Random Fuzzy Clustering Granular Hyperplane Classifier

WEI LI\(^1\), KAIQIANG ZHANG\(^1\), YUMIN CHEN\(^2\), (Member, IEEE), CHAO TANG\(^2\), XIAOYU MA\(^1\), AND YOUMENG LUO\(^1\)

\(^1\)School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China
\(^2\)School of Artificial Intelligence and Big Data, Hefei University, Hefei 230601, China

Corresponding author: Wei Li (drweili@hotmail.com)

This work was supported in part by the Science and Technology Planning Guidance Project of Xiamen of China under Grant 3502Z20179038, in part by the National Natural Science Foundation of China under Grant 61976183, in part by the Natural Science Foundation of Fujian Province of China under Grant 2019J01850, in part by the Key Teaching and Research Project of Hefei University of China under Grant 2018 hfjyxm09, and in part by the Natural Science Foundation of Anhui Province of China under Grant 2008085MF202.

ABSTRACT Granular computing is a method of studying human intelligent information processing, which has advantage of knowledge discovery. In this paper, we convert a classification problem of sample space into a classification problem of fuzzy clustering granular space and propose a random fuzzy clustering granular hyperplane classifier (RFCGHC) from the perspective of granular computing. Most classifiers are only used to process numerical data, RFCGHC can process not only non-numerical data, such as information granules, but also numerical data. The classic granulation method is generally serial granulation, which has high time complexity. We design a parallel distributed granulation method to enhance efficiency. First, a clustering algorithm with adaptive cluster center number is proposed, where the ratio of standard deviation between categories and standard deviation within categories is as evaluation criterion. The clusters and the optimal amount of cluster centers can be achieved by the method. On the basis of these, sample set can be divided into many subsets and each sample can be granulated by these cluster centers. Then, a fuzzy clustering granular space can be formed, where fuzzy clustering granules, fuzzy clustering granular vectors, and their operators can be defined. In order to get the optimal hyperplane in the fuzzy clustering granular space to classify these samples, we design a loss function and evaluate each category with probability by fuzzy clustering granular hyperplane. In solving the loss function, genetic algorithm based on fuzzy clustering granules is adopted. Experimental results and theoretical analysis show that RFCGHC has good performance.

INDEX TERMS Classification, granular computing, fuzzy sets, parallel distributed granulation, fuzzy clustering granular hyperplane.

I. INTRODUCTION Classification problem is one of the common problems faced by human beings in production and life. There are three important issues of machine learning: How to get reliable knowledge from data and experience, how to build a classifier, and how to assist people in completing the tasks of pattern recognition and data analysis. As machine learning technology gradually becomes an essential force to enhance the advancement of computer and automation field, it is of great value to study construction of classifiers based on machine learning. According to structural characteristics of classifiers, it can be divided into statistical learning classifiers, deep learning classifiers, etc. Here, the statistical learning classifier starts from probability statistics, constructs a mathematical model, and uses data to train and adjust the model to obtain a predictor. Deep learning classifiers specifically refer to classifier models that use deep neural networks as the main model. According to the learning approach, the classifier construction method can be summarized into unsupervised learning, and reinforcement learning, and supervised learning. When using a supervised learning to construct a classifier model, a given set of inputs is required to determine the ground truth. This input and output data are named as a training sample set. The learning system can adjust the model parameters by a certain optimization method according to the difference between the ground truth (label) and the predicted output. Common classifier models such as
statistical learning classification methods represented by support vector machines [1], decision trees [2], and deep learning classification methods represented by convolutional neural networks [3] can be obtained by supervised training. Supervised learning relies upon high quality training samples, and the number of training samples is positively related to model complexity and number of parameters. Currently widely used deep neural network classifiers have complex structures and huge parameters. Training an efficient and accurate deep neural network model often requires tens of thousands or even hundreds of thousands of training samples, and its labeling task is very heavy. When an unsupervised learning method is used to construct a classifier, the training of the model does not rely on external supervision signals. That is, it is the lack of input and output sample pairs for the current environment. The learning system can transfer available knowledge and experience from other models and data, thereby unsupervised classification of current tasks can be done. Typical unsupervised classifier include domain adaptation [4], Zero Short Learning [5], etc. It is rare to use reinforcement learning to build a single classifier. This method is generally used to build an agent, which has multiple functions such as classification, perception, decision generation, and action output. In reinforcement learning, the external environment only gives an evaluation of the output results generated by the learning system, rather than real reference data. The learning system strengthens the rewarded actions and output to adjust parameters and structure [6]. In the same way, the learning system will suppress the actions that are punished and avoid incorrect outputs and decisions. Reinforcement learning is widely used in scenarios that require real-time response and output, such as the game of Go [7], etc. Common reinforcement learning methods include Q learning [8], strategy gradient learning [9], etc. The actual situation of the finite number of training samples greatly restricts construction and application of the classifier [10], [11]. From the 1980s to the beginning of this century, most of the statistical learning classifier were oriented to a small number of sample sets. On the one hand, this stems from the constraints of the development level of sensing technology and computer hardware. But then, it is often greatly difficult to acquire data in specific application fields. In the past ten years, information science and technology have developed rapidly, and massive amounts of complex data from different data sources with high-level semantic relevance have emerged in industry and life. In most cases, the data collected by different sensors for different tasks is complex and heterogeneous. In the face of specific application scenarios, low data quality is an extremely common phenomenon. On the one hand, this low quality lies in the relatively large amount of data and the extremely limited number of labels. On the other hand, it lies in the incompleteness or inaccuracy of the labels. With the substantial increase in sample feature dimensions, sample quality is currently one of the main problems that need to be faced when we constructing classifiers based on deep learning technology, and it is also a hot research issue that scholars are paying attention to.

