in R^k, i = 1, 2, \dots, n\}$ , where  $x_i$  is a  $d \times 1$  input vector and  $y_i$  is a  $k \times 1$  target vector, the SLFNs with  $m$  hidden nodes can be modeled by

$$f(x_i) = \sum_{j=1}^m \beta_j g(w_j \cdot x_i + b_j), \quad i = 1, 2, \dots, n \quad (1)$$

where  $w_j = (w_{j1}, w_{j2}, \dots, w_{jd})^T$  is the weight vector connecting the  $j$ th hidden node with the input nodes,  $b_j$  is the bias of the  $j$ th hidden node. The  $w_j$  and  $b_j$  are randomly assigned.  $\beta_j = (\beta_{j1}, \beta_{j2}, \dots, \beta_{jm})^T$  is the weight vector connecting the  $j$ th hidden node with the output nodes. The parameters  $\beta_j (j = 1, 2, \dots, m)$  may be estimated by least square fitting with the given training set  $D$ , i.e., we have

$$f(x_i) = \sum_{j=1}^m \beta_j g(w_j \cdot x_i + b_j) = y_i \quad (2)$$

Equation (2) can be written in a more compact format as

$$H\beta = Y \quad (3)$$

where

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_m \cdot x_1 + b_m) \\ \vdots & \dots & \vdots \\ g(w_1 \cdot x_n + b_1) & \dots & g(w_m \cdot x_n + b_m) \end{bmatrix} \quad (4)$$

$$\beta = (\beta_1^T, \dots, \beta_m^T)^T \quad (5)$$

and

$$Y = (y_1^T, \dots, y_n^T)^T \quad (6)$$

$H$  is the hidden layer output matrix of SLFN, where the  $j$ th column of  $H$  is the  $j$ th hidden nodes output vector with respect to inputs  $x_1, x_2, \dots, x_n$ , and the  $i$ th row of  $H$  is the output vector of the hidden layer with respect to input  $x_i$ . If the number of hidden nodes is equal to the number of distinct training samples, the matrix  $H$  is square and invertible, and SLFN can approximate these training samples with zero error. But generally, the number of hidden nodes is much less than the number of training samples. Therefore,  $H$  is a non-square matrix and one can't expect to find an exact solution of the system (3). Fortunately, we can find its smallest norm least square solution by solving the following optimization problem (Cao et al. 2018; Huang et al. 2012).

$$\min_{\beta} \|H\beta - Y\| \quad (7)$$

The smallest norm least-squares solution of (7) can be easily obtained by

$$\hat{\beta} = H^\dagger Y \quad (8)$$

where  $H^\dagger = (HH^T)^{-1}H$  is the Moore–Penrose generalized inverse of matrix  $H$ . The ELM can be described by the Algorithm 1 (Cao et al. 2018).

---

#### Algorithm 1: ELM Algorithm

---

**Input:** Training data set

$D = \{(x_i, y_i) | x_i \in R^d, y_i \in R^k, i = 1, 2, \dots, n\}$ , an activation function  $g$ , and the number of hidden nodes  $m$

**Output:** weights matrix  $\beta$ .

1 **for** ( $j = 1; j \leq m; j = j + 1$ ) **do**

2 | Randomly assign input weights  $w_j$  and  $b_j$ ;

3 **end**

4 Calculate the hidden layer output matrix  $H$ ;

5 Calculate output weights matrix  $\beta = H^\dagger Y$ .

---

In order to make the outputs of the trained SLFN by ELM to be a approximated posterior probability distribution, we can transform the outputs of SLFN by the following softmax function.

$$p(w_j|x) = \frac{e^{y_j}}{\sum_{i=1}^k e^{y_i}} \tag{9}$$

where  $x$  is a testing instance,  $p(w_j|x)$  is the approximation of the posterior probability that instance  $x$  belongs to class  $j$ .

### 2.2 Fuzzy integral

In this section, we present the notations related to fuzzy integral (Ralescu and Adams 1980). Let  $D = \{(x_i, y_i)|x_i \in R^d, y_i \in R^k, i = 1, 2, \dots, n\}$  be a training set,  $Y = \{\omega_1, \omega_2, \dots, \omega_k\}$  be a set of class labels,  $L = \{L_1, L_2, \dots, L_l\}$  be a set of classifiers trained on  $D$  or on subsets of  $D$ . In the context of ensemble learning,  $L_i(1 \leq i \leq l)$  are called base classifiers or component classifiers. For  $\forall x \in R^d$ ,  $L_i$  assigns a class label to  $x$  from  $Y$ . As given by Kuncheva Kuncheva (2001), we may define the output of classifier  $L_i$  to be a  $k$ -dimensional vector:

$$L_i(x) = (p_{i1}(x), p_{i2}(x), \dots, p_{ik}(x)) \tag{10}$$

where  $p_{ij}(x) \in [0, 1](1 \leq i \leq l; 1 \leq j \leq k)$  denotes the support degree given by classifier  $L_i$  to the hypothesis that  $x$  comes from class  $\omega_j$ ,  $\sum_{j=1}^k p_{ij}(x) = 1$ . In this paper,  $p_{ij}(x)$  is an estimation of the posterior probability  $p(\omega_j|x)$  by classifier  $L_i$ .

Given  $L = \{L_1, L_2, \dots, L_l\}$ ,  $Y = \{\omega_1, \omega_2, \dots, \omega_k\}$ , and arbitrary testing sample  $x$ . The following matrix is called decision matrix with respect to  $x$ .

$$DM(x) = \begin{bmatrix} p_{11}(x) & \dots & p_{1j}(x) & \dots & p_{1k}(x) \\ \vdots & & \vdots & & \vdots \\ p_{i1}(x) & \dots & p_{ij}(x) & \dots & p_{ik}(x) \\ \vdots & & \vdots & & \vdots \\ p_{l1}(x) & \dots & p_{lj}(x) & \dots & p_{lk}(x) \end{bmatrix} \tag{11}$$

where the  $i$ th row of the matrix is the output of classifier  $L_i$ , the  $j$ th column of the matrix is the support degree from classifiers  $L_1, L_2, \dots, L_l$  for class  $\omega_j$ .

Let  $P(L)$  be the power set of  $L$ , the fuzzy measure on  $L$  is a set function:  $g : P(L) \rightarrow [0, 1]$ , which satisfies the following two conditions:

- (1)  $g(\emptyset) = 1, g(L) = 1;$
- (2) For  $\forall A, B \subseteq L$ , if  $A \subset B$ , then  $g(A) \leq g(B)$ .

For  $\forall A, B \subseteq L$  and  $A \cap B = \emptyset$ ,  $g$  is called  $\lambda$ -fuzzy measure, if it satisfies the following condition:

$$g(A \cup B) = g(A) + g(B) + \lambda g(A)g(B) \tag{12}$$

where  $\lambda > -1$  and  $\lambda \neq 0$ .

The value of  $\lambda$  can be determined by solving the following Eq. (12).

$$\lambda + 1 = \prod_{i=1}^l (1 + \lambda g_i) \tag{13}$$

where  $g_i = g(\{L_i\})$ , which is called fuzzy density of classifier  $L_i$ . It is noted that Eq. (13) has only one solution which meets the conditions  $\lambda > -1$  and  $\lambda \neq 0$ . Usually,  $g_i$  can be determined by the following formula (Zhan et al. 2012):

$$g_i = \frac{p_i}{\sum_{j=1}^l p_j} \delta. \tag{14}$$

where  $\delta \in [0, 1]$  and  $p_i$  is testing accuracy or verification accuracy of classifier  $L_i(1 \leq i \leq l)$ .

Let  $h : L \rightarrow [0, 1]$  be a function defined on  $L$ . The Choquet fuzzy integral and the Sugeno fuzzy integral of function  $h$  with respect to  $g$  are defined by (15a) and (15b), respectively.

$$(C) \int h d\mu = \sum_{i=2}^{l+1} (h(L_{i-1}) - h(L_i)) g(A_{i-1}), \tag{15a}$$

$$(S) \int h d\mu = \bigvee_{i=1}^l (h(L_i) \wedge g(A_i)). \tag{15b}$$

where  $h(L_1) \geq h(L_2) \geq \dots \geq h(L_l), h(L_{l+1}) = 0, A_{i-1} = \{L_1, L_2, \dots, L_{i-1}\}$ .

Given a testing instance  $x$ , when we use fuzzy integral to integrate the  $l$  trained base classifiers  $L_1, L_2, \dots, L_l$  for classifying  $x$ , we first compute decision matrix  $DM(x)$ , and then sort  $j$ th ( $1 \leq j \leq k$ ) column of  $DM(x)$  in descending order and obtain  $(p_{i1j}, p_{i2j}, \dots, p_{ilj})$ .

If the trained base classifiers  $L_1, L_2, \dots, L_l$  are integrated by the Choquet fuzzy integral with formula (15a) for classifying  $x$ , then the support degree  $p_j(x)$  is calculated by the following formula (16).

$$p_j(x) = \sum_{t=2}^{l+1} (p_{i_{t-1}j}(x) - p_{ij}(x)) g(A_{t-1}) \tag{16}$$

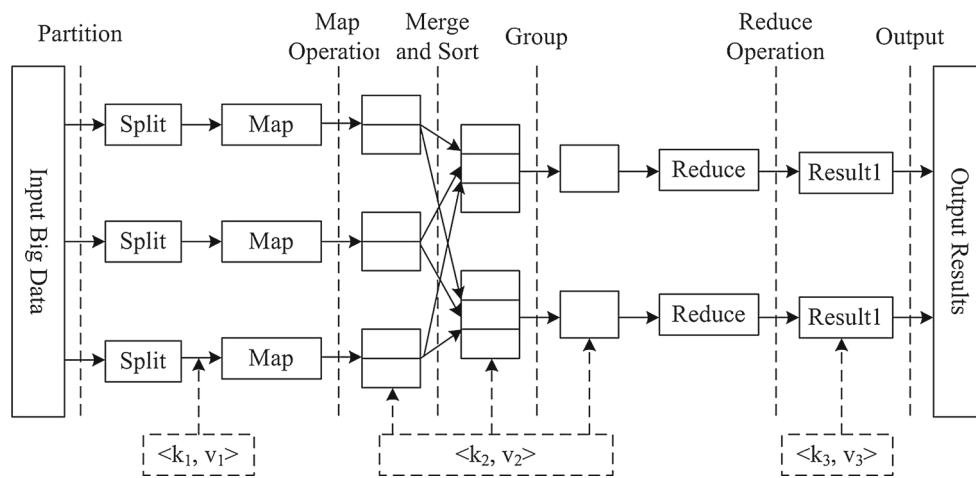
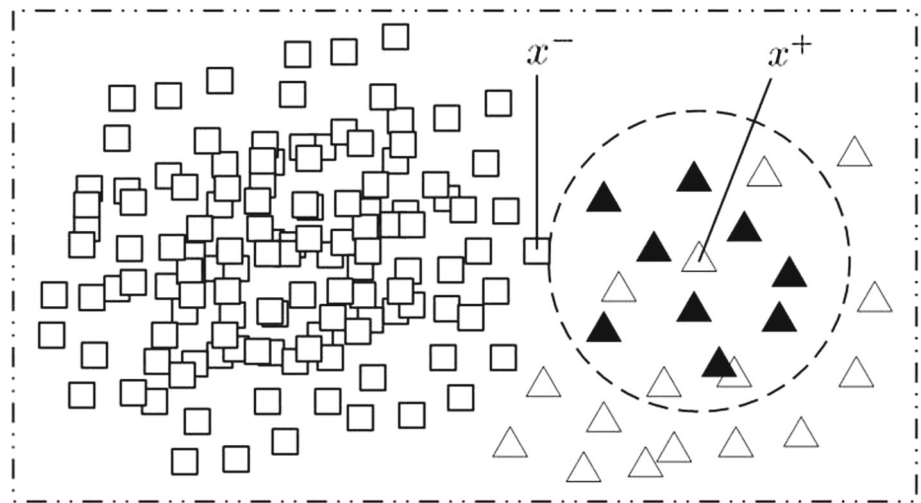
If the trained base classifiers  $L_1, L_2, \dots, L_l$  are integrated by the Sugeno fuzzy integral with formula (15b) for classifying  $x$ , then the support degree  $p_j(x)$  is calculated by the following formula (17).

$$p_j(x) = \max_{1 \leq t \leq l} \{ \min\{p_{ij}, g(A_t)\} \} \tag{17}$$

In this paper, we use the Choquet fuzzy integral to integrate the trained base classifiers.



**Fig. 1** The idea of randomly oversampling approach with  $p = 9$



**Fig. 2** Data manipulation processes of MapReduce

### 3 The proposed algorithm

In this section, we introduce the proposed algorithm which includes three stages: (1) randomly oversampling referred to enemy nearest neighbor which is found by a parallel framework with MarReduce. (2) Construction of balanced training sets and training base classifiers SLFNs with ELM. (3) Ensemble the trained SLFNs with fuzzy integral.

Let  $S = S^+ \cup S^-$  be imbalanced big data,  $|S^+| = n^+$ ,  $|S^-| = n^-$ ,  $n^+ \ll n^-$ . The  $S^+$  and  $S^-$  denote the set of positive class instances and negative class instances in  $S$ , respectively. The idea of our proposed randomly oversampling approach can be depicted by Fig. 1, where the symbol “□” denotes a negative instance, “△” denotes a positive instance, while “▲” denotes a positive instance randomly generated with uniform distribution within the nearest circle of  $x^+$  which is a positive instance,  $x^-$  is its enemy nearest neighbor. For each  $x^+ \in S^+$ , we randomly generate  $p$  instances in its corresponding nearest neighbor circle (or

hypersphere in high dimensional space) with uniform distribution,  $p$  is a parameter defined by user.

Because that  $n^+ \ll n^-$ ,  $S$  is a big data set indicates that  $S^-$  is a big data set. For  $\forall x^+ \in S^+$ , in order to efficiently find its enemy nearest neighbor from big data set  $S^-$ , MapReduce (Dean and Ghemawat 2008; Zhai et al. 2016, 2017; Ludwig 2015) is employed in the proposed algorithm. MapReduce is a parallel programming framework which uses the strategy of divide and conquer to deal with big data (Emani et al. 2015; Wang 2015). It automatically partitions big data into some small pieces and deploys them into different parallel computing nodes (see Fig. 2). Based on the idea of functional programming language LISP (Berkeley and Bobrow 1964), MapReduce provides a simple and feasible method of parallel programming, two functions Map and Reduce are used to fulfill the basic parallel computing tasks. Map and Reduce are two abstract interfaces of parallel programming. For a given specific big data problem, one only needs to implement the two functions, many programming details in system bottom

level have been abstracted away for users. In the proposed algorithm, the designed Map and Reduce functions are presented in Algorithms 2 and 3, respectively.

---

**Algorithm 2:** Map function

---

```

Input: <  $k_1, v_1$  >
Output: <  $k_2, v_2$  >
1 // Finding the enemy nearest neighbor of  $x_i$ ;
2 for ( $i = 1; i < PositiveSet.size(); i = i + 1$ ) do
3   for ( $j = 1; j < NegativeSet.size(); j = j + 1$ ) do
4     distance=EuclideanDistance(PositiveSet.size(i),
5     NegativeSet.size(j));
6     if ( $distance \leq minDistance$ ) then
7       | minDistance=distance;
8     end
9   end
10 // Oversampling within nearest neighbor
    hypersphere of  $x_i$ ;
11 OverSampleSet.add(newSample);
12 // Output oversampled positive instances;
13 for ( $i = 1; i < OverSampleSet.size(); i = i + 1$ ) do
14 | context.write(OverSampleSet.get(i),NullWritable.get());
15 end
16 Output <  $k_2, v_2$  >
    
```

---



---

**Algorithm 3:** Reduce function

---

```

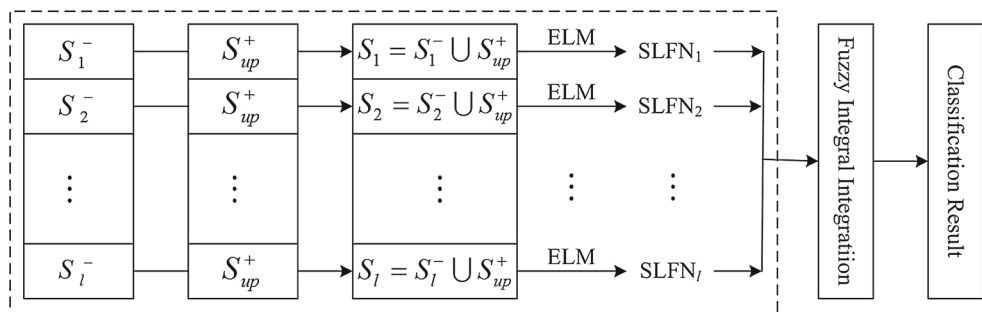
Input: <  $k_2, v_2$  >
Output: <  $k_3, v_3$  >
1 for (NullWritable v2 : v2s) do
2 | context.write(OverSample,NullWritable.get());
3 end
4 Output <  $k_3, v_3$  >
    
```

---

After randomly oversampling, we obtain a new positive instance subset  $S_{up}^+$  whose size is  $q = (1 + p)n^+$ . Next we construct  $l$  balanced training sets and train  $l$  base classifiers SLFNs on the  $l$  balanced training sets with ELM, this process can be depicted by Fig. 3 (the part in the dashed rectangle), where  $S^- = S_1^- \cup S_2^- \cup \dots \cup S_l^-$  and  $l$  is determined by the following formula (18).

$$l = \frac{n^-}{q} = \frac{n^-}{(1 + p)n^+} \tag{18}$$

**Fig. 3** The process of constructing  $l$  balanced training sets and training  $l$  base classifiers SLFNs (the part in the dashed rectangle)



The  $l$  is the number of balanced training sets, it is also the number of base classifiers of an ensemble. Given an imbalanced big data set, because that  $n^-$  is a constant, we can find from formula (18) that  $l$  is actually determined by  $q$ , while  $q$  is determined by  $p$  which is the number of randomly generated positive instances with uniform distribution in a nearest neighbor hypersphere.

The final stage of the proposed algorithm is to integrate the trained base classifiers SLFNs with fuzzy integral. In the proposed algorithm, we use the Choquet fuzzy integral to integrate the trained SLFNs whose results are transformed into the interval  $[0, 1]$  by softmax function (9). The pseudo code of the proposed algorithm is given in Algorithm 4.

### 4 Experimental results and analysis

For convenience, the proposed algorithm is denoted by MR-FI-ELM. If the ELM algorithm is replaced with BP algorithm to train the basic classifiers in the proposed algorithm MR-FI-ELM, then the corresponding algorithm is denoted by MR-FI-BP. In order to verify the effectiveness of MR-FI-ELM, we conduct two experiments on 6 data sets and present a statistical analysis on the experimental results. The first experiment is to compare the MR-FI-ELM with three related algorithms: SMOTE (Chawla et al. 2002), SMOTE-RF-BigData (Río et al. 2014) and MR-V-ELM (Zhai et al. 2017) on two measures:  $F$ -measure and  $G$ -mean. The second experiment is to compare MR-FI-ELM with MR-FI-BP on running time. All experiments are carried out on 6 data sets which include 2 artificial imbalanced big data sets, 3 UCI imbalanced big data sets (Lichman 2013) and 1 real world data set (Liu and Liu 2016). The basic information of the 6 data sets is listed in Table 1. In Table 1, the imbalanced ratio is the ratio between the number of positive class instances and the number of negative class instances.

The first artificial imbalanced big data set denoted by Artificial-1 which is generated with the following formula (19). If  $f(x_1, x_2, \dots, x_{10}) > 1$ , then the corresponding instances are belonged to negative class, otherwise the corresponding instances are belonged to positive class. The

**Algorithm 4:** The ensemble algorithm based on fuzzy integral for binary imbalanced big data classification

```

Input:  $D = S^+ \cup S^-$ , the binary imbalanced big data;  $T$ , testing set;  $l$ , the number of base classifiers;  $p$ , the number of oversampling positive instances.
Output:  $j^*$ , the class label of  $x \in T$ .
1 // The first stage: oversampling with MapReduce;
2 for ( $i = 1; i \leq n^+; i = i + 1$ ) do
3   For  $x_i^+ \in S^+$ , find its enemy nearest neighbor  $x_i^-$  from  $S^-$  with MapReduce;
4   Randomly generate  $p$  positive instances with uniform distribution in the nearest neighbor hypersphere of  $x_i^+$ ;
5 end
6 // The second stage: Constructing balanced training and training base classifiers;
7 Partition  $S^-$  into  $l$  subsets  $S_1^-, S_2^-, \dots, S_l^-$ ;
8 for ( $i = 1; i \leq l; i = i + 1$ ) do
9   Construct  $l$  balanced training sets  $S_i = S_i^- \cup S_{up}^+$ ;
10  Train base classifier SLFN $_i$  with ELM, and soft-maximize its outputs, obtain a probability distribution  $(p_{i1}(x), p_{i2}(x), \dots, p_{ik}(x))$ ;
11 end
12 // The third stage: integrating the trained base classifiers by fuzzy integral;
13 Calculate fuzzy densities  $g_i (1 \leq i \leq l)$  by (14);
14 Calculate parameter  $\lambda$  by (13);
15 for ( $\forall x \in T$ ) do
16   Calculate  $DM(x)$ ;
17   for ( $j = 1; j \leq k; j = j + 1$ ) do
18     Sort  $j^{th}$  column of  $DM(x)$  in descending order and obtain  $(d_{i1j}, d_{i2j}, \dots, d_{ijj})$ ;
19     Set  $g(A_1) = g_{i1}$ ;
20     for ( $t = 2; t \leq l; t = t + 1$ ) do
21       Calculate  $g(A_t) = g_{it} + g(A_{t-1}) + \lambda g_{it} g(A_{t-1})$ ;
22     end
23     // Calculate  $p_j(x)$  by (9),  $d_{i_{i+1}j}(x) = 0$ ;
24     Calculate  $p_j(x) = \sum_{t=2}^{l+1} [d_{i_{t-1}j}(x) - d_{itj}(x)] g(A_{t-1})$ ;
25   end
26 end
27 Calculate  $p_{j^*}(x) = \operatorname{argmax}_{1 \leq j \leq k} \{p_j(x)\}$ ;
28 Output  $j^*$ .

```

**Table 1** The basic information of the 6 data sets

Data sets	Number of instances	Number of attributes	Imbalance ratio
Artificial-1	241,335	10	0.11 (10:90)
Artificial-2	321,564	2	0.11 (10:90)
Skin_segment	218,704	3	0.11 (10:90)
PokerHand	470,120	10	0.11 (10:90)
CoverType	232,350	54	0.10 (09:91)
Liver	13,000	5	0.04 (04:96)

artificial-1 includes 24,134 positive instances and 217,201 negative instances.

$$f(x_1, x_2, \dots, x_{10}) = \frac{x_1^2}{a_1^2} + \frac{x_2^2}{a_2^2} + \dots + \frac{x_{10}^2}{a_{10}^2} \tag{19}$$

**Table 2** The mean vectors and covariance matrices of two Gaussian distributions

$i$	$\mu_i$	$\Sigma_i$
1	$(1.0, 1.0)^T$	$\begin{pmatrix} 0.6 & -0.2 \\ -0.2 & 0.6 \end{pmatrix}$
2	$(2.5, 2.5)^T$	$\begin{pmatrix} 0.2 & -0.1 \\ -0.1 & 0.2 \end{pmatrix}$

The second artificial imbalanced big data set denoted by Artificial-2 which is generated with two two-dimensional Gaussian distribution  $p(x|\omega_i) \sim N(\mu_i, \Sigma_i)$  ( $i = 1, 2$ ), which are corresponding to two classes, their mean vectors and covariance matrices are given in Table 2. The artificial-2 includes 32,156 positive instances and 318,348 negative instances.

The first UCI data set is transformed from data set Skin\_segment (Lichman 2013), it includes 21,870 positive class instances and 196,834 negative class instances. The second UCI data set is transformed from data set PokerHand (Lichman 2013), it includes 47,012 positive class instances and 423,108 negative class instances. The third UCI data set is transformed from UCI data set CoverType (Lichman 2013), it includes 20,911 positive class instances and 211,439 negative class instances. For convenience, the three transformed UCI data sets are also denoted by Skin\_segment, PokerHand and CoverType.

The real-world data set is a liver data set (Liu and Liu 2016) with two imbalanced classes, which includes 12,500 negative class instances and 500 positive class instances. All instances were collected from a hospital located in Baoding of Hebei Province, we omitted the name of the hospital and removed three attributes name, sex and data from the original liver data set due to protection of privacy of the patients. Among the 6 data sets, Liver has the highest imbalanced ratio 0.04, but it is not a large data set rather than a medium-size data set. Liver data set has 5 attributes, which are alanine aminotransferase (ALT), aspartate aminotransferase (AST), total protein (TP), albumin (ALB) and globulin (GLB). We have three purposes to carry out the experiment on this medium-size data set. The first one is to verify whether the proposed algorithm is feasible on medium-size data set? the second one is to verify whether the proposed algorithm is feasible on extremely imbalanced data sets? the third one is to verify the applicability of the proposed algorithm, i.e., to verify whether the proposed algorithm can deal with practical problems?

**4.1 Experiment 1: compare MR-FI-ELM with SMOTE, SMOTE-RF-BigData and MR-V-ELM on F-measure and G-mean**

The experiments are conducted on a cloud computing platform with 5 nodes. The configuration of the cloud computing



**Table 3** The configuration of the cloud computing platform

Items	Configuration
Operating system	Ubuntu 13.04
Hadoop	Hadoop 0.20.2
JDK	JDK-7u71-linux-i586
Eclipse	Eclipse-java-luna-SR1-linux

**Table 4** The configuration of the nodes of the cloud computing platform

Items	Configuration
CPU	Inter Xeon E5-4603 with two cores, 2.0GZ
Memory	8G
Network card	Broadcom 5720 QP 1GB
Hard disk	1TB

		True class	
		p	n
Hypothesis output	Y	TP (True Positives)	FP (False Positives)
	N	FN (False Negatives)	TN (True Negatives)

**Fig. 4** The confusion matrix

platform is given in Table 3, the configuration of nodes of the cloud computing platform is given in Table 4.

In the scenario of binary imbalanced classification, let  $p$  and  $n$  are the true positive class label and true negative class label, respectively,  $Y$  and  $N$  are the predicted positive class label and predicted negative class labels, respectively. Then, a representation of classification performance can be formulated by a confusion matrix (see Fig. 4).

The commonly used assessment metrics for imbalanced data classification algorithms include Precision, Recall,  $F$ -measure and  $G$ -mean, their definition are given as follows (He and Garcia 2009).

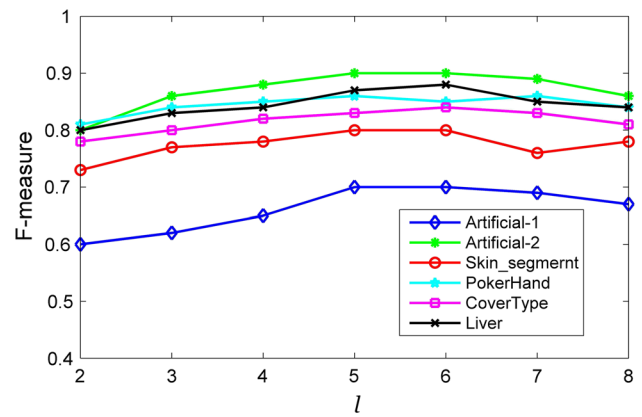
$$Precision = \frac{TP}{TP + FP} \tag{20}$$

$$Recall = \frac{TP}{TP + FN} \tag{21}$$

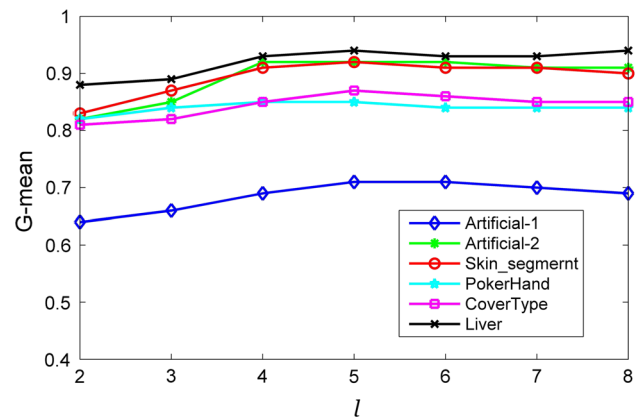
$$F\text{-measure} = \frac{(1 + \beta)^2 \times Recall \times Precision}{\beta^2 \times Recall + Precision} \tag{22}$$

where  $\beta$  is a coefficient to adjust the relative importance of precision versus recall (usually,  $\beta = 1$ ).

$$G\text{-mean} = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} \tag{23}$$



**Fig. 5** The relationship between the parameter  $l$  and  $F$ -measure



**Fig. 6** The relationship between the parameter  $l$  and  $G$ -mean

In this paper, we use  $F$ -measure and  $G$ -mean as the measures to evaluate the performance of the proposed algorithms. From formula (18), we can find that the parameter  $p$ , i.e., the number of oversampling instances, determines the parameter  $l$ , i.e., the number of basic classifiers, while  $l$  has a significant influence on the performance of the integrated classification system. In other words, the parameter  $p$  has a significant influence on the performance of the integrated classification system. We experimentally analyze the relationships between the parameter  $l$  and  $F$ -measure and  $G$ -mean, the results are given in Figs. 5 and 6. From Figs. 5 and 6, we can find that the suitable value of  $l$  should be 5 or 6. In our experiments, we set the parameter  $l = 5$  for the six selected data sets. The experimental results of  $F$ -measure and  $G$ -mean compared with SMOTE, SMOTE+RF-BigData and MR-V-ELM are listed in Tables 5 and 6, respectively. The experimental results listed in Tables 5 and 6 demonstrate that the performance of the proposed algorithm outperforms the three related algorithms. We realize that the reasons why the MR-FI-ELM outperforms the three related algorithms on the performance include the following two points: (1) as a tool for classifier fusion, fuzzy integral can well model the interactions among

**Table 5** The experimental results of  $F$ -measure

Data sets	MR-FI-ELM	SMOTE	SMOTE+RF-BigData	MR-V-ELM
Artificial-1	0.6985	0.6623	0.6744	0.6652
Artificial-2	0.9006	0.8559	0.8709	0.8708
Skin_segment	0.7864	0.6233	0.7510	0.7519
PokerHand	0.8562	0.8211	0.8361	0.8334
CoverType	0.8323	0.8310	0.8248	0.8106
Liver	0.8742	0.8210	0.8419	0.8527

**Table 6** The experimental results of  $G$ -mean

Data sets	MR-FI-ELM	SMOTE	SMOTE+RF-BigData	MR-V-ELM
Artificial-1	0.7062	0.6430	0.6918	0.6814
Artificial-2	0.9210	0.8860	0.8910	0.9097
Skin_segment	0.9201	0.8763	0.9009	0.8566
PokerHand	0.8531	0.7230	0.8291	0.8418
CoverType	0.8650	0.7764	0.8407	0.8315
Liver	0.9421	0.9060	0.9205	0.9206

the basic classifiers which have complementary information. (2) The proposed oversampling method extends the domain of positive instances, which enhances the classification ability of the integrated classification system. In addition, from the experimental results on the data set liver, we can find that (1) the proposed algorithm is not only feasible on imbalanced big data sets, but also feasible on medium-size data sets, (2) the proposed algorithm is also effective on extremely imbalanced data sets. It imply for the experimental results on the data set liver that the proposed algorithm can effectively deal with medium-size real life tasks.

#### 4.2 Experiment 2: compare MR-FI-ELM with MR-FI-BP on running time

The MR-FI-ELM and the MR-FI-BP are similar, the only difference is that the former uses non-iterative algorithm ELM to train basic classifiers, whereas the later uses iterative algorithm BP to train basic classifiers. The purpose of this experiment is to verify that the MR-FI-ELM is more efficient than the MR-FI-BP due to the non-iterative learning. In this experiment, we mainly focus on comparing the running time that the two algorithms used on the six data sets under the circumstance that the accuracy ( $F$ -measure and  $G$ -mean) at the same level as given in the second columns of Tables 5 and 6, the experimental results of the running time of the two algorithms are given in Table 7. From Table 7, we can find that for every data set the running time of MR-FI-ELM is much lower than the time of MR-FI-BP. In other words, from the view point of experiments, the conclusion that the MR-FI-ELM is more efficient than the MR-FI-BP is true.

**Table 7** The comparison of MR-FI-ELM and MR-FI-BP on running time (second)

Data sets	MR-FI-ELM	MR-FI-BP
Artificial-1	72.50	184.29
Artificial-2	100.25	234.55
Skin_segment	119.32	285.14
PokerHand	80.31	199.00
CoverType	88.76	216.47
Liver	55.02	100.09

#### 4.3 The statistical analysis on the experimental results of $F$ -measure and $G$ -mean

In order to further verify the effectiveness of the proposed algorithm, we statistically analyzed the experimental results of  $F$ -measure and  $G$ -mean with paired  $T$  test in confidence level 0.05 (Janez 2006; Zhai 2011; Zhai et al. 2012). For the limitation of pages, we only provide here the statistical analysis on  $F$ -measure and  $G$ -mean, and omit the statistical analysis on running time. Firstly, for each data set and for each algorithm, we run the 10-fold cross-validation 10 times and obtain 4 statistics denoted by  $X_1$ ,  $X_2$ ,  $X_3$  and  $X_4$  (where  $X_i$  ( $i = 1, 2, 3, 4$ ) are 100-dimensional vector) corresponding to SMOTE, SMOTE+RF-BigData, MR-V-ELM and MR-FI-ELM, respectively. Next the paired  $T$  test is applied to the experimental results by computing the values of MATLAB function  $ttest2(X_1, X_4)$ ,  $ttest2(X_2, X_4)$  and  $ttest2(X_3, X_4)$ . The results of the statistical analysis on the  $F$ -measure and the  $G$ -mean are listed in Table 8. From the  $p$  values listed in Table 8, we can confirm that MR-FI-ELM

**Table 8** The results of statistical analysis on  $F$ -measure and  $G$ -mean

Data sets	$F$ -measure			$G$ -mean		
	$p$ value 1	$p$ value 2	$p$ value 3	$p$ value 1	$p$ value 2	$p$ value 3
Artificial-1	1.846e−06	1.910e−05	2.781e−03	2.002e−06	1.155e−04	3.056e−03
Artificial-1	2.113e−05	1.515e−04	1.804e−04	2.109e−05	1.971e−04	2.159e−04
Skin_segment	2.777e−06	3.104e−04	1.999e−04	2.702e−06	2.616e−05	1.777e−05
PokerHand	1.999e−05	1.847e−05	3.006e−05	1.505e−05	1.874e−04	2.304e−04
CoverType	1.555e−06	2.089e−04	2.010e−04	1.664e−05	2.000e−05	1.405e−04
Liver	3.003e−05	2.508e−05	1.880e−03	1.874e−05	1.382e−04	2.507e−03

statistically outperforms SMOTE, SMOTE+RF-BigData and MR-V-ELM.

## 5 Conclusion

Based on MapReduce, non-iterative learning, ensemble learning and oversampling, this paper proposed an efficient classification algorithm for imbalanced big data. The proposed algorithm MR-FI-ELM has three advantages: (1) its idea is simple and it is easy to implement. (2) It not only can effectively classify imbalanced big data, but also can deal with practical problems with medium-size. (3) It is feasible and effective for extremely imbalanced data. The experimental results show that (1) the MR-FI-ELM is much more feasible and outperforms the three related algorithms: SMOTE, SMOTE+RF-BigData and MR-V-ELM, (2) the MR-FI-ELM is more efficient than MR-FI-BP. In the future work, we will investigate that the parameter  $p$  is dynamically changed and that the influence of the dynamic  $p$  on the performance of the proposed algorithm. In addition, we will extend the proposed algorithm to the scenario of multiple classification of imbalanced big data.

**Acknowledgements** This research is supported by The National Natural Science Foundation of China (71371063) and by The Natural Science Foundation of Hebei Province (F2017201026).

## Compliance with ethical standards

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

Abdallah ACB, Frigui H, Gader P (2012) Adaptive local fusion with fuzzy integrals. *IEEE Trans Fuzzy Syst* 20(5):849–864

Abdi L, Hashemi S (2016) To combat multi-class imbalanced problems by means of over-sampling techniques. *IEEE Trans Knowl Data Eng* 28(1):238–251

Bao L, Juan C, Li JT et al (2016) Boosted near-miss under-sampling on SVM ensembles for concept detection in large-scale imbalanced datasets. *Neurocomputing* 172:198–206

Berkeley EC, Bobrow DG (1964) The programming language LISP: its operation and applications. *Program Lang LISP Oper Appl* 21(99):1803–1824

Cai MJ, Li QG, Ma JM (2017) Knowledge reduction of dynamic covering decision information systems caused by variations of attribute values. *Int J Mach Learn Cybern* 8(4):1131–1144

Cao WP, Wang XZ, Ming Z et al (2018) A review on neural networks with random weights. *Neurocomputing* 275:278–287

Cao P, Zhao D, Zaiane O (2013) An optimized cost-sensitive SVM for imbalanced data learning. In: *PAKDD part II. LNAI 7819*, pp 280–292

Castro CL, Braga AP (2013) Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. *IEEE Trans Neural Netw Learn Syst* 24(6):888–899

Chawla NV, Lazarevic A, Hall LO, et al (2003) SMOTEBoost: improving prediction of the minority class in boosting. In: *Proceeding of knowledge discovery in databases*, pp 107–119

Chawla NV, Bowyer KW, Hall LO et al (2002) SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res* 16:321–357

Das B, Krishnan NC, Cook DJ (2015) RACOG and wRACOG: two probabilistic oversampling techniques. *IEEE Trans Knowl Data Eng* 27(1):222–234

Dean J, Ghemawat S (2008) MapReduce: simplified data processing on large clusters. *Commun ACM* 51(1):107–113

Ding S, Zhang N, Zhang J et al (2017) Unsupervised extreme learning machine with representational features. *Int J Mach Learn Cybern* 8(2):587–595

El-Monsef MEA, El-Gayar MA, Aqeel RM (2017) A comparison of three types of rough fuzzy sets based on two universal sets. *Int J Mach Learn Cybern* 8(1):343–353

Emani CK, Cullot N, Nicolle C (2015) Understandable Big Data: a survey. *Comput Sci Rev* 17:70–81

Fan W, Stolfo SJ, Zhang J, Chan PK (1999) Adacost: misclassification cost-sensitive boosting. Presented at the 6th international conference on machine learning, San Francisco, pp 97–105

Fernández A, RÍo SD, Chawla NV et al (2017) An insight into imbalanced Big Data classification: outcomes and challenges. *Complex Intell Syst* 3(2):105–120

Galar M, Fernández A, Barrenechea E et al (2012) A review on ensembles for the class imbalance problem-bagging boosting and hybrid-based approaches. *IEEE Trans Syst Man Cyber Part C Appl Rev* 42(4):463–484

Galar M, Fernández A, Barrenechea E et al (2013) EUSBoost: enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern Recogn* 46(12):3460–3471

Ghanavati M, Wong RK, Chen F, et al (2014) An effective integrated method for learning big imbalanced data. In: *IEEE international congress on Big Data*, pp 691–698

He HB, Garcia EA (2009) Learning from imbalanced data. *IEEE Trans Knowl Data Eng* 21(9):1263–1284

- Huang GB (2015) What are extreme learning machines? Filling the gap between Frank Rosenblatts dream and John von Neumanns puzzle. *Cognit Comput* 7(3):263–278
- Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. *Neurocomputing* 70:489–501
- Huang GB, Chen L, Siew CK (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans Neural Netw* 17(4):879–892
- Huang GB, Wang DH, Lan Y (2011) Extreme learning machines: a survey. *Int J Mach Learn Cybern* 2(2):107–122
- Huang GB, Zhou HM, Ding XJ, Zhang R (2012) Extreme learning machine for regression and multiclass classification. *IEEE Trans Syst Man Cybern Part B* 42(2):513–529
- Huang G, Huang GB, Song S, You K (2015) Trends in extreme learning machines: a review. *Neural Netw* 61:32–48
- Igelnik B, Pao YH (1995) Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE Trans Neural Netw* 6(6):1320–1329
- Janez D (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7(1):1–30
- Joshi M, Kumar V, Agarwal R (2001) Evaluating boosting algorithms to classify rare classes: comparison and improvements. In: *Proceeding of IEEE international conference on data mining*, pp 257–264
- Krawczyk B (2016) Learning from imbalanced data: open challenges and future directions. *Prog Artif Intell* 5:221–232
- Krawczyk B, Woźniak M, Schaefer G (2014) Cost-sensitive decision tree ensembles for effective imbalanced classification. *Appl Soft Comput* 14:554–562
- Kuncheva LI (2001) Combining classifiers: soft computing solutions. In: Pal SK, Pal A (eds) *Pattern recognition: from classical to modern approaches*. World Scientific, Singapore, pp 427–451
- Lee W, Jun CH, Lee JS (2017) Instance categorization by support vector machines to adjust weights in AdaBoost for imbalanced data classification. *Inf Sci* 381:92–103
- Li KW, Shao MW, Wu WZ (2017) A data reduction method in formal fuzzy contexts. *Int J Mach Learn Cybern* 8(4):1145–1155
- Lichman M (2013) UCI machine learning repository. School of Information and Computer Science, University of California, Irvine. <http://archive.ics.uci.edu/ml>
- Liu XM, Liu B (2016) A liver data set with five attributes and two imbalanced classes. <https://github.com/ShenData/data.git>
- Liu XY, Wu J, Zhou ZH (2009) Exploratory undersampling for class imbalance learning. *IEEE Trans Syst Man Cybern B Cybern* 39(2):539–550
- López V, del Ro S, Bentez JM et al (2015) Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data. *Fuzzy Sets Syst* 258(1):5–38
- Ludwig SA (2015) MapReduce-based fuzzy c-means clustering algorithm: implementation and scalability. *Int J Mach Learn Cybern* 6(6):923–934
- Mao WT, Wang JW, Xue ZN (2017) An ELM-based model with sparse-weighting strategy for sequential data imbalance problem. *Int J Mach Learn Cybern* 8(4):1333–1345
- Maurya CK, Toshniwal D, Venkatarao GV (2016) Online sparse class imbalance learning on big data. *Neurocomputing* 216:250–260
- Meng M, Wei J, Wang JB et al (2017) Adaptive semi-supervised dimensional reduction based on pairwise constraints weighting and graph optimizing. *Int J Mach Learn Cybern* 8(3):793–805
- Ofek N, Rokach L, Stern R et al (2017) Fast-CBUS: a fast clustering-based undersampling method for addressing the class imbalance problem. *Neurocomputing* 243:88–102
- Oscar FR, Beatriz PS, Bertha GB (2017) An incremental non-iterative learning method for one-layer feedforward neural networks. *Appl Soft Comput*. <https://doi.org/10.1016/j.asoc.2017.07.061>
- Pao YH, Phillips SM, Sobajic DJ (1992) Neural-net computing and the intelligent control of systems. *Int J Control* 56(2):263–289
- Pao YH, Park GH, Sobajic DJ (1994) Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing* 6(94):163–180
- Ralescu D, Adams G (1980) The fuzzy integral. *J Math Anal Appl* 75(2):562–570
- Ramentol E, Vluymans S, Verbiest N et al (2015) IFROWANN: imbalanced fuzzy-rough ordered weighted average nearest neighbor classification. *IEEE Trans Fuzzy Syst* 23(5):1622–1637
- Ren ZL, Wei CP (2017) A multi-attribute decision-making method with prioritization relationship and dual hesitant fuzzy decision information. *Int J Mach Learn Cybern* 8(3):755–763
- Reznik AM (1999) Non-iterative learning for neural networks. In: *Proceedings of the international joint conference on neural networks (IJCNN1999)*, vol 2, pp 1374–1379
- Río SD, López V, Benítez JM et al (2014) On the use of MapReduce for imbalanced big data using random forest. *Inf Sci* 285:112–137
- Rivera WA (2017) Noise reduction a priori synthetic over-sampling for class imbalanced data sets. *Inf Sci* 408:146–161
- Rivera WA, Xanthopoulos P (2016) A priori synthetic over-sampling methods for increasing classification sensitivity in imbalanced data sets. *Expert Syst Appl* 66:124–135
- Schmidt WF, Kraaijveld MA, Duin RPW (1991) A non-iterative method for training feedforward networks. In: *Proceedings of the international joint conference on neural networks (IJCNN1991)*, vol 2, pp 19–24
- Schmidt WF, Kraaijveld MA, Duin RPW (1992) Feed forward neural networks with random weights. In: *Proceedings of 11th IAPR international conference on pattern recognition methodology and systems*, vol 2, pp 1–4
- Seiffert C, Khoshgoftaar T, Hulse JV et al (2010) Rusboost: a hybrid approach to alleviating class imbalance. *IEEE Trans Syst Man Cybern Part A Syst Hum* 40(1):185–197
- Sun Y, Kamel MS, Wong AK et al (2007) Cost-sensitive boosting for classification of imbalanced data. *Pattern Recogn* 40(12):3358–3378
- Triguero I, Galar M, Merino D, et al (2016) Evolutionary under-sampling for extremely imbalanced big data classification under apache spark. In: *2016 IEEE congress on evolutionary computation (CEC2016)*, pp 640–647
- Tsang ECC, Sun BZ, Ma WM (2017) General relation-based variable precision rough fuzzy set. *Int J Mach Learn Cybern* 8(3):891–901
- Wang XZ (2015) Uncertainty in learning from Big Data-editorial. *J Intell Fuzzy Syst* 28(5):2329–2330
- Wang XZ, Dong CR (2009) Improving generalization of fuzzy if-then rules by maximizing fuzzy entropy. *IEEE Trans Fuzzy Syst* 17(3):556–567
- Wang XZ, Xing HJ, Li Y et al (2015) A study on relationship between generalization abilities and fuzziness of base classifiers in ensemble learning. *IEEE Trans Fuzzy Syst* 23(5):1638–1654
- Wang XZ, Zhang TL, Wang R (2017) Noniterative deep learning: incorporating restricted Boltzmann machine into multilayer random weight neural networks. *IEEE Trans Syst Man Cybern Syst*. <https://doi.org/10.1109/TSMC.2017.2701419>
- Wang XZ, Wang R, Xu C (2017) Discovering the relationship between generalization and uncertainty by incorporating complexity of classification. *IEEE Trans Cybern*. <https://doi.org/10.1109/TCYB.2017.2653223>
- Wang R, Wang XZ, Kwong S et al (2017) Incorporating diversity and informativeness in multiple-instance active learning. *IEEE Trans Fuzzy Syst* 25(6):1460–1475
- Wang XZ, Wang R, Xu C (2018) Discovering the relationship between generalization and uncertainty by incorporating complexity of classification. *IEEE Trans Cybern* 48(2):703–715
- Ye J (2017) Multiple attribute group decision making based on interval neutrosophic uncertain linguistic variables. *Int J Mach Learn Cybern* 8(3):837–848

- Zhai JH (2011) Fuzzy decision tree based on fuzzy-rough technique. *Soft Comput* 15(6):1087–1096
- Zhai JH, Xu HY, Wang XZ (2012) Dynamic ensemble extreme learning machine based on sample entropy. *Soft Comput* 16(9):1493–1502
- Zhai JH, Wang XZ, Pang XH (2016) Voting-based instance selection from large data sets with MapReduce and random weight networks. *Inf Sci* 367:1066–1077
- Zhai JH, Zhang Y, Zhu HY (2017) Three-way decisions model based on tolerance rough fuzzy set. *Int J Mach Learn Cybern* 8(1):35–43
- Zhai JH, Zhang SF, Wang CX (2017) The classification of imbalanced large data sets based on MapReduce and ensemble of ELM classifiers. *Int J Mach Learn Cybern* 8(3):1009–1017
- Zhan YZ, Zhang J, Mao QR (2012) Fusion recognition algorithm based on fuzzy density determination with classification capability and supportability. *Pattern Recognit Artif Intell* 25(2):346–351
- Zhang L, Zhang D (2015) Domain adaptation extreme learning machines for drift compensation in E-nose systems. *IEEE Trans Instrum Meas* 64(7):1790–1801
- Zhang L, Zhang D (2016) Robust visual knowledge transfer via extreme learning machine based domain adaptation. *IEEE Trans Image Process* 25(10):4959–4973
- Zhang L, Zuo WM, Zhang D (2016) LSDT: latent sparse domain transfer learning, for visual adaptation. *IEEE Trans Image Process* 25(3):1177–1191
- Zhao SY, Wang XZ, Chen DG et al (2013) Nested structure in parameterized rough reduction. *Inf Sci* 248:130–150
- Zhao SY, Chen H, Li CP et al (2015) A novel approach to building a robust fuzzy rough classifier. *IEEE Trans Fuzzy Syst* 23(4):769–786
- Zong WW, Huang GB, Chen YQ (2013) Weighted extreme learning machine for imbalance learning. *Neurocomputing* 101(3):229–242