A Novel Spark-based Attribute Reduction and Neighborhood Classification for Rough Evidence

A Novel Spark-based Attribute Reduction and Neighborhood Classification for Rough Evidence

Weiping Ding, *Senior Member IEEE*, Ying Sun, Ming Li, Jun Liu, *Senior Member IEEE*, Hengrong Ju, Jiashuang Huang, and Chin-Teng Lin, *Fellow, IEEE*

***Abstract*—Neighborhood classification (NEC) algorithms has been widely used to solve classification problems. Most traditional NEC algorithms employ the majority voting mechanism as the basis for final decision-making. However, this mechanism hardly considers the spatial difference and label uncertainty of the neighborhood samples, which may increase the possibility of the misclassification. In addition, the traditional NEC algorithms need to load the whole data into memory at once, which is computationally inefficient when the size of data set is large. To address these problems, we propose a novel Spark-based attribute reduction and neighborhood classification for rough evidence in this paper. Specifically, we first construct a multi-granular sample space using parallel undersampling method. Then, we evaluate the significance of attribute by neighborhood rough evidence decision error rate and remove the redundant attribute on different samples subspaces. Based on this attribute reduction algorithm, we design a parallel attribute reduction algorithm which is able to compute equivalence classes in parallel and parallelize the process of searching for candidate attributes. Finally, we introduce the rough evidence into the classification decision of traditional NEC algorithms and parallelize the classification decision process. Furthermore, the proposed algorithms are conducted in the Spark parallel computing framework. Experimental results on both small and large-scale data sets show that the proposed algorithms outperform the benchmarking algorithms in the classification accuracy and the computational efficiency.**

***Index Terms*—Rough sets, Dempster-Shafer theory, Parallel attribute reduction, Parallel neighborhood classification, Spark framework.**

I. Introduction

In recent years, with the booming development of information technology, the exponential growth of data has caused serious challenges to traditional data mining and analysis technologies, and brought valuable opportunities to the development of various industries [1], [2]. Because of the huge scale and low value density of big data, it has brought several challenges to traditional data mining techniques. The current big data processing technologies [3], [4] mainly use cloud computing, distributed computing and parallel computing to slice large-scale data into multiple subsets, thereby reducing the scale of data and making it computable. However, these techniques do not remove the redundant attributes and information from the large-scale data, thereby reducing the uncertainty among the large-scale data.

Granular computing has gradually developed into a new branch in the field of artificial intelligence. The granular computing models have been proposed include: rough set [5], fuzzy set [6], fuzzy rough set [7], three-way decision [8], etc. Rough set is one of the granular computing models, which can deal with and analyze the uncertain information in the decision-making process. Attribute reduction [9], [10], [11] is an important concept in rough sets, which can reduce the complexity and uncertainty of data by eliminating redundant attributes. How to use rough set theory to extract the huge potential value from large- scale data has long been one of the research topics of many scholars. For the "5V" characteristic of large-scale data, Li et al. [12] discussed the applicability of the a PICKT solution for big data analysis based on the theories of Granular Computing and Rough Set. Qian et al. [13], [14] and Zhang et al. [15] used MapReduce operations to aggregate equivalence classes to reduce the repetitive calculation time of equivalence classes, and proposed a parallel rough set extension model.

In order to avoid partial information loss in the process of during data discretization, Hu et al. [16] proposed a neighborhood rough set model based on the neighborhood relation, which can directly deal with numerical data. In the neighborhood rough set model, different neighborhood space samples can be derived according to different neighborhood radius, and the overall sample space can be depicted from different granularities. In addition, in response to the problems encountered by the neighborhood rough set model when dealing with complex data, the researchers proposed the corresponding neighborhood rough set extension models, i.e., decision-neighborhood rough set [17], [18], local neighborhood rough sets [19], [20], multi-granularity neighborhood rough sets [21], [22], etc.

This work is supported in part by the Natural Science Foundation of China under Grant 61976120, 62006128 and 62102199, the Natural Science Foundation of Jiangsu Province under Grant BK20191445, and the Natural Science Key Foundation of Jiangsu Education Department under Grant 21KJA510004 (*Corresponding author: Weiping Ding.*)

W. Ding, Y. Sun, M. Li, H. Ju, and J. Huang are with the School of Computer Science and Technology, Nantong University, Nantong 226019, China. E-mail: dwp9988@163.com, sunying_1997829@163.com, liming_2014@163.com, juhengrong@ntu.edu.cn, hjshdym@163.com.

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

C.-T. Lin is with the Centre for Artificial Intelligence, FEIT, University of Technology Sydney, Ultimo, NSW 2007, Australia. E-mail: chin-teng.lin@uts.edu.au.

Hu et al. [23] proposed a neighborhood classifier (NEC) which based on the neighborhood rough set model. Unlike the traditional K-NN model which selects the nearest k samples, the neighborhood classifier obtains the surrounding neighborhood (SN) samples via neighborhood radius. The neighbors of a sample will consider not only in terms of closeness but also in terms of their spatial distribution regarding that sample. However, KNN and NEC assign a category label to the unclassified sample via a majority voting mechanism where all nearest neighbors are considered equally important. And it hardly considers the spatial difference [24], [25], [26] and label uncertainty [27], [28], [29] of neighborhood samples. To address the two shortcomings of the majority voting mechanism, Denoeux [27] applied Dempster-Shafer (D-S) evidence theory to K-NN to improve classification accuracy, which fused evidence support information of all neighborhood samples to predict the sample label. Although D-S evidence theory can fully consider the spatial distance and label information, it still easily suffers from conflicting evidence information between different labels [30]. In order to reduce the conflict of evidence information between different classes, this paper enhances the local evidence information of the majority class samples in the neighborhood by the rough membership function.

Another challenge is that the time complexity and computation cost of the attribute reduction and classification are very high when dealing with large-scale data. Parallel algorithm in distributed computing framework is a practical pathway which can reduce the time complexity and improve computational efficiency. Spark is a distributed computing framework based on in-memory computing with faster computation speed and better iterative performance than Hadoop MapReduce [31]. Yin et al. [32] designed a new parallel attribute reduction4 based on Spark to resolve the limitations of MapReduce. Luo et al. [33], [34] proposed a novel Spark parallel attribute reduction based on a rough hypercuboid model, which employed two parallel strategies, vertical partitioning and horizontal partitioning. However, there are not only nominal data but also numerical data in large-scale data. In the above studies, most of the parallel extension models based on classical rough sets have been implemented, but this is only applicable to nominal data and cannot directly process large-scale numerical data.

In this paper, we proposed a novel Spark-based attribute reduction and neighborhood classification for rough evidence, which aims to address the drawbacks of attribute reduction and neighborhood classification in massive data processing, to better deal with data redundancy and reduce data uncertainty. The main contributions of the proposed work are summarized below.

(1) We first introduce the rough membership function into the combination rule of D-S evidence theory and propose the rough evidence. The rough evidence can enhance the local evidence information of the majority class samples in the neighborhood to obtain more reliable evidence information and reduce the degree of evidence information confliction. Furthermore, we employ rough evidence for the attribute reduction process to evaluate attribute significance and the neighborhood classification process to obtain decision information.

(2) Through Spark parallel computing mechanism, multi-granular sample space based on Spark is constructed using parallel random undersampling algorithm, which can reflect the distinction of universe in terms of multi-granularity.

(3) Based on the rough evidence, this paper implements a new Spark parallel attribute reduction based on neighborhood rough evidence decision error rate (called SPAR-NREDER), which can significantly improve the computational efficiency of attribute reduction. Moreover, the neighborhood rough evidence decision error rate can more precisely reflect the situation of misclassified samples in the boundary region.

(4) Based on the proposed parallel attribute reduction algorithm, this paper implements a novel Spark-based rough evidence neighborhood classification (called SRENEC), which can be fused by parallel the rough evidence information of intra-class samples and inter-class samples in the neighborhood. This algorithm not only improve the classification accuracy of the NEC algorithm, but also solve the problem that the NEC algorithm is computationally inefficient when the size of data set is large.

In addition, to better demonstrate the effectiveness of the proposed algorithm in this paper, experimental studies are conducted to evaluate the classification performance of the algorithm through imbalanced benchmark data.

The rest of this paper is organized as follows. Some preliminaries are provided in Section II. In Section III, the SPAR-NREDER and the SRENEC algorithms are introduced in details respectively. A series of experiments are conducted on both small and large-scale datasets in Section IV, and then we compare and analyze the experimental results and verify the performance of the SPAR-NREDER and SRENEC. The discussion of this paper is in Section V. The conclusions of this paper are demonstrated in Section VI.

II. Preliminaries

A. Neighborhood Rough Sets and Classification Methods

From the perspective of neighborhood granulation in numerical space, Hu et al. [16] introduced the neighbor-hood space into rough set, and reinterpret the neighbor-hood rough set model. In addition, Hu et al. [23] pro-posed a neighborhood classifier based on neighborhood rough sets.

Definition 1 ([16]): In a decision information system $S=<U,C\bigcup D,V,f>$, let $B\subseteq C$ be a subset of condition attributes. The neighborhood space with respect to B for $\forall x_i \in U$ is defined as:

$$\delta_B(x_i)=\{x_j \mid x_j\in U, dis_B(x_i,x_j)\}, \tag{1}$$

where $dis_B(x_i,x_j)$ is the distance with respect to B between the sample x_i and x_j.

Definition 2 ([16]): In a decision information system $S=<U,C\bigcup D,V,f>$, for arbitrary $X\subseteq U$ and $B\subseteq C$, N is a neighborhood relation on U, then the lower and upper approximation sets of X in terms of N with respect to B are defined and denoted respectively as:

$$\underline{N_B}(X)=\{x_i\in U \mid \delta_B(x_i)\subseteq X\}, \tag{2}$$

$$\overline{N_B}(X)=\{x_i\in U \mid \delta_B(x_i)\bigcap X\neq\varnothing\}. \tag{3}$$

The positive, negative and boundary regions of X in terms of N with respect to B are defined and denoted respectively as:

$$POS_B\left(X\right)=\underline{N_B}\left(X\right), \tag{4}$$

$$NEG_B\left(X\right)=U-\underline{N_B}\left(X\right), \tag{5}$$

$$BND_B\left(X\right)=\overline{N_B}\left(X\right)-\underline{N_B}\left(X\right). \tag{6}$$

Definition 3 ([35]): In a decision information system $S=<U,C\bigcup D,V,f>$, for arbitrary $B\subseteq C$, $X\subseteq U$, $\forall x_i\in U$, the rough membership function of x_i with respect to X is defined as:

$$\mu_X^{\delta_B}\left(x_i\right)=\frac{\left|\delta_B\left(x_i\right)\bigcap X\right|}{\left|\delta_B\left(x_i\right)\right|}, \tag{7}$$

where $\delta_B\left(x_i\right)$ is the neighborhood information granule of sample x_i.

Definition 4 ([16]): In a decision information system $S=<U,C\bigcup D,V,f>$, for arbitrary $x_i\in U$ and $B\subseteq C$, the true category label of x_i is $\omega\left(x_i\right)$, and $ND\left(x_i\right)$ is the category label of the sample x_i predicted by the neighborhood classification algorithm. For the case of misclassified samples, a 0-1 loss function is expressed as:

$$\lambda(\omega(x_i)\mid ND(x_i))=\begin{cases}0, & \omega(x_i)=ND(x_i)\\ 1, & otherwise\end{cases}. \tag{8}$$

The neighborhood decision error rate (called NDER) is calculated as [16]:

$$NDER=\frac{1}{M}\sum_{i=1}^{M}\lambda(\omega(x_i)\mid ND(x_i)), \tag{9}$$

where M denotes the number of samples.

Definition 5 ([16]): In a decision information system $S=<U,C\bigcup D,V,f>$, for arbitrary $B\subseteq C$, if B is a subset of attribute reduction based on the neighborhood decision error rate, then the following properties for B hold:

(1) $NDER_B\leq NDER_C$;

(2) for $\forall B^{'}\subseteq B$, there are $NDER_{B^{'}}\geq NDER_B$.

Hu et al. [23] proposed the NEC algorithm based on the neighborhood rough set model, which divides the data space into neighborhood spaces with different centers according to the neighborhood radius. And this algorithm predicts the category labels of unclassified samples from the distribution of neighborhood sample labels.

Definition 6 ([23]): Let $\delta_B\left(x_i\right)$ be the neighborhood space of unclassified sample x_i and x_j be any sample in the neighborhood space whose category label is ω_j. According to the traditional NEC algorithm, then the category label of x_i can be obtained as:

$$\omega=\underset{\omega_k\in\{\omega_1,\omega_2,\ldots,\omega_m\}}{\arg\max}\left(\sum_{x_j\in\delta_B(x_i)}H\left(\omega_j,\omega_k\right)\right), \tag{10}$$

where $H\left(\omega_j,\omega_k\right)=\begin{cases}1, & \omega_j=\omega_k\\ 0, & otherwise\end{cases}$ is the judgment function.

B. Dempster-Shafer Evidence Theory

D-S evidence theory [27] is a theory of evidence fusion based on belief trust function, which is an extension of Bayesian theory in probability theory. In the traditional Bayesian probability theory, the answer to a question is the most probable answer among all possible answers. While in D-S evidence theory, the answer to a question is obtained by fusing all possible answers.

Definition 7 ([27]): In the discriminative framework Ω, the sample space can be partitioned into multiple subsets , where for any $X_s\subseteq\Omega$ corresponds to a basic probability number $M\in\left[0,1\right]$, which can be called the basic probability assignment (BPA) function of the power set 2^{Ω}, and satisfies:

$$M(\phi)=0, \tag{11}$$

$$\sum\nolimits_{s=1}^{d}M(\{X_s\})=1, \tag{12}$$

in which the basic probability of impossible event is 0 and $M\left(\left\{X_s\right\}\right)$ is the basic probability function of X_s.

Definition 8 ([27]): According to the basic probability assignment function, the belief function Bel and the plausibility function Pl are expressed respectively as:

$$Bel(\{X_s\})=\sum\nolimits_{B\subseteq X_s}M(B), \tag{13}$$

$$Pl(\{X_s\})=1-Bel(\{\overline{X_s}\})=\sum\nolimits_{B\bigcap X_s\neq\varnothing}M(B), \tag{14}$$

where $Bel(\{X_s\})$ is interpreted as a measure of the total belief committed to X_s and $Pl(\{X_s\})$ defines the degree of failing to doubt in X_s.

C. Spark Framework

Apache Spark is a revolutionary framework that can perform in-memory computing faster than Hadoop MapReduce for large-scale preprocessing. Compared with Hadoop MapReduce, Spark [2], [4] is characterized by fast computation speed, ease of use, high generality, and run everywhere. Spark can cache the intermediate data into memory during computation and can persist data into disk when memory overflows.

Resilient distributed dataset (RDD) [36] is a fault-tolerant parallel data structure, which is the core computational model in Spark. RDDs had two types of parallel operations [37]: transformation which returns a pointer to the new RDD, and action which returns a value to the deriver after running the computation. Spark provides four APIs, i.e., Java, Scala, R, and Python. Meanwhile, many developers had introduced a series of components based on Spark, such as Spark SQL [38], Spark Streaming [39], MLLib [40], and GraphX [41], forming a cloud computing platform for large-scale data preprocessing.

The flow of Spark task execution is shown in Fig. 1. The process of task execution is partitioned into two phases in Spark [42]:

(1) In the first stage, data is read from HDFS to generate RDD objects, and a directed acyclic graph (DAG) is constructed according to the lineage graph relationship between RDDs, and RDD operations are divided into different tasks.

(2) In the second stage, different tasks are assigned to slaves for execution through the cluster manager.

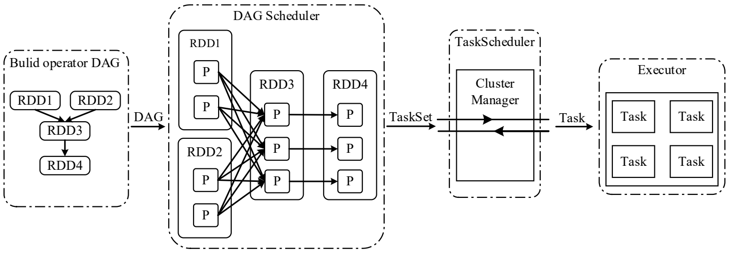


Fig. 1. Flow chart of Spark task execution.

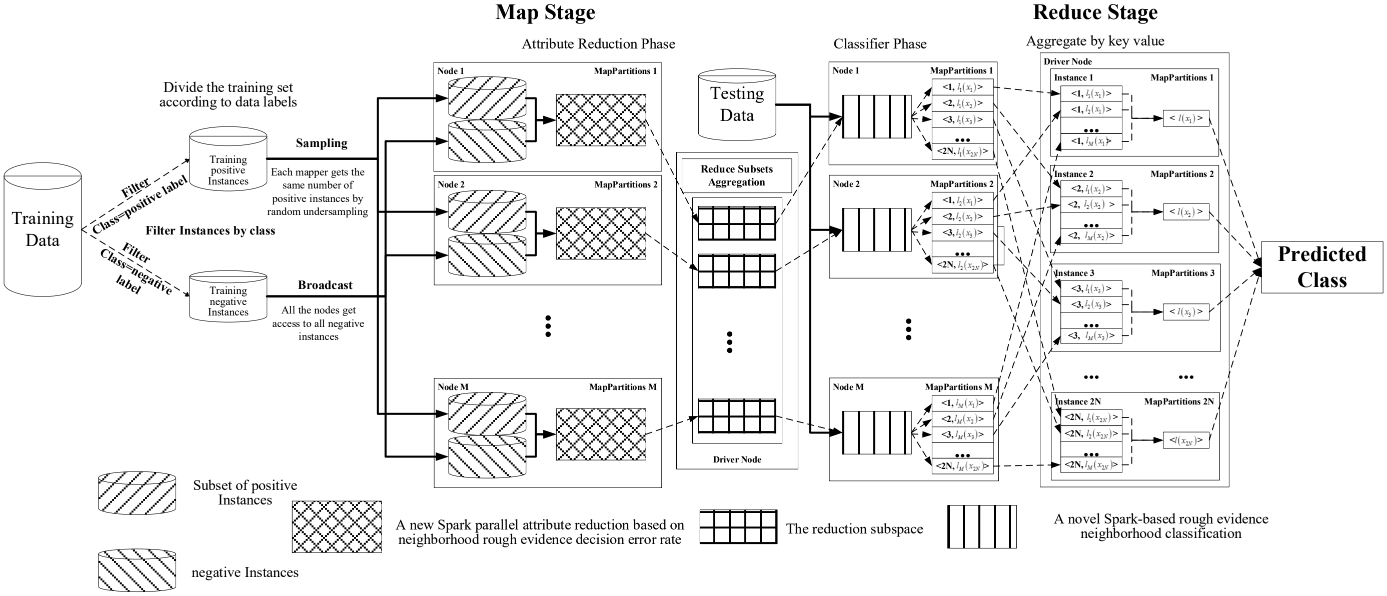

Map Stage
Reduce Stage
Attribute Reduction Phase
Classifier Phase
Aggregate by key value
Divide the training set according to data labels
Training Data
Filter Class=positive label
Filter Class=negative label
Filter Instances by class
Training positive Instances
Training negative Instances
Sampling
Each mapper gets the same number of positive instances by random undersampling
Broadcast
All the nodes get access to all negative instances
Node 1
MapPartitions 1
Node 2
MapPartitions 2
Node M
MapPartitions M
Testing Data
Reduce Subsets Aggregation
Driver Node
Instance 1
Instance 2
Instance 3
Instance 2N
MapPartitions 2N
Predicted Class
Subset of positive Instances
negative Instances
A new Spark parallel attribute reduction based on neighborhood rough evidence decision error rate
The reduction subspace
A novel Spark-based rough evidence neighborhood classification

Fig. 2. The data processing of a novel Spark-based attribute reduction and neighborhood classification for rough evidence.

III. A Novel Spark-based Attribute Reduction and Neighborhood Classification for Rough Evidence

In this section, we propose a novel Spark-based attribute reduction and neighborhood classification for rough evidence. Specifically, we first construct a multi-granular sample space using parallel undersampling method. Then, we evaluate the significance of attribute by neighborhood rough evidence decision error rate and remove the redundant attribute on different samples subspaces. Based on this attribute reduction algorithm, we design a new Spark parallel attribute reduction based on neighborhood rough evidence decision error rate (called SPAR-NREDER) which is able to compute equivalence classes in parallel and parallelize the process of searching for candidate attributes. Finally, we introduce the rough evidence into the classification decision of traditional NEC algorithms and parallelize the classification decision process. Thus, we propose a novel Spark-based rough evidence neighborhood classification (called SRENEC), which can be fused by parallel the rough evidence information of intra-class samples and inter-class samples in the neighborhood. Furthermore, the proposed algorithms are conducted in the Spark parallel computing framework. The process of SRENEC algorithm is shown in Fig. 2.

A. The Method of Fusing the Rough Evidence Information

The rough evidence is a novel decision rule using the rough membership function and D-S evidence theory. The rough membership function is applied to the global evidence information merging process, which enhances the local evidence information of the majority class samples in the neighborhood and reduces the degree of conflict between the local evidence information of the different classes of samples. Specifically, the rough evidence first computes the evidence information of the neighborhood samples of sample x_t that support its belonging to decision class X_q. Next, the local evidence information can be fused to the evidence information provided by the neighborhood samples with decision class X_q of sample x_t for its belonging to decision class X_q by the Demspter's rule. Then, the local rough evidence information strengthens the local evidence information provided by the majority class samples in the neighborhood. Finally, according to the Demspter's rule, the global evidence information can be obtained by fusing the local evidence information of different classes of sample x_t.

Definition 9 ([27]): In a decision information system $S=<U,C\bigcup D,V,f>$, the decision attributes D divides the university U into several non-intersecting equivalence classes, which can be denominated as $\pi = U/D = \left\{X_1, X_2, \ldots, X_d\right\}$. The category labels of the samples $x_j \in U$ can be denoted as $\omega = \left\{1, 2, \ldots, d\right\}$. For arbitrary $x_j \in U$, the category label of x_j is $\omega_j = q \in \left\{1, 2, \ldots, d\right\}$. Specially, the samples pairs $\left(x_j, X_q\right)$ are the evidence support information for x_t in the decision equivalence class X_q. The evidence support information is defined as:

$$M^{t,j}(\{X_q\}) = \beta_0 \cdot \exp\{-\gamma_q \cdot (dis(x_t, x_j))^2\}, \tag{15}$$

$$M^{t,j}(\pi) = 1 - M^{t,j}(\{X_q\}), \tag{16}$$

where $0 < \beta_0 < 1$, $\gamma_q > 0$, δ_R is represented as a neighborhood of x_t with respect to the R.

Obviously, $M^{t,j}(\{X_q\})$ is the belief of the evidence support information provided by x_j to support that x_t belongs to decision equivalence class X_q and $M^{t,j}(\pi)$ is the belief of the evidence information provided by x_j to support that x_t belongs to decision equivalence class π. The evidence support information provided by x_j is mainly determined by two parameters and the distance between sample x_t and its neighborhood sample. β_0 is a class-independent parameter and γ_q is the reciprocal of the mean distance between all training samples with category label q.

Then, the local evidence support information provided by the neighborhood samples of sample x_t with category label q can be aggregated, where the combination rules of the D-S evidence theory can be denoted as $M_q^t = \oplus_{x_j \in \delta_{R_i}^q(x_t)} M^{t,j}$. The neighborhood space of an unclassified sample x_t with the category label q can be denoted $\delta_B^q(x_t)$.

Definition 10 ([27]): According to the D-S evidence theory, the local evidence support information of $\delta_B^q(x_t)$ for x_t is expressed as:

$$M_q^t(\{X_q\}) = 1 - \prod\nolimits_{x_j \in \delta_B^q(x_t)} M^{t,j}(\pi), \tag{17}$$

$$M_q^t(\{\pi\}) = \prod\nolimits_{x_j \in \delta_B^q(x_t)} M^{t,j}(\pi). \tag{18}$$

In the whole neighborhood sample space, the evidence support information from different labels may arise conflicts in the integration process, resulting in imprecise global evidence support information obtained finally. Rough membership function can reflect the ratio of samples with different labels in the neighborhood space. Thus, we introduce the rough membership function into the global evidence support information merging process, which can enhance the local evidence information provided by the majority classes in the neighborhood. The local rough evidence information provided by neighborhood samples with the same class label q is given by [35]:

$$RM_q^t(\{X_q\}) = \mu_{X_q}^{\delta} \cdot M_q^t(\{X_q\}), \tag{19}$$

$$RM_q^t(\{\pi\}) = 1 - \mu_{X_q}^{\delta} \cdot M_q^t(\{X_q\}). \tag{20}$$

Similarly, the global rough evidence information from all samples in the whole neighborhood space $M^t = \oplus_{q=1}^d M_q^t$ can be generated by fusing the local rough evidence information M_q^t, and it is obtained as [27], [35]:

$$RM^t(\{X_q\}) = \frac{RM_q^t(\{X_q\}) \prod\nolimits_{r \neq q} RM_r^t(\pi)}{K} \quad q = 1, 2, \ldots, d, \tag{21}$$

$$RM^t(\pi) = \frac{\prod\nolimits_{q=1}^d RM_q^t(\pi)}{K}, \tag{22}$$

where K is the normalization factor and is calculated as follows:

$$K = \sum\nolimits_{q=1}^d RM_q^t(\{X_q\}) \cdot \prod\nolimits_{r \neq q} RM_r^t(\pi) + \prod\nolimits_{q=1}^d RM_q^t(\pi). \tag{23}$$

Definition 11 ([27], [35]): The *RBel* function and *RPl* function of unclassified sample x_t with respect to X_q are obtained respectively as:

$$RBel^t(\{X_q\}) = RM^t(\{X_q\}), \tag{24}$$

$$RPl^t(\{X_q\}) = RM^t(\{X_q\}) + RM^t(\pi). \tag{25}$$

It is not difficult to notice that $RBel^t(\{X_q\})$ is the belief of sample x_t which is committed to X_q.

Definition 12 ([27]): In the neighborhood $\delta_B(x_t)$ of unclassified sample x_t, for arbitrary $x_k \in \delta_B(x_t)$, the category label of x_k is ω_k. Then the predicted category label of x_t in the decision information system S is computed by:

$$l(x_t) = \arg\max_{\omega_k}(RBel^t(\{X_k\})). \tag{26}$$

Example 1: For a sample y, its category label and neighborhood are l_1 and $\delta_C(y) = \{x_1, x_2, x_3, x_4, x_5\}$, respectively. The partition set of $\delta_C(y)$ with the category label is $\delta_C(y)/D = \{X_1, X_2\} = \{\{x_1, x_3, x_5\}, \{x_2, x_4\}\}$. We suppose that the evidence information provided by each sample of neighborhood $\delta_C(y)$ is $\{0.5, 0.85, 0.5, 0.4, 0.6\}$. According to the D-S evidence theory, the local evidence information provided by difference category label is $M_1^y(\{X_1\}) = 0.9$, $M_1^y(\{X_2\}) = 0.91$ and the global evidence information provided by the whole neighborhood is $M^y(\{X_1\}) = 0.4475$, $M^y(\{X_2\}) = 0.5028$. Thus, the predicted label of y is l_2. However, the true label is difference from the predicted label. Therefore, in this situation the D-S evidence theory result in the misclassification of samples.

According to Eq. (19)-(23), the local rough evidence information provided by difference category label is $RM_1^y(\{X_1\}) = 0.54, RM_1^y(\{X_2\}) = 0.364$ and the global rough evidence information provided by the whole neighborhood is $M^y(\{X_1\}) = 0.4275, M^y(\{X_2\}) = 0.1345$. Thus, the predicted label of y is l_1, which is the same as the true label.

Therefore, it is necessary to fuse the evidence information by the rough evidence information. For samples with complex data distribution, the rough evidence information can improve the accuracy of classification.

B. A New Spark Parallel Attribute Reduction based on Neighborhood Rough Evidence Decision Error Rate

The running time of the attribute reduction algorithm depends mainly on the scale of the dataset and the number of its attributes, whose time complexity is $O(|C|^2 \bullet |U|^2)$ ($|C|$ is the number of conditional attributes and $|U|$ is the number of samples in the decision information system). It is obvious that the attribute reduction algorithm not only has a high time cost but also suffers from low computational efficiency when dealing with large-scale datasets. The purpose of algorithm parallelism is to be able to execute multiple tasks or process different subsets of data in parallel, reducing the computational complexity and increasing the computational efficiency of the algorithm. Thus, we implement Spark parallel attribute reduction based on neighborhood rough evidence decision error rate , named SPAR-NREDER, which reduces the time complexity from $O(|C|^2 \bullet |U|^2)$ to $O(|C|^2 \bullet |U_i|^2)$ ($|U_i|$ is the number of samples in the decision information sub-system). The SPAR-NREDER evaluates the significance of attributes via the neighborhood rough evidence decision error rate which can more accurately reflect the misclassification of boundary region samples.

According to the definition of neighborhood rough evidence, the rough belief function of x_t is $RBel^t(\{X_q\}) = RM^t(\{X_q\})$, and then the predicted label of x_t is $l(x_t) = \arg\max_{\omega_k}(RBel^t(\{X_k\}))$. If the predicted label of x_t is the same as its true label, then $\lambda(l_i(x_t) \mid ND_i(x_t)) = 0$, otherwise $\lambda(l_i(x_t) \mid ND_i(x_t)) = 1$. The neighborhood rough evidence decision error rate (NREDER) is defined by

$$NREDER = \frac{1}{M}\sum\nolimits_{t=1}^{M} \lambda(l(x_t) \mid ND(x_t)) , \tag{27}$$

where M is the number of samples in S .

Following the process of attribute reduction, the significance of candidate attributes based on NREDER is calculated by

$$SIG(a,R,D) = NREDER_R(D) - NREDER_{R\cup\{a\}}(D) . \tag{28}$$

In SPAR-NREDER, we achieve parallelized computation of the equivalence classes and the significance of candidate attributes, which reduces the time cost consumed by iterative computation in the attribute reduction process and improves the computational efficiency of the attribute reduction algorithm. The pseudocode of SPAR-NREDER is shown in Algorithm 1. Its main steps are mentioned as follows.

First, Algorithm 1 sends the collection of decision information sub-systems $\{S_1, S_2, \ldots, S_i, \ldots, S_m\}$ to the corresponding slave node $slave_i$.

Second, Algorithm 1 computes decision equivalence classes and neighborhood equivalence classes in different decision information sub-system S_i by parallel.

Third, Algorithm 1 calculates the significance of candidate attributes in different decision information sub-system S_i via parallel mechanism, and adds the conditional attributes with the largest value of attribute significance to R_i , and then computes the NREDER of the attribute reduction subset R_i until it satisfies $NREDER_{R_i} \le NREDER_c$.

Fourth, Algorithm 1 outputs the collection of attribute reduction subset $\{R_1, R_2, \ldots, R_m\}$ at each slave node.

To illustrate the NREDER-based attribute reduction process more clearly, the specific computation process on the decision information sub-system is given in the following Example 2.

Algorithm 1 (SPAR-NREDER): Spark parallel attribute reduction based on neighborhood rough evidence decision error rate

Input: Decision information sub-systems $\{S_1, S_2, \ldots, S_m\}$, neighborhood radius δ .

Output: Attribute reduction subsets $\{R_1, R_2, \ldots, R_m\}$.

```
1 :  Let $\{R_1, R_2, \ldots, R_m\} \leftarrow \varnothing$ .
2 :  For each $S_i$ partition, $i \in [1,2,\ldots,m]$ do
3 :      $Decision\_Class \leftarrow S_i.select('Class').rdd$
                  $.zip(S_i.select('Index')).rdd$
                  $.reduceByKey(lambda\ x,y:\ x+y).collect()$ .
4 :      Compute $Distance\_Matrix$ .
5 :      $Neighbor\_Class \leftarrow sc.parallelize(Distance\_Matrix)$
                  $.filter(lambdax: x < \delta).collect()$ .
6 :      Compute $NREDER_C$ for the conditional attributes $C$ .
7 :      For each attribute $a_j \in C - R_i$ do
8 :          Calculate the attribute significance $SIG(a_j, R_i, D)$ .
9 :      End
10:      Do
11:          Select the best candidate attribute $a_k$ .
12:          If $SIG(a_k, R_i, D) > 0$ then
13:              Add the candidate attribute $a_k$ to the attribute reduction
                 subsets $R_i$ .
14:          End If
15:          Calculate $NREDER_{R_i}$ for the attribute reduction subsets $R_i$ .
16:      Until $NREDER_{R_i} \le NREDER_C$
17:  End
18:  Output $\{R_1, R_2, \ldots, R_m\}$ .
```

Example 2: A decision information sub-system $S_1 = < U_1, C \cup D, V, f >$ is presented in Table I, where $\{a_1, a_2, a_3, a_4, a_5\}$ are the set of conditional attributes and d is the decision attribute. According to the calculation steps in the third line of Algorithm 1, we compute the decision equivalence classes $U_1 / D = \{X_1, X_2\} = \{\{x_2, x_3, x_4, x_8\}, \{x_1, x_5, x_6, x_7, x_9\}\}$ on S_1 . The distance matrix is computed as follows:

$$Distance\_Matrix = \begin{bmatrix} 0 & 0.6128 & \cdots & 0.6812 & 0.3134 \\ 0.6128 & 0 & \cdots & 0.4532 & 0.8984 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0.6812 & 0.4532 & \cdots & 0 & 0.9855 \\ 0.3134 & 0.8984 & \cdots & 0.9855 & 0 \end{bmatrix} .$$

We compute the neighborhood equivalence classes. For sample x_1 , its neighborhood is $\delta(x_1) = \{x_7\}$. Then, we compute the $NREDER_C$ for the set of conditional attributes C . According to Eq. (27), we have $NREDER_C = 0$.

In the process of searching for candidate attributes, we will iteratively calculate the significance of each candidate attribute and select the candidate attribute with the greatest attribute significance in turn until the attribute reduction subset satisfies the stopping criterion.

For the first iteration, we calculate the significance for each conditional attribute. We have $SIG(a_1, R_1, D) = 0.11$, $SIG(a_2, R_1, D) = 0.11$, $SIG(a_3, R_1, D) = 0$, $SIG(a_4, R_1, D) = 0$, and $SIG(a_5, R_1, D) = 0.11$. We select the candidate attribute a_1 and append it to the attribute reduction subset $R_1 = R_1 \cup a_1$. We compute the $NREDER_{R_1}$ for the attribute reduction subset R_1 and have $NREDER_{R_1} = 0.11$.

For the second iteration, we have $SIG(a_2, R_1, D) = 0.11$, $SIG(a_3, R_1, D) = -0.11$, $SIG(a_4, R_1, D) = -0.11$, and $SIG(a_5, R_1, D) = 0$. We select the candidate attribute a_2 and append it to the attribute reduction subset $R_1 = R_1 \bigcup a_2$. We compute the $NREDER_{R_1}$ for the attribute reduction subset R_1 and have $NREDER_{R_1} = 0$. Because of $NREDER_{R_1} \leq NREDER_C$, we stopped the attribute search process and obtained the final attribute reduction subset $R_1 = \{a_1, a_2\}$.

TABLE I

A DECISION INFORMATION SUB-SYSTEM S_1

U_1	a_1	a_2	a_3	a_4	a_5	D
x_1	0.35	0.74	0.59	0.50	0.48	1
x_2	0.06	0.43	0.54	0.39	0.17	0
x_3	0.29	0.58	0.60	0.38	0.15	0
x_4	0.23	0.55	0.75	0.56	0.15	0
x_5	0.47	0.92	0.52	0.35	0.18	1
x_6	0.65	0.72	0.77	0.55	0.50	1
x_7	0.41	0.74	0.62	0.59	0.37	1
x_8	0.05	0.51	0.25	0.65	0.20	0
x_9	0.47	0.88	0.74	0.50	0.62	1

C. A Novel Spark-based Rough Evidence Neighborhood Classification

To improve the decision-making capability of the neighborhood classifier and reduce the conflicts between the local evidence information with different labels in the evidence fusion process, we introduce rough evidence information into the process of neighborhood classification. In addition, neighborhood classifier suffers from high computational time-consumption and inefficiency when dealing with large-scale datasets, whose time complexity is $O(|U_{train}| \bullet |U_{test}|)$ ($|U_{train}|$ is the number of training samples and $|U_{test}|$ is the number of testing samples). Therefore, we propose a novel Spark-based rough evidence neighborhood classification, called SRENEC, which reduces the time complexity from $O(|U_{train}| \bullet |U_{test}|)$ to $O(|U^i_{train}| \bullet |U_{test}|)$ ($|U^i_{train}|$ is the number of training samples in the decision information sub-system). The SRENEC is able to fuse rough evidence information of neighborhood samples to improve the classification performance of the neighborhood classifier and reduce its computation time. The pseudocode of SRENEC is shown in Algorithm 2. Its main steps are mentioned as follows.

First, Algorithm 2 updates the decision information sub-system S_i by R_i to obtain the new decision information sub-system $S_i^{'} =< U_i, R_i \bigcup D, V_a, f >$ in each slave node, and then broadcast the unclassified sample x_t to all slave nodes.

Second, Algorithm 2 computes the rough evidence information to get the rough belief value $RBel^t$ in each slave node, and then obtains the predicted label $l_i(x_t)$ of x_t in the corresponding slave node $slave_i$.

Third, Algorithm 2 aggregates the predicted label of unclassified sample x_t on all slave nodes to get the final prediction label and outputs it.

Algorithm 2 (SRENEC): Spark-based rough evidence neighborhood classification

Input: Decision information system $S = \langle U, C \bigcup D, V_a, f \rangle$, neighborhood radius δ, and unclassified sample x_t.

Output: The predicted category label $l(x_t)$ of x_t.

```
Compute new decision information sub-systems $\{S_1, S_2, \ldots, S_m\}$ by Algorithm 3.
Calculate attribute reduction subspaces $\{R_1, R_2, \ldots, R_m\}$ by Algorithm 1.
Broadcast $x_t$ to each slaver node $x_t \leftarrow sc.broadcast(x_t)$.
For each $S_i$ partition, $i \in [1, 2, \ldots, m]$ do
    Update the sub-system $S_i$ by $R_i$.
    Compute decision equivalence class $\pi_i = \{X_1^i, X_2^i, \ldots, X_d^i\}$.
    Calculate the neighborhood space $\delta_{R_i}(x_t)$ of $x_t$.
    For each sample pair $(x_j, X_q^i)$, $x_j \in \delta_{R_i}(x_t)$ and $X_q^i \in \pi_i$ do
        Compute $M^{t,j}(\{X_q^i\})$ and $M^{t,j}(\pi_i)$, respectively, and they are computed by
            $M^{t,j}(\{X_q^i\}) = \beta_0 \cdot \exp\{-\gamma_q \cdot (dis(x_t, x_j))^2\}$,   (29)
            $M^{t,j}(\pi_i) = 1 - M^{t,j}(\{X_q^i\})$,   (30)
        where $0 < \beta_0 < 1$, $\gamma_q > 0$.
    End
    Merge the evidence information provided by samples with the same label to obtain the local evidence information $M_q^t(\{X_q^i\})$ and $M_q^t(\pi_i)$, and they are redefined by
        $M_q^t(\{X_q^i\}) = 1 - \prod_{x_j \in \delta_{R_i}^q(x_t)} M^{t,j}(\pi_i)$,   (31)
        $M_q^t(\{\pi_i\}) = \prod_{x_j \in \delta_{R_i}^q(x_t)} M^{t,j}(\pi_i)$.   (32)
    Compute the local rough evidence information $RM_q^t(\{X_q^i\})$ and $RM_q^t(\pi_i)$, and they are is obtained by
        $RM_q^t(\{X_q^i\}) = \mu_{X_q^i}^{\delta} \cdot M_q^t\left(\left\{X_q^i\right\}\right)$,   (33)
        $RM_q^t(\{\pi_i\}) = 1 - RM_q^t(\{X_q^i\})$.   (34)
    Aggregate the local rough evidence information with different labels to get the global rough evidence information $RM^t(\{X_q^i\})$ and $RM^t(\pi_i)$, and they are is calculated by
        $RM^t(\{X_q^i\}) = \frac{RM_q^t(\{X_q^i\}) \prod_{r \neq q} RM_r^t(\pi_i)}{K} \quad q = 1, 2, \ldots, d$,   (35)
        $RM^t(\pi_i) = \frac{\prod_{q=1}^{d} RM_q^t(\pi_i)}{K}$,   (36)
    where $K$ is the normalization factor, which is computed by
        $K = \sum_{q=1}^{d} RM_q^t(\{X_q^i\}) \cdot \prod_{r \neq q} RM_r^t(\pi_i) + \prod_{q=1}^{d} RM_q^t(\pi_i)$.   (37)
    The rough belief function $RBel^t(\{X_q^i\})$ is generated by fusing global rough evidence information, and it is given by
        $RBel^t(\{X_q^i\}) = RM^t(\{X_q^i\})$.   (38)
    In $S_i$, the predicted label of unclassified sample $x_t$ is calculated by
```

```
            $l_i(x_t) = \arg\max_{\omega_{i_k}}(Bel^t(\{X_k\}))$ .                (39)
End
Aggregate the collection of predicted labels $l_i(x_t)$ from different partitions to get the final prediction label which is defined by
            $l(x_t) = \arg\max_{\omega_{i_k}}(\sum_{\omega_{j_t}\in\{\omega_{1_t},\omega_{2_t},\ldots,\omega_{m_t}\}} H(\omega_{j_t},\omega_{k_t}))$ ,        (40)
where $H(\omega_{j_t},\omega_{k_t}) = \begin{cases} 1, & \omega_{j_t} = \omega_{k_t} \\ 0, & otherwise \end{cases}$ is the judge function.
Output the predicted label $l(x_t)$ of unclassified sample $x_t$ .
```

To illustrate the computational process of neighborhood decision making based on rough evidence more clearly, the following Example 3 presents the specific computational results on the decision information subsystem.

Example 3: According to the Example 2, the decision information sub-system S_1 is updated to $S_1^{'} = < U_1, R_1 \bigcup D, V, f >$. For a unclassified sample $x_{10} = \{0.12, 0.71, 0.67, 0.36, 0\}$, its neighborhood is $\delta(x_{10}) = \{x_1, x_3, x_4, x_8\}$.

According to Eq. (29), we compute the evidence information provide by its neighbors. We have $M^{10,1}(\{X_1^1\}) = 0.8307$, $M^{10,3}(\{X_0^1\}) = 0.8559$, $M^{10,4}(\{X_0^1\}) = 0.8719$, and $M^{10,8}(\{X_0^1\}) = 0.8577$.

The local evidence information $M_1^{10}(\{X_1^1\}) = 0.8307$ and $M_0^{10}(\{X_0^1\}) = 0.9974$ is obtained by Eq. (31).

According to the distribution of category labels in the neighborhood of unclassified sample x_{10} , its rough membership is $\mu_{X_0^1}^{\delta} = 0.75$ and $\mu_{X_1^1}^{\delta} = 0.25$, respectively. According to Eq. (33), we compute the local rough evidence information. We have $RM_1^{10}(\{X_1^1\}) = 0.2076$ and $RM_0^{10}(\{X_0^1\}) = 0.7480$.

The global rough evidence information $RM^{10}(\{X_1^1\}) = 0.0619$ and $RM^{10}(\{X_0^1\}) = 0.7016$ is calculated by Eq. (35).

The rough belief function of the unclassified sample x_{10} is $RBel^{10}(\{X_1^1\}) = 0.0619$ and $RBel^{10}(\{X_0^1\}) = 0.7016$.

Therefore, we assign the category label 0 with the maximum rough belief value to the unclassified sample x_{10} .

IV. Experimental Analysis

In the section, the objective of the experiments are conducted to evaluate the feasibility and effectiveness of the proposed method in six UCI imbalanced datasets. Furthermore, to verify the performance of algorithm parallelization in terms of speedup and sizeup, we conduct a series of experiments on three large-scale datasets. All the experiments are conducted on a personal computer with Windows 10， Intel (R) Core (TM) i9-128900K CPU@3.19GHz and RAM 64GB. The integrated development environment is JetBrains PyCharm 2020.3.5 and the programming language used is Python. We build the virtual environment of Hadoop-2.7.1 and Spark-2.4.6-preview on Windows 10 system, and the cluster mode is set to “local”. This experiment is compiled by the Python API in Spark, and manages 16 CPU cores and 24 logical processors by setting the parameter in the “local” mode to “local [*]”.

A. *Small data analysis*

1) *Experimental Data Pre-processing*

Regarding the public datasets, we selected six medical datasets from the UCI database with different IR. Table II depicts the relevant statistical information of these six imbalanced datasets.

In order to reduce the imbalance rate (called IR) of the dataset, this paper performs parallel randomly under sampling and expands the data size via parallel computing mechanism of Spark. First, the data of the same size as the negative class is generated by parallel random undersampling from the positive class. Then, the subset of positive class is reconstructed with the negative class to obtain multiple new data subsets. The pseudo-code for parallel random undersampling is discussed in Algorithm 3.

TABLE II
Dataset Description

ID	Dataset	Samples	Attributes	Classes	IR
1	WPBC	194	34	2	3.2
2	ZAS	303	22	2	2.5
3	WDBC	569	31	2	1.7
4	ILP	579	11	2	2.5
5	PIMA	768	9	2	1.9
6	DRD	1,151	20	2	1.1

Definition 13: In a decision information system $S = < U, C \bigcup D, V, f >$, where $U = \{x_1, x_2, \ldots, x_M\}$. Assume that the collection of the positive samples and the negative samples are $U^P = \{y_1, y_2, \ldots, y_P\}$ and $U^N = \{z_1, z_2, \ldots, z_N\}$ respectively, and $P + N = M$. The collection of new decision information sub-systems $\{S_1, S_2, \ldots, S_m\}$ by the parallel random undersampling can be defined as:

$$S_i = \langle U_i, C \bigcup D, V, f \rangle, \tag{41}$$

where $U_i = U_i^P \bigcup U^N$, $U_i^P = \{y_j \in U^P \mid j = Rand(1, P)\}$, and $|U_i^P| = N$.

Algorithm 3 (PRUS): Parallel random undersampling algorithm

Input: Decision information system $S = \langle U, C \bigcup D, V, f \rangle$.

Output: Decision information sub-systems $\{S_1, S_2, \ldots, S_m\}$.

```
1: Convert the data to $S \leftarrow spark.read.csv(path)$ via Spark.
2: $U^N = S.filter(S['label'] == negative\_label)$ .
3: Generate $S_i^{'}$ based on the $m$ partitions.
4: For each $S_i^{'}$ partition, $i \in [1, 2, \ldots, m]$ do
5:     According to Eq. (41), $U_i^P = S_i^{'}.filter(S_i^{'}['label'] == positive\_label).rdd.takesample$ .
6:     $S_i = U_i^P.union(U^N)$ .
7: End
8: return $\{S_1, S_2, \ldots, S_m\}$ .
```

According to Def. 13, a multi-granular sample space based on Spark is constructed using parallel random undersampling algorithm. In Algorithm 3, the original decision information system is loaded by *spark.read.csv*() (line1). Then the algorithm filters out a subset of negative class samples (line2).

After that, a multi-granular sample space $\{S_1, S_2, \ldots, S_m\}$ can be generated by merging a subset of negative samples with a subset of positive samples (line3~7). Finally, Algorithm 3 outputs a multi-granular sample space $\{S_1, S_2, \ldots, S_m\}$ (line8).

In this experiment, we use the hold-out method to divide the raw dataset into training and test data at the ratio of 4:1. The training data is preprocessed by Algorithm 3, and the new subset with IR of 1 is reconstructed. Meanwhile, the data scale is expanded and reorganized into 15 subsets by using the parallelize operator in Spark. Table III describes the relevant statistical information related to the preprocessed dataset.

TABLE III
PREPROCESSED DATASET DESCRIPTION

ID	Dataset	Samples	Scale of Data Expansion
1	WPBC	1,380	7.11
2	ZAS	2,610	8.61
3	WDBC	6,360	11.18
4	ILP	4,950	8.55
5	PIMA	8,040	10.47
6	DRD	16,200	14.07

2) *The performance of SPAR-NREDER algorithm*

To examine the performance of AR-NREDER and SPAR-NREDER evaluation functions, this section presents two comparative experimental results based on six UCI imbalance datasets. Firstly, we compare the running time of the attribute reduction algorithms, including AR-NREDER and SPAR-NREDER. Secondly, we compare the classification performance of different attribute reduction subsets on neighborhood classifiers (called NEC), K-NN and Logistic Regression classifiers (called LR). Fig. 3 indicates the performance of AR-NREDER and SPAR-NREDER in terms of running time and the efficiency. Table IV demonstrates the classification performance of three attribute reduction subsets on three classifiers.

In Fig. 3, it is obviously that SPAR-NREDER has far less running time than AR-NREDER, which indicates that SPAR-NREDER can effectively reduce the running time of AR-NREDER. For example, in the WDBC dataset, the running time of SPAR-NREDER is much less than that of AR-NREDER. The red line in Fig. 3 shows a trend that as the scale of datasets and the number of its conditional attribute grow, both the reduced running time and the improved computational efficiency of SPAR-NREDER also increase. The experimental results show that the parallel attribute reduction algorithm can dramatically reduce the running time and improve the computational efficiency of the attribute reduction algorithm with excellent performance.

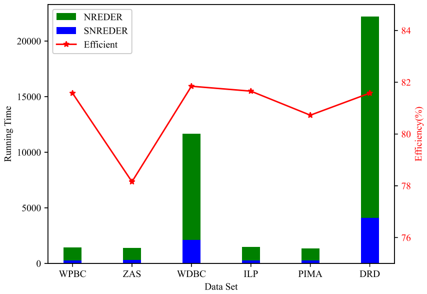


Fig. 3. The running time and the efficient of AR-NREDER and SPAR-NREDER.

As can be seen from Table IV, SPAR-NREDER has higher classification performance on NEC and K-NN classifiers for the majority of datasets. For NEC, the classification performance of SPAR-NREDER is lower than AR-NREDER only on the DRD dataset, and is better than the others attribute reduction on the rest of the datasets. For K-NN, the classification performance of SPAR-NREDER is lower than AR-NREDER on the DRD and ILP datasets, and is better than the others attribute reduction on the rest of the datasets. Although SPAR-NREDER performs poorly on LR, its classification performance for all classifiers on the ZAS dataset outperforms the others attribute reduction. Thus, SPAR-NREDER not only reduces the computational time of the attribute reduction, but also improves the performance of the attribute reduction subset.

3) *The performance of SRENEC algorithm*

To examine the performance of the algorithm SRENEC in terms of computation time and classification, this section presents comparative experiments based on six UCI imbalance datasets. The neighborhood classification algorithm (called NEC) [23], the evidence neighborhood classification algorithm (called ENEC) [27], the rough evidence neighborhood classification algorithm (called RENEC), and the Spark-based rough evidence neighborhood classification (called SRENEC). The running time and the efficient of RENEC and SRENEC with the same data size are given in Fig. 4.

In Fig. 4, it is clearly that SRENEC has far less running time than RENEC, which indicates that SRENEC can effectively reduce the running time of RENEC. For example, in the DRD dataset, the running time of SRENEC is much less than that of RENEC. Compared to the parallel attribute reduction algorithm, the parallel classification algorithm improves the most computational efficiency on the DRD dataset rather than WDBC dataset, because the DRD dataset has the largest size. The red line in Fig. 4 shows a trend that as the scale of datasets and the number of its conditional attribute grow, both the reduced running time and the improved computational efficiency of SRENEC also increase. The experimental results show that the parallel classification algorithm can dramatically reduce the running time and improve the computational efficiency of the classification algorithm with excellent performance.

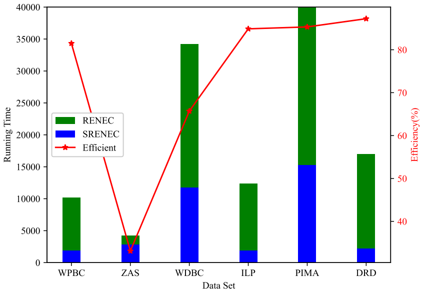


Fig. 4. The running time and the efficient of RENEC and SRENEC.

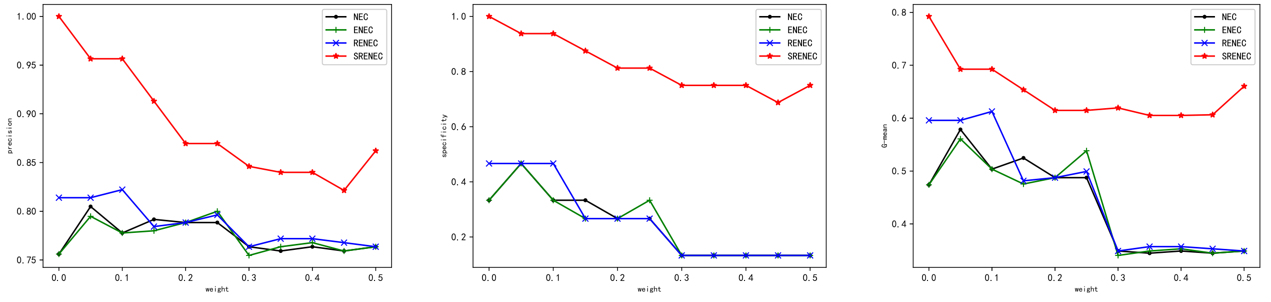
Then, we compare the classification performance of four classifiers, NEC, ENEC, RENEC and SRENEC, with three classification evaluation metrics of precision, specificity and G-mean. The neighborhood radius is formulated as $\delta = \min(dis_B^R(x_i, x_j)) + \omega \cdot (\max(dis_B^R(x_i, x_j)) - \min(dis_B^R(x_i, x_j)))$, and

TABLE IV
CLASSIFICATION ACCURACY COMPARISONS WITH THREE ATTRIBUTE REDUCTIONS

Dataset	NEC			K-NN			LR		
	AR-NDER	AR-NREDER	SPAR-NREDER	AR-NDER	AR-NREDER	SPAR-NREDER	AR-NDER	AR-NREDER	SPAR-NREDER
WPBC	0.6154	**0.6410**	**0.6410**	0.6154	**0.6410**	**0.6410**	**0.7949**	0.7436	0.7436
ZAS	0.6393	0.6557	**0.6721**	0.7049	0.6393	**0.7377**	0.7705	0.7049	**0.7869**
WDBC	0.9211	0.9298	**0.9386**	0.9298	**0.9386**	**0.9386**	**0.9386**	0.9298	0.9211
ILP	0.6638	0.6810	**0.6897**	**0.6983**	0.6552	0.6466	0.7155	0.7241	**0.7414**
PIMA	0.6364	0.6364	**0.6753**	0.6818	0.6818	**0.7143**	**0.7857**	**0.7857**	0.7403
DRD	0.5974	**0.6017**	0.5974	0.5974	**0.6104**	0.5974	0.7056	**0.7100**	0.7056

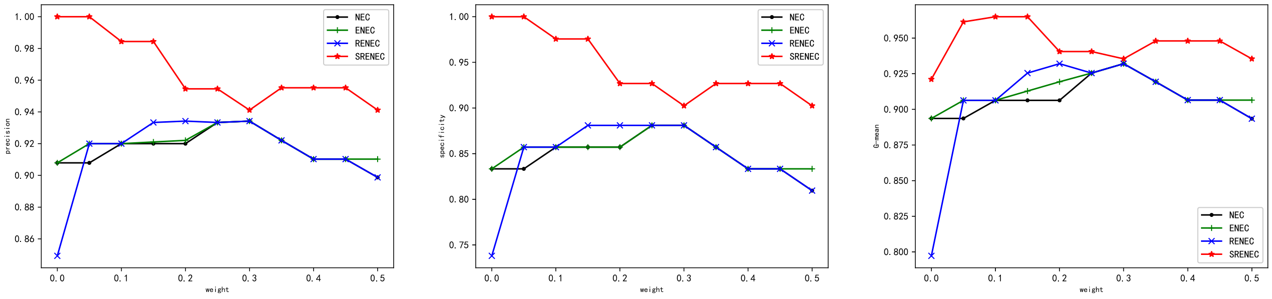
TABLE V
CLASSIFICATION COMPARISONS WITH FOUR CLASSIFIERS

Dataset	Precision				Specificity				G-means			
	NEC	ENEC	RENEC	SRENEC	NEC	ENEC	RENEC	SRENEC	NEC	ENEC	RENEC	SRENEC
WPBC	0.7241	0.7241	0.7500	**1**	0.2727	0.2727	0.3636	**1**	0.4523	0.4523	0.5222	**0.7958**
ZAS	0.7778	0.7778	0.8222	**0.9565**	0.3333	0.3333	0.4667	**0.9375**	0.5036	0.5036	0.6127	**0.6926**
WDBC	0.9200	0.9200	0.9200	**0.9545**	0.8571	0.8571	0.8571	**0.9756**	0.9063	0.9063	0.9063	**0.9650**
ILP	0.7317	0.7529	0.7614	**0.9778**	0.3529	0.3824	0.3824	**0.9697**	0.5082	0.5463	0.5590	**0.7493**
PIMA	0.7414	0.7818	0.7876	**0.9114**	0.4545	0.5636	0.5636	**0.8511**	0.6284	0.6997	0.7118	**0.8161**
DRD	0.5833	0.5854	0.5746	**0.6923**	0.5536	0.5446	0.4911	**0.7838**	0.5706	0.5740	0.5637	**0.6538**



(a) Precision (b) Specificity (c) G-mean

Fig. 5. Classification comparisons with four classifiers in ZAS dataset.



(a) Precision (b) Specificity (c) G-mean

Fig. 6. Classification comparisons with four classifiers in WDBC dataset.

we setup the weight is $\omega = 0.1$. The results of the classification are shown in Table V. A comparison of the classification results with four classifiers on the ZAS and WDBC dataset for ten different neighborhood radius is illustrated in Figs. 5-6. It is not difficult to conclude from Table V that the classification performance of SRENEC is optimal. And the RENEC performs well on most datasets in terms of classification performance. The results of classification show that, in the majority of datasets, the rough evidence information can enhance the capability of neighborhood decision-making, to increase the accuracy of classification. What's more, the SRENEC constructs a multi-granularity sample space and characterizes the samples from different granularity by parallel undersampling methods, which obtains more accurate classification results. For example, in the WPBC dataset, when RENEC is carried out, the precision is increased from 0.7241 to 0.75, the specificity is enhanced from 0.2727 to 0.3636, and the G-mean is raised from 0.4523 to 0.5222. When SRENEC is undertaken, the precision is increased from 0.7241 to 1, the specificity is enhanced from 0.2727 to 1, and the G-mean is raised from 0.4523 to 0.7958. Its improvement ratio on the three classification evaluation metrics is 27.59%, 72.73%, 34.35% respectively. It can be seen that the specificity of SRENEC shows the highest improvement ratio. However, in the DRD dataset, the classification performance of RENEC is inferior to that of NEC and ENEC. In general, the classification performance of SRENEC and RENEC is better than that of others classifiers.

As can be seen from Figs. 5-6, there are a trend that as the weight of neighborhood radius increases, the classification performance of the four algorithms under three evaluation metrics first rises, then falls and finally stabilizes. Obviously, the classification performance of SRENEC is consistently better than that of the others three algorithms. For RENEC, in some cases, the classification performance of it is better than that of the other three algorithms.

B. Large-scale data analysis

To examine the advantages of SPAR-NREDER and SRENEC, three large-scale datasets are added to further confirm the feasibility and effectiveness of parallelized computation, and the relevant statistical information of the added large-scale datasets is provided in Table VI. During the experimental analysis, the feasibility and effectiveness of SPAR-NREDER and SRENEC are focused mainly in terms of two evaluation metrics of speedup and sizeup.

TABLE VI
THE DESCRIPTION OF LARGE-SCALE DATASET

ID	Dataset	Samples	Attributes	Classes
1	SUSY	5,000,000	19	2
2	HEPMASS	10,500,000	28	2
3	HIGGS	11,000,000	28	2

1) The speedup of parallel algorithm

The speedup ratio of parallel algorithm measures the running time of the parallel algorithm by increasing the number of partitions in experiment while ensuring a constant data size, and it is computed by

$$Speedup(p) = \frac{T_2}{T_p}, \qquad (42)$$

where p is the number of partitions, T_2 is the running time of on two partitions, T_p is the running time on p partitions.

TABLE VII
THE RUNNING TIME OF SPAR-NREDER WITH DIFFERENT PARTITION NUMBERS (/S)

Dataset	The number of partitions				
	2	4	6	8	10
SUSY	7,249,296	2,183,942	1,121,482	551,320	542,967
HEPMASS	9,127,176	3,399,322	1,926,980	1,389,426	1,247,552
HIGGS	18,047,257	5,878,546	3,265,961	2,646,643	2,082,147

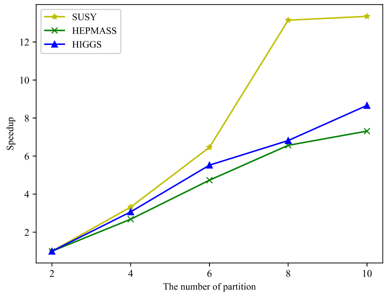


Fig. 7. The speedup of SPAR-NREDER.

In this experiment, the number of partitions is setup as $2, 4, 6, 8, 10$, and the running time consumed by the parallel algorithm is calculated on these partition numbers to compare the performance of the parallel algorithm in terms of speedup.

Table VII and Fig. 7 show the running time and the speedup ratio variation curves of the SPAR-NREDER for different number of partitions, respectively. And then Table VIII and Fig. 8 show the running time and the speedup ratio variation curves of the SRENEC algorithm for different number of partitions, respectively. It is not difficult to observe from Figs. 7-8 that as the number of partitions increases SPAR-NREDER and SRENEC take less time to process the same size dataset. However, for the SUSY dataset, the computational efficiency of SPAR-NREDER and SRENEC improvement significantly decreases when the number of partitions is increased from 8 to 10, which indicates that some partitions are note fully used in the parallel processing of SUSY. Based on the experimental results of speedup, it is not hard to conclude that our proposed parallel algorithms can effectively process large-scale datasets.

TABLE VIII
THE RUNNING TIME OF SRENEC WITH DIFFERENT PARTITION NUMBERS (/S)

Dataset	The number of partitions				
	2	4	6	8	10
SUSY	8,086,363	2,615,549	1,426,885	802,582	789,722
HEPMASS	10,548,241	4,170,785	2,464,827	1,829,452	1,712,422
HIGGS	19,742,276	6,743,357	3,875,210	3,757,569	2,588,363

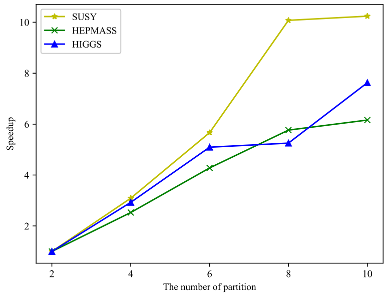


Fig. 8. The speedup of SRENEC.

2) The sizeup of parallel algorithm

The sizeup ratio of parallel algorithm shows the change in the running time of the parallel algorithm by increasing the size of datasets experiment while keeping the number of partitions, and it is computed by

$$Sizeup(D, p) = \frac{T_{D_1}}{T_{D_p}}, \qquad (43)$$

where D is one dataset, T_{D_1} is the running time for D, T_{D_p} is the running time for $p \times D$.

In this experiment, we set the number of partitions as 2, and then gradually grow the size of datasets from D to $8D$ by 2 times.

TABLE IX
THE RUNNING TIME OF SPAR-NREDER WITH DIFFERENT SCALE OF DATASETS (/S)

Dataset	The multiples of D			
	D	$2D$	$4D$	$8D$
SUSY	110,564	348,471	1,003,525	4,933,438
HEPMASS	433,928	1,055,153	2,556,667	6,544,484
HIGGS	519,733	1,329,162	4,633,389	11,566,334

Table IX and Fig. 9 show the running time and the sizeup ratio variation curves of the SPAR-NREDER for different size of datasets, respectively. And then Table X and Fig. 10 show the running time and the sizeup ratio variation curves of the SRENEC algorithm for different size of datasets, respectively. The sizeup ratio change curves in Figs. 9-10 show that when the size of datasets gradually grows, the running time of

SPAR-NREDER and SRENEC also increases. Therefore, according to the experimental results of sizeup, our proposed parallel algorithms have a good sizeup performance.

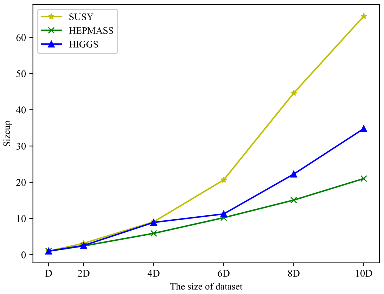


Fig. 9. The sizeup of SPAR-NREDER.

TABLE X
THE RUNNING TIME OF SRENEC WITH DIFFERENT SIZE OF DATASETS (/S)

Dataset	The multiples of D			
	D	$2D$	$4D$	$8D$
SUSY	121,332	385,060	1,135,904	5,451,248
HEPMASS	456,042	1,126,889	2,807,404	7,399,377
HIGGS	544,083	1,401,842	4,904,663	12,485,335

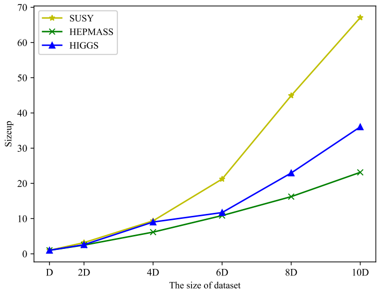


Fig. 10. The sizeup of SRENEC.

V. DISCUSSION

The traditional NEC algorithm hardly considers the spatial difference and label uncertainty of the neighborhood samples, which may increase the possibility of the misclassification. Compared with majority voting mechanism, the rough evidence can obtain the more precise decision-making information, which demonstrates that the rough evidence can improve the classification accuracy. The rough evidence is a novel decision rule using the rough membership function and D-S evidence theory. The rough membership function is applied to the global evidence information merging process, which enhances the local evidence information of the majority class samples in the neighborhood and reduces the degree of conflict between the local evidence information of the different classes of samples. Therefore, the rough evidence is more suitable for the neighborhood decision process than the majority voting mechanism and the D-S evidence.

In attribute reduction, the experimental results of NREDER show that the AR-NREDER is more precise than the AR-NDER in evaluating the attribute significance by the number of misclassified samples. Moreover, the experimental results of RENEC indicates that the rough evidence outperformed the majority voting mechanism and the D-S evidence in classification. Therefore, we conclude that the rough evidence is suitable for processing the data with complicated sample distributions and can significantly improve the neighborhood decision making ability.

For another challenge that the AR-NREDER and the RENEC is computationally inefficient handling the large-scale data, we have implemented SPAR-NREDER and SRENEC respectively by using the parallel computing advantage of Spark framework. First, the multi-granular sample space obtained by Spark-based parallel random undersampling can depict the overall sample space from different granularities, which can improve the algorithm performance more significantly. Then, the SPAR-NREDER can compute equivalence classes in parallel and parallelize the process of searching for candidate attributes on different samples subspaces. Finally, the SRENEC can compute the neighborhood space and rough evidence information in parallel on different attribute reduction subsets.

The running time and the efficient of SPAR-NREDER and SRENEC show that the Spark parallel computing framework can significantly improve the computational efficiency of attribute reduction and neighborhood classification algorithms. Furthermore, the classification accuracy results of SPAR-NREDER under different classifiers show that it not only improves the computational efficiency of attribute reduction, but also enhances the performance of attribute reduction subsets. And the classification results of SRENEC indicate that it not only improves the computational efficiency of neighborhood classification, but also strengthens the neighborhood decision making ability. In addition, experimental results of SPAR-NREDER and SRENEC algorithms on large-scale data demonstrate that it has favorable performance in terms of speedup and sizeup.

VI. CONCLUSIONS

Since there is a characteristic that the larger size of large-scale data, the lower value density, the neighborhood classifier cannot effectively deal with the classification problem of large-scale data. In this paper, partial improvements have been proposed to address the limitations of the neighborhood classifier. We present a novel Spark-based attribute reduction and neighborhood classification for rough evidence. We first introduce the rough membership function into the combination rule of D-S evidence theory to enhance the local evidence information of the majority class samples in the neighborhood and obtain more reliable evidence information. Second, a multi-granular sample space based on Spark is constructed using parallel random undersampling algorithm to reflect the distinction of universe in terms of multi-granularity. Third, a new Spark parallel attribute reduction based on neighborhood rough evidence decision error rate is proposed to more precisely evaluate attribute significance and improve the computational efficiency of attribute reduction. Finally, we propose a novel Spark-based rough evidence neighborhood classification, which can be fused by parallel the rough evidence information of intra-class samples and inter-class samples in the neighborhood. Experimental results on small datasets show that the classification accuracy of the attribute reduction subset obtained by the SPAR-NREDER parallel attribute algorithm is better than NDER and NREDER, and the classification performance of the SRENEC parallel classification algorithm is also superior to NEC, ENEC and RENEC, which indicates that the rough evidence can effectively improve the neighborhood decision making ability and reduce the evidence information provided by the outlier samples in the neighborhood. In additional, we verify that the proposed SPAR-NREDER and SRENEC parallel algorithms have favorable performance in terms of speedup and sizeup via experiment on large-scale datasets. Our future research work will focus on how to further optimize the search process of attribute reduction.

REFERENCES

[1] L. N. Zhou, S. S. Pan, J. W. Wang, et al. "Machine learning on big data: opportunities and challenges," *Neurocomputing*, vol. 237, pp. 350-361, May. 2017.

[2] A. Jaiswal, V. K. Dwivedi, O. P. Yadav. "Big Data and its Analyzing Tools: A Perspective," in *Proc. IEEE 6th ICACCS'20*, 2020, pp. 560-565.

[3] T. R. Rao, P. Mitra, R. Bhatt, et al. "The big data system, components, tools, and technologies: a survey," *Knowl. Inf. Syst.*, vol. 60, no. 3, pp. 1165-1245, Sept. 2019.

[4] S. J. Tang, B. S. He, C. Yu, et al. "A survey on spark ecosystem: Big data processing infrastructure, machine learning, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 71-91, Jan. 2022.

[5] X. H. Zhang, D. Q. Miao, C. H. Liu, et al. "Constructive methods of rough approximation operators and multigranulation rough sets," *Knowl. Based Syst.*, vol. 91, pp. 114-125, Jan. 2016.

[6] W. P. Ding, C. T. Lin, Z. H. Cao. "Deep neuro-cognitive co-evolution for fuzzy attribute reduction by quantum leaping PSO with nearest-neighbor memeplexes," *IEEE Trans. Cybern*., vol. 49, no. 7, pp. 2744-2757, July. 2019.

[7] W. P. Ding, J. D. Wang, J. H. Wang. "Multigranulation consensus fuzzy-rough based attribute reduction," *Knowl. Based Syst.*, vol. 198, pp. 105945, June. 2020.

[8] Y. Y. Yao. "Three-way granular computing, rough sets, and formal concept analysis," *Int. J. Approx. Reason.*, vol. 116, pp. 106-125, Jan. 2020.

[9] W. P. Ding, W. Pedrycz, I. Triguero, et al. "Multigranulation Supertrust Model for Attribute Reduction," *IEEE Trans. Fuzzy Sys.*, vol. 29, no. 7, pp. 1395-1408, June. 2021.

[10] C. Z. Wang, Y. Huang, M. W. Shao, et al. "Feature Selection Based on Neighborhood Self-Information," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 4031-4032, Sept. 2020.

[11] M. Hu, E. C. C. Tsang, Y. T. Guo, et al. "Fast and Robust Attribute Reduction Based on the Separability in Fuzzy Decision Systems," *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 5559-5572, June. 2022.

[12] T. R. Li, C. Luo, H. M. Chen, et al. "PICKT: a solution for big data analysis," in *Proc. RSKT*, 2015, pp. 15-25.

[13] J. Qian, L. Ping, X. D. Yue, et al. "Hierarchical attribute reduction algorithms for big data using MapReduce," *Knowl. Based Syst.*, vol. 73, pp. 18-31, Jan. 2015.

[14] J. Qian, M. Xia, X. D. Yue. "Parallel knowledge acquisition algorithms for big data using MapReduce," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 6, pp. 1007-1021, June. 2018.

[15] J. B. Zhang, J. S. Wong, T. R. Li, et al. "A comparison of parallel large-scale knowledge acquisition using rough set theory on different MapReduce runtime systems," *Int. J. Approx. Reason.*, vol. 55, no. 3, pp. 896-907, Mar. 2014.

[16] Q. H. Hu, W. Pedrycz, D. R. Yu, et al. "Selecting discrete and continuous features based on neighborhood decision error minimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 1, pp. 137-150, Feb. 2010.

[17] Y. H. Qian, X. Y. Liang, G. P. Lin, et al. "Local multigranulation decision-theoretic rough sets," *Int. J. Approx. Reason.*, vol. 82, pp. 119-137, Mar. 2017.

[18] X. D. Fan, W. D. Zhao, C. Z Wang, et al. "Attribute reduction based on max-decision neighborhood rough set model," *Knowl. Based Syst.*, vol. 151, pp. 16-23, Jul. 2018.

[19] Y. H. Qian, X. Y. Liang, Q. Wang, et al. "Local rough set: a solution to rough data analysis in big data," *Int. J. Approx. Reason.*, vol. 97, no. 38-63, June. 2018.

[20] Q. Wang, Y. H. Qian, X. Y. Liang, et al. "Local neighborhood rough set," *Knowl. Based Syst.*, vol. 153, pp. 53-64, Aug. 2018.

[21] G. P. Lin, Y. H. Qian, J. J. Li. "NMGRS: neighborhood-based multigranulation rough sets," Int. J. Approx. Reason., vol. 53, no. 7, pp. 1080-1093, Oct. 2012.

[22] Y. Z. Li, M. J. Cai, J. Zhou, et al. "Accelerated multi-granularity reduction based on neighborhood rough sets," *Appl. Intel.*, pp. 1-16, Apr. 2022.

[23] Q. H. Hu, D. Yu, Z. X. Xie. "Neighborhood classifiers," *Expert Syst. with Appl.*, vol. 34, no. 2, pp. 866-876, Feb. 2008.

[24] Ö. F. Ertuğrul, M. E. Tağluk. "A novel version of k nearest neighbor: dependent nearest neighbor," *Appl. Soft Comput.*, vol. 55, pp. 480-490, June. 2017.

[25] W. Li, Y. M. Chen, Y. P. Song. "Boosted K-nearest neighbor classifiers based on fuzzy granules," *Knowl. Based Syst.*, vol. 195, pp. 105606, May. 2020.

[26] Z. B. Pan, Y. D. Wang, W. P. Ku. "A new general nearest neighbor classification based on the mutual neighborhood information," *Knowl. Based Syst.*, vol. 121, pp. 142-152, April. 2017.

[27] T. Denoeux. "A k-nearest neighbor classification rule based on Dempster-Shafer theory," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 5, pp. 804-813, May. 1995.

[28] Y. Q. Zhang, G. Cao, B. S. Wang, et al. "A novel ensemble method for k-nearest neighbor," *Pattern Recogn.*, vol. 85, pp. 13-25, Jan. 2019.

[29] Z. G. Liu, Q. Pan, J. Dezert. "Classification of uncertain and imprecise data based on evidence theory," *Neurocomputing*, vol. 133, pp. 459-470, June. 2014.

[30] C. F. Lian, S. Ruan, T. Denoeux. "An evidential classifier based on feature selection and two-step classification strategy," *Pattern Recognition*, vol. 48, no. 7, pp. 2318-2327, July, 2015.

[31] H. C. Lu, F. J. Hwang, Y. H. Huang. "Parallel and distributed architecture of genetic algorithm on Apache Hadoop and Spark," *Appl. Soft Comput.*, vol. 95, pp. 106497, Oct. 2020.

[32] L. Z. Yin, L. Y. Qin, Z. H. Jiang, et al. "A fast parallel attribute reduction algorithm using Apache Spark," *Knowl. Based Syst.*, vol. 212, pp. 106582, Jan. 2021.

[33] C. Luo, S. Z. Wang, T. R. Li, et al. (2021, Sept). "Spark rough hypercuboid approach for scalable feature selection," *IEEE Trans. Knowl. Data Eng.* [Online]. pp. 1. Available: https://ieeexplore.ieee.org/abstract/document/9537667.

[34] C. Luo, S. Z. Wang, T. R. Li, et al. (2022, May). "Large-Scale Meta-Heuristic Feature Selection Based on BPSO Assisted Rough Hypercuboid Approach," *IEEE Trans. Neur. Net. LEAR.* [Online]. pp. 1-15. Available: https://ieeexplore.ieee.org/abstract/document/9773310.

[35] H. R. Ju, W. P. Ding, X. B. Yang, et al. "Robust supervised rough granular description model with the principle of justifiable granularity," *Appl. Soft. Comput.,* vol. 110, pp. 107612, Oct. 2021.

[36] M. M. Khan, M. A. U. Alam, A. K. Nath, et al. "Exploration of memory hybridization for RDD caching in Spark," in *Proc. ACM SIGPLAN ISMM*, 2019, pp. 41-52.

[37] M. Dessokey, S. M. Saif, S. Salem, et al. "Memory Management Approaches in Apache Spark: A Review" in *AISI*, 2020, pp. 394-403.

[38] L. Baldacci, M. Golfarelli. "A cost model for Spark SQL," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 5, pp. 819-832, May. 2018.

[39] D. Z. Cheng, X. B. Zhou, Y. Wang, et al. "Adaptive scheduling parallel jobs with dynamic batching in spark streaming," *IEEE Trans. Parall. Distr. Syst.*, vol. 29, no. 12, pp. 2672-2685, Dec. 2018.

[40] X. Meng, J. Bradley, B. Yavuz, et al. "Mllib: Machine learning in apache spark," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1235-1241, Jan. 2016.

[41] Z. Tang, M. S. He, Z. M. Fu, et al. "IncGraph: An improved distributed incremental graph computing model and framework based on spark graphX," *IEEE Trans. Knowl. Data Eng.*, early access, Aug. 2020.

[42] Z. M. Fu, Z. Tang. "Optimizing speculative execution in spark heterogeneous environments," *IEEE Trans. on Cloud Comput.*, vol. 10, no. 1, pp. 568-582, Jan. 2019.

Weiping Ding (M'16-SM'19) received the Ph.D. degree in Computer Science, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2013. From 2014 to 2015, he is a Postdoctoral Researcher at the Brain Research Center, National Chiao Tung University (NCTU), Hsinchu, Taiwan. In 2016, He was a Visiting Scholar at National University of Singapore (NUS), Singapore. From 2017 to 2018, he was a Visiting Professor at University of Technology Sydney (UTS), Ultimo, NSW, Australia. He is currently a professor of the School of

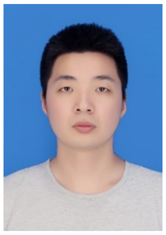
Information Science and Technology, Nantong University, Nantong, China, and also the supervisor of Ph.D postgraduate by City University of Macau. His research interests include deep neural networks, multimodal machine learning, granular data mining, and medical images analysis. He has published over 150 scientific articles in refereed international journals, such as IEEE T-FS, T-NNLS, T-CYB, T-SMCS, T-BME, T-EVC, T-II, T-ETCI, T-CDS, T-ITS and T-AI. He has co-authored two books. His ten authored/co-authored papers have been selected as ESI Highly Cited Papers. Dr. Ding served/serves on the Editorial Board of Knowledge-Based Systems, Information Fusion, Engineering Applications of Artificial Intelligence and Applied Soft Computing. He served/serves as an Associate Editor of IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Fuzzy Systems, IEEE/CAA Journal of Automatica Sinica, Information Sciences, Neurocomputing, Swarm and Evolutionary Computation, and so on.

Li Ming, born in 1996. Postgraduate. His main research interests include Data mining, Granular Computing and Big data analytics.

Sun Ying, born in 1997. Postgraduate. Her main research interests include Granular Computing, Rough Set and Deep Learning.

Jun Liu 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. He has been working in the field of Artificial Intelligence for many years. His current research is focused on two themes: 1) knowledge-centralized data analytics under uncertainty for sensing decision making, with applications in management, engineering, and industry field etc. (e.g., safety and risk analysis; policy decision making; security/disaster management; and heath care and smart home); 2) logic and automated reasoning methods for intelligent systems. He has over 200 publications in these areas.

Hengrong Ju received the B.Sc. and M.Sc. degrees in Computer Science and Technology from Jiangsu University of Science and Technology in 2012 and 2015, respectively, and the Ph.D. degree in Management Science and Engineering from Nanjing University in 2019. From 2017 to 2018, he worked as a visiting scholar at the Department of Electrical and Computer Engineering, University of Alberta. He is an associate professor with School of Information Science and Technology at Nantong University. He has authored or co-authored more than 20 scientific papers in international journals and conferences. His current research interests include knowledge discovery and granular computing.



Jiashuang Huang received his Ph.D. degree in College of Computer Science and Technology from Nanjing University of Aeronautics and Astronautics in 2020. From 2018 to 2019, he was a Visiting Scholar at University of Wollongong (UOW), Wollongong , NSW, Australia. He is a lecturer with the School of Information Science and Technology at Nantong University. His recent research is to analyze the brain network by using machine learning methods. He has published more than 20 research peer-reviewed journal and conference papers, including IEEE TMI, IEEE JBHI, Medical Image Analysis, Bioinformatics, AAAI, MICCAI etc.

Chin-Teng Lin received a Bachelor's of Science from National Chiao-Tung University (NCTU), Taiwan in 1986, and holds Master's and PhD degrees in Electrical Engineering from Purdue University, USA, received in 1989 and 1992, respectively. He is currently a distinguished professor and Co-Director of the Australian Artificial Intelligence Institute within the Faculty of Engineering and Information Technology at the University of Technology Sydney, Australia. He is also an Honorary Chair Professor of Electrical and Computer Engineering at NCTU. For his contributions to biologically inspired information systems, Prof Lin was awarded Fellowship with the IEEE in 2005, and with the International Fuzzy Systems Association (IFSA) in 2012. He received the IEEE Fuzzy Systems Pioneer Award in 2017. He has held notable positions as editor-in-chief of IEEE Transactions on Fuzzy Systems from 2011 to 2016; seats on Board of Governors for the IEEE Circuits and Systems (CAS) Society (2005-2008), IEEE Systems, Man, Cybernetics (SMC) Society (2003-2005), IEEE Computational Intelligence Society (2008-2010); Chair of the IEEE Taipei Section (2009-2010); Distinguished Lecturer with the IEEE CAS Society (2003-2005) and the CIS Society (2015-2017); Chair of the IEEE CIS Distinguished Lecturer Program Committee (2018-2019); Deputy Editor-in-Chief of IEEE Transactions on Circuits and Systems-II (2006-2008); Program Chair of the IEEE International Conference on Systems, Man, and Cybernetics (2005); and General Chair of the 2011 IEEE International Conference on Fuzzy Systems. Prof Lin is the co-author of Neural Fuzzy Systems (Prentice-Hall) and the author of Neural Fuzzy Control Systems with Structure and Parameter Learning (World Scientific). He has published more than 380 journal papers including over 170 IEEE journal papers in the areas of neural networks, fuzzy systems, brain-computer interface, multimedia information processing, cognitive neuro-engineering, and human-machine teaming, that have been cited more than 26,980 times. Currently, his h-index is 75, and his i10-index is 339.