There is advantage of knowledge discovery in Granular Computing. It is an information processing theory and method for studying human intelligence [12]. Its basic idea and concept is to describe and process uncertain information and obtain knowledge [13]. Granular computing uses the three-element model of granular, granular layer and granular structure to abstract, decompose, synthesize, transform and analyze real problems. Hence, it can form an information analysis, processing and problem solving method, which is similar to human thinking, cognition and reasoning. The ternary thought of granular computing points out a formal and systematic granular computing research framework from the perspectives of philosophy, methods and mechanisms, namely, structural thinking, structural problem solving, and structural information processing [14]. Granular computing can use three methods, such as inside-out, bottom-up, and top-down in a multi-level granular structure description space to decompose complex problems into several small problems, reduce complexity to simplicity, divide and conquer. We can find an approximate solution to the problem between granular structures, thereby giving a granular calculation method to deal with uncertain knowledge. Just as Bo Zhang and Ling Zhang, the founders of the quotient space of granular computing theory, pointed out, “A recognized feature of human intelligence is the ability to observe and analyze the same problem from different granules. People can not only solve problems in the world of different granules, but also quickly jump from one granular world to another granular world, back and forth freely, without difficulty [15].” It can be seen that how to describe, define, and construct the three main characteristics of causation, organization, and granulation of human cognition are difficult problems in the current field of granular computing research.

The thought of granular computing originated from the concept of fuzzy information granulation put forward by Zadeh, the father of American fuzzy sets [16]. He believed that human cognition has three main characteristics, namely granulation, organization, and causation [16]. Hobbs proposed concept “Granularity” in 1985 [18]. Pawlak introduced the problem of knowledge granularity in approximate space from rough set angle [19]. After that, Lin proposed concept “Granular Computing” based on binary relations [20]. Yager and his colleagues also discussed value of granular computing in intelligent engineering [21]. Yao studied construction and calculation of granules in granular computing in 2000 [23]. On the basis of discussing the basic principles and problems of granular computing, Yao presented a granular computing model on the basis of set theory in 2004 [24]. Four years later, he studied the past, present, and future development context of granular computing and proposed the ternary theory of granular computing [12], [25]. In 2009, he deeply explored the connection between granular computing and cognitive science, and built a framework of granular computing for cognitive concept learning [26]. In order to integrate concepts and methods in several different fields, Feiyue Wang and his colleagues proposed a
computational theoretical framework for a language dynamic system based on word calculation [27]. Guoyin Wang and his colleagues proposed a granular computing model based on tolerant relations [28]. The granular computing model based on single granularity only seeks an approximate solution to the problem from a single angle. In recent years, researchers have shifted the focus of research to multiple granularity. Compared with single granularity, multi granularity integrates the relationship between different single granularity, so it has stronger expressive power, and problem solving based on multi granularity often has advantages in complexity and performance. Yuhua Qian and his colleagues presented a multi-granulation rough set model. This is an extension of classic single granularity rough set theory. They gave definition of set approximation by multiple equivalence relations on the corpus in the model [29]. When actually solving complex problems, the multi-granularity rough set model can think of the comprehensive evaluation of multiple-granulation. Weizhi Wu and his teamwork proposed a multi-scale rough set model [30]. Feng Zhu et al. studied three different covering rough set models in the literature [31], and gave the approximate conversion relationship between the models [32]. Feng and his teamwork proposed a hierarchical rough set model in the literature [33]. Wei Li et al. studied a set of classification models from the angle of granular computing and achieved good results [34]–[36]. Granular computing has been widely used, such as using granular computing and reply state networks to build error diagnosis models [37], building neural networks from a granular perspective for classification [38], [39] etc.

II. CONTRIBUTIONS

This paper constructs a random fuzzy clustering granular hyperplane classifier from the perspective of granular computing. Main contributions are as follows:

- Aiming at the problem of high time complexity in the granulation process, we propose a parallel distributed fuzzy clustering granulation approach. According to this principle, we designed a clustering algorithm with an adaptive amount of cluster centers. The algorithm is based on the ratio of standard deviation between clusters and standard deviation within clusters to optimize the number of clusters and cluster centers. On the basis of these optimal clusters, fuzzy granulation can be executed parallely.

- We define the concepts on fuzzy clustering granule, fuzzy clustering granular vector, and fuzzy clustering granular hyperplane and design the operators between them. In fuzzy clustering granule space, we can find an optimal fuzzy clustering granular hyperplane to classify samples, of which parameters can be learned by optimizing the loss function by genetic algorithm. This approach can be used for task of binary classification or multiple classification.

III. THE PROBLEM

Given a classification system $S = (X, A, V, f, Y)$, where $X = \{x_1, x_2, \ldots, x_N\}$ is an sample set, $A = \{a_1, a_2, \ldots, a_M\}$ is non-empty attribute set, $V = \bigcup_{a \in A} V_a$ is a range, $f : X \times A \rightarrow V$ is information function $(f(x, a) \in V_a, a \in A, x \in X)$, $Y = \{y_1, y_2, \ldots, y_L\}$ is a category set or label set, and $y_i$ is a label corresponding to $x_i (x_i \in X, y_i \in Y)$, our aim is to construct a classification model, get parameters of model by training sample set, and predict category of test sample.

IV. THE MAIN ALGORITHM

To solve the classification problem introduced above, the model is designed as follows: First, to reduce the complexity of fuzzy granulation as much as possible and granulate parallelly, the data is clustered to get cluster centers and the cluster number. Second, data is fuzzy granulated by the cluster centers. In this process, fuzzy cluster granule, fuzzy cluster granular vector are defined and operators are also designed in the fuzzy cluster granular space. Next, fuzzy clustering hyperplane is defined and loss function is also designed. To minimize the loss function, genetic algorithm is adopted in fuzzy clustering granular space. In other words, optimal fuzzy granular vectors can be learned by genetic algorithm. Finally, given a test sample, its category can be predicted by calculating the maximum probability of categories. The overview is shown in Figure 1.

A. PRINCIPLE OF ADAPTIVE RANDOM CLUSTERING

The classic $K$-means algorithm requires the amount of cluster centers in advance. This parameter is critical to the clustering effect. To avoid the abnormality of the final result caused by the sensitivity, we propose an adaptive random clustering algorithm. We know that if the standard deviation between clusters is large and the standard deviation within clusters is small, then the clustering effect is good. Here, $\sigma_c$ denotes the standard deviation between clusters and $\sigma_i$ is the standard deviation within the $k$-th clusters. We employ $\sigma_c^2 / \sum_{k=1}^{K} \sigma_i^2$ as evaluation of clustering performance. The larger the value is, the better the clustering effect is. Our aim is to randomly adjust cluster centers and their number to make the ratio continue to increase until the maximum times of iterations is reached. In each iteration, we can get a cluster center set and merge cluster center and evaluation value into a set. When the maximum amount of iterations is reached, a series of cluster centers and evaluation values can be obtained. Then the maximum evaluation value and its corresponding to cluster center set can be selected, namely the parameters can be learned. The detailed stage is as follows.

Step 1. Delete samples with missing attribute values.

Step 2. Normalize attribute value of the sample.

Step 3. Initialize the maximum iterations as $\text{Max}$, set the evaluation set to an empty set $\text{EvSet} \leftarrow \phi$ composed of cluster center and the number of cluster center, and set the current iteration number $t = 1$. 
In this section, we introduce how to achieve data fuzzy granulation with the help of cluster center sets. The fuzzy granulation can be executed parallelly distributed. Time complexity will be greatly reduced compared with the classical approach.

**B. FUZZY GRANULATION OF DATA**

In this section, we introduce how to achieve data fuzzy granulation with the help of cluster center sets. The fuzzy granulation can be executed parallelly distributed. Time complexity will be greatly reduced compared with the classical approach.

**Step 4.** Random generate the number of cluster center \( K \in (\sqrt{N},N) \), and random select \( K \) samples as cluster centers \( c_1, c_2, \ldots, c_K \).

**Step 5.** Calculate the distance from each sample point to \( K \) cluster centers, find the cluster center closest to each sample point, and assign the sample to the closest cluster. Then recalculate the center of each cluster and set it as the new cluster center.

**Step 6.** Compare the sum of the standard deviations of the cluster with the threshold. Namely, if \( \sum_{k=1}^{K} \sigma_k \geq \text{Threshold} \), go **Step 5**, otherwise go **Step 7**.

**Step 7.** Assign current cluster center set \( E_t \leftarrow \{c_1, c_2, \ldots, c_K\} \).

**Step 8.** Update evaluation set \( \text{EVSet} \leftarrow \text{EVSet} \cup E_t \).

**Step 9.** Update current iteration number \( t \leftarrow t + 1 \).

**Step 10.** If current iteration number \( t > \text{Max} \), go **Step 11**, otherwise go **Step 4**.

**Step 11.** Select the optimal cluster center set corresponding to the largest ratio of the standard deviation between the clusters and the sum of the standard deviation within the cluster in \( \text{EVSet} \) and its number. That is, \( \hat{E} = \arg \max \frac{\sigma_d}{\sum_{j=1}^{K} \sigma_j} \)

\[ C^* = \{c_1, c_2, \ldots, c_{|E|}\}, \quad K^* = |C^*| . \]

We have given the principle of adaptive random clustering mentioned above.

**PREDICTION CATEGORY**

Given a cluster center set \( C = \{c_1, c_2, \ldots, c_K\} \), an sample set \( X \), and an attribute set \( A \), the similarity between \( \forall x_i \in X \) and \( \forall c_j \in C \) can be defined:

\[ s_0(x_i, c_j) = 1 - \frac{|f(a, x_i) - f(a, c_j)|}{\max(0, \min(1, f(a, x_i) - f(a, c_j)))} \]

(1)

where \( f(a, x_i) \in [0,1] \) and \( f(a, c_j) \in [0,1] \). Obviously, \( s_0(x_i, c_j) \in [0,1] \). Thus, a fuzzy clustering granule induced by \( x_i \) can be defined:

\[ G_a(x_i) = \frac{s_0(x_i, c_1)}{c_1} + \frac{s_0(x_i, c_2)}{c_2} + \cdots + \frac{s_0(x_i, c_K)}{c_K} . \]

(2)

For simplicity, it can also be written as

\[ G_a(x_i) = \sum_{k=1}^{K} \frac{s_0(x_i, c_k)}{c_k} . \]

(3)

“−” denotes delimiter and “+” represents the union of elements. In other words, \( G_a(x_i) \) is neighborhood set of \( x_i \) in the cluster center set. The cardinal of Fuzzy clustering granule \( G_a(x_i) \) can be calculated by

\[ |G_a(x_i)| = \sum_{j=1}^{K} s_0(x_i, c_j) . \]

(4)

For \( \forall x, x' \in X \), a set of operators of fuzzy clustering granules induced by \( x' \) and \( x \) can be defined:

\[ G_a(x) \lor G_a(x') = \sum_{j=1}^{K} \min \left( 1, \frac{s_0(x, c_j)^p + s_0(x', c_j)^p}{c_j} \right) \]

(5)

\[ \lor \]

\[ c_j \]

**PARAMETER LEARNING**
\[ G_d(x) \wedge G_d(x') = \sum_{j=1}^{K} \left( 1 - \min\{1, [(1 - s_d(x, c_j))^{\beta} + (1 - s_d(x', c_j))^{\beta} \}^{\frac{1}{\beta}} \right) \]

(6)

\[ G_d(x) - G_d(x') = \sum_{j=1}^{K} \left( s_d(x, c_j) - s_d(x', c_j) \right) \]

(7)

\[ G_d(x) \oplus G_d(x') = G_d(x) \lor G_d(x') - G_d(x) \wedge G_d(x'), \]

(8)

where \( \beta \) is a parameter of metric and it satisfies \( \beta > 0 \). For \( \forall B \subseteq A \) and \( B = \{a_1, a_2, \ldots, a_{|B|}\}, |B| \leq |A| \), fuzzy clustering granular vector induced by \( x \) on \( B \) can be defined:

\[ \tilde{G}_B(x) = \frac{G_{d_1}(x)}{a_1} + \frac{G_{d_2}(x)}{a_2} + \cdots + \frac{G_{d_{|B|}}(x)}{a_{|B|}} = \sum_{i=1}^{B} \frac{G_{d_i}(x)}{a_i} \]

(9)

The above symbol "\( \wedge \)" means union and the symbol "\( \ominus \)" means separator. Its modulus can be expressed by

\[ |\tilde{G}_B(x)| = \sum_{i=1}^{B} |G_{d_i}(x)|. \]

(10)

The operators of fuzzy clustering granular vectors are given below. Let \( \tilde{G}_B(x) \) and \( \tilde{G}_B(x') \) be fuzzy clustering granular vectors induced by \( x \) and \( x' \) on attribute subset \( B \). The operators between them are as follows:

\[ \tilde{G}_B(x) \lor \tilde{G}_B(x') = \sum_{i=1}^{B} \frac{G_{d_i}(x) \lor G_{d_i}(x')}{a_i} \]

(11)

\[ \tilde{G}_B(x) \wedge \tilde{G}_B(x') = \sum_{i=1}^{B} \frac{G_{d_i}(x) \wedge G_{d_i}(x')}{a_i} \]

(12)

\[ \tilde{G}_B(x) - \tilde{G}_B(x') = \sum_{i=1}^{B} \frac{G_{d_i}(x) - G_{d_i}(x')}{a_i} \]

(13)

\[ \tilde{G}_B(x) \oplus \tilde{G}_B(x') = \sum_{i=1}^{B} \frac{G_{d_i}(x) \oplus G_{d_i}(x')}{a_i} \]

(14)

The similarity of fuzzy clustering granular vectors can be defined:

\[ s(\tilde{G}_B(x), \tilde{G}_B(x')) = \frac{1}{|B| \cdot |C|} \sum_{a \in B} \frac{|G_{d_a}(x) \oplus G_{d_a}(x')|}{|G_{d_a}(x) \lor G_{d_a}(x')|}. \]

(15)

From the above information granulation process, it can be drawn that the obtained information granules are all on the basis of the cluster center set and are fuzzy. The granular space composed of these fuzzy cluster granules is called the fuzzy cluster granular space.

**Theorem 1:** For \( \forall x_i, x_j \in X \), the similarity of fuzzy clustering granule satisfies

\[ 0 \leq s(\tilde{G}_B(x_i), \tilde{G}_B(x_j)) \leq 1 \]

(16)

**Proof:** According to equation (3), we have \( G_{d_i}(x_i) = \sum_{k=1}^{K} s_{d_i}(x_{i_k}, c_k) \), \( G_{d_j}(x_j) = \sum_{k=1}^{K} s_{d_i}(x_{j_k}, c_k) \). As defined in equation (1), we can obtain \( 0 \leq s_{d_i}(c_k, x_i) \leq 1 \) and \( 0 \leq s_{d_i}(c_k, x_j) \leq 1 \). According to equation (4),

\[ 0 \leq |G_{d_i}(x_i)| = \sum_{c_j \in C} s_{d_i}(c_j, x_i) \leq |C| \]

is established. Then it is defined by fuzzy clustering granular vector (see equation (9)), we can get

\[ \tilde{G}_B(x_i) = \sum_{a \in B} \frac{G_{d_a}(x_i)}{a_i}, \text{ of which module is } \tilde{G}_B(x_i) = |B| \cdot |C|. \]

Furthermore, we have \( 0 \leq |\tilde{G}_B(x_i)| \leq |B| \cdot |C|. \) Since \( G_{d_i}(x_j) \oplus G_{d_j}(x_j) = G_{d_i}(x_j) \lor G_{d_j}(x_j) - G_{d_i}(x_j) \wedge G_{d_j}(x_j) \) is established, \( 0 \leq \sum_{a \in B} \frac{|G_{d_a}(x_i) \oplus G_{d_a}(x_j)|}{|G_{d_a}(x_i) \lor G_{d_a}(x_j)|} \leq |B| \cdot |C| \) is also established. We divide both sides by \( |B| \cdot |C| \) and obtain \( 0 \leq \frac{1}{|B| \cdot |C|} \sum_{a \in B} \frac{|G_{d_a}(x_i) \oplus G_{d_a}(x_j)|}{|G_{d_a}(x_i) \lor G_{d_a}(x_j)|} \leq 1 \), namely \( 0 \leq s(\tilde{G}_B(x_i), \tilde{G}_B(x_j)) \leq 1 \) is established. This completes proof.

**Theorem 2:** (Monotonicity) For \( \forall x \in X \), attribute subsets \( B, E \) satisfy to \( B \subseteq E \subseteq A \). Let \( \tilde{G}_B(x) \) and \( \tilde{G}_E(x) \) be fuzzy clustering granular induced by \( x \) on \( B \) and \( E \) respectively. Then \( |\tilde{G}_B(x)| \leq |\tilde{G}_E(x)| \) is established.

**Proof:** According to \( \tilde{G}_B(x) = \sum_{i=1}^{B} \frac{G_{d_i}(x)}{a_i} \), \( \tilde{G}_E(x) = \sum_{i=1}^{E} \frac{G_{d_i}(x)}{a_i} \) (see equation (9)) and \( B \subseteq E \subseteq A \), for \( a \in B \), then we have \( a \in E \) and \( |B| \leq |E| \). Namely, if \( G_{d_i}(x) \in \tilde{G}_B(x) \), we can get \( G_{d_i}(x) \in \tilde{G}_E(x) \). To sum up, equation \( |\tilde{G}_B(x)| \leq |\tilde{G}_E(x)| \) is established. This completes proof.

Below we give an example to calculate the metric of fuzzy clustering granular vector.

**Example 1.** As demonstrated in Table 1, given an sample set \( X = \{x_1, x_2, x_3, x_4\} \), an attribute set \( A = \{a_1, a_2, a_3\} \), a label set \( Y = \{1, 0\} \), a cluster center set \( C = \{c_1, c_2\} \), and metric parameter \( \beta = 2 \), the fuzzy clustering granulation is as follows:

| \( X \) | \( C \) | \( \beta \) | \( a_1 \) | \( a_2 \) | \( a_3 \) | \( l \) | \( p \) |
|--------|---|---|---|---|---|---|---|
| \( x_1 \) | \( 0.3 \) | \( 0.2 \) | \( 0.2 \) | 1 | 2 |
| \( x_2 \) | \( 0.4 \) | \( 0.3 \) | \( 0.1 \) | 1 | 2 |
| \( x_3 \) | \( 0.5 \) | \( 0.4 \) | \( 0.3 \) | 0 | 2 |
| \( x_4 \) | \( 0.2 \) | \( 0.4 \) | \( 0.2 \) | 0 | 2 |
| \( c_1 \) | \( 0.2 \) | \( 0.3 \) | \( 0.1 \) | N/A | N/A |
| \( c_2 \) | \( 0.3 \) | \( 0.2 \) | \( 0.3 \) | N/A | N/A |

We take samples \( x_1 \) and \( x_2 \) as an example. On the attribute \( a_1 \), we have

\[ s_{a_1}(x_1, c_1) = 1 - [0.3 - 0.2] = 0.9, \]
\[ s_{a_1}(x_2, c_1) = 1 - [0.3 - 0.3] = 1, \]
\[ s_{a_2}(x_1, c_1) = 1 - [0.2 - 0.3] = 0.9, \]
\[ s_{a_2}(x_2, c_1) = 1 - [0.2 - 0.2] = 1, \]
\[ s_{a_3}(x_1, c_1) = 1 - [0.2 - 0.1] = 0.9, \]
\[ s_{a_3}(x_2, c_1) = 1 - [0.2 - 0.3] = 0.9. \]

As defined in fuzzy clustering granule, we can obtain

\[ G_{d_1}(x_1) = \frac{0.9}{c_1} + \frac{1}{c_2}, G_{d_2}(x_1) = \frac{0.9}{c_1} + \frac{1}{c_2}, G_{d_3}(x_1) = \frac{0.9}{c_1} + \frac{0.9}{c_2}. \]
The fuzzy clustering granular vector induced by $x_1$ on $A$ can be written as $\hat{G}_A(x_1) = \sum_{i=1}^{3} \frac{G_a(x_1)/a_i}{a_i}$. Similarly, the fuzzy cluster granular vector induced by $x_2$ on $A$ can be written as $\hat{G}_A(x_2) = \sum_{i=1}^{3} \frac{G_a(x_2)/a_i}{a_i}$, where $G_a(x_1) = \frac{0.8}{c_1} + \frac{0.9}{c_2}$. According to

$$G_a(\bar{x}_1) \bigvee G_a(\bar{x}_2) = \sum_{j=1}^{K} \min\{ \{s_a(x, c)\}^{p} + s_a(x', c)\}^{1/p}$$

and $p = 2$, we can get

$$G_a(\bar{x}_1) \bigvee G_a(\bar{x}_2) = \min\{ \{s_a(x_1, c_1)^2 + s_a(x_2, c_1)^2\}^{1/2} \}$$

$$+ \min\{ \{s_a(x_1, c_2)^2 + s_a(x_2, c_2)^2\}^{1/2} \}$$

$$= \frac{1}{c_1} + \frac{1}{c_2}$$

Similarly, we can obtain

$$G_a(\bar{x}_1) \bigwedge G_a(\bar{x}_2) = 1 - \min\{ \{1 - s_a(x_1, c_1)^2 + (1 - s_a(x_2, c_1)^2)\}^{1/2} \}$$

$$+ \min\{ \{1 - s_a(x_1, c_2)^2 + (1 - s_a(x_2, c_2)^2)\}^{1/2} \}$$

$$= \frac{0.7764}{c_1} + \frac{0.9}{c_2}$$

Furthermore, we have

$$G_a(\bar{x}_1) \bigoplus G_a(\bar{x}_2) = G_a(\bar{x}_1) \bigvee G_a(\bar{x}_2) - G_a(\bar{x}_1) \bigwedge G_a(\bar{x}_2)$$

$$= \frac{1}{c_1} + \frac{1 - 0.9}{c_2} = 0.2236 + \frac{0.1}{c_2}$$

$$|G_a(\bar{x}_1) \bigoplus G_a(\bar{x}_2)| = \frac{1}{c_1} + \frac{1 - 0.9}{c_2} = 0.2236$$

Therefore, under metric parameter $p = 2$, the similarity between fuzzy clustering granular vector induced by $x_1$ and one induced by $x_2$ can be calculated by

$$s(\hat{G}_A(x_1), \hat{G}_A(x_2)) = \frac{1}{|A| \times |C|} \sum_{a \in A} |G_a(\bar{x}_1) \bigoplus G_a(\bar{x}_2)|$$

$$= 0.0235.$$
By maximizing likelihood function $L(W, \bar{Q})$, the optimal value of $W$ and $\bar{Q}$ can be obtained.

### D. MULTI-CLASSIFICATION PROBLEM

The model mentioned above is a binary-classification model, we think of the multi-classification model below. Let discrete random variable $y$ satisfy $y \in \{1, 2, \ldots, D\}$. For calculation convenience, an indicator function can be introduced as follows:

$$I_d(y) = \begin{cases} 1, & \text{if } y = d \\ 0, & \text{if } y \neq d \end{cases} \quad (21)$$

If a classification belongs to $d$, it can be denoted by $\bar{y}_d = [0, 0, \ldots, I_d(y), \ldots, 0]^T$ where $T$ is transposition. If $\bar{G}_A(x)$ belongs to $d$, its probability can be defined:

$$P(y = I_d(y)) = \frac{\exp(\bar{W}_d \cdot \bar{G}_A(x) \vee \bar{Q}_d)}{\sum_{d=1}^{D} \exp(\bar{W}_d \cdot \bar{G}_A(x) \vee \bar{Q}_d)} \quad (22)$$

where $W_d, Q_d$ are the weights corresponding to classification $d$ and $W_d, Q_d \in \mathbb{R}^M$.

Similar to the binary classification problem, the likelihood function can be defined:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \sum_{d=1}^{D} I_d(y_i) \log(P(y = I_d(y_i)) \bar{G}_A(x_i))) \quad (23)$$

where $\theta = \{(\bar{W}_i, \bar{Q}_i)|i = 1, 2, \ldots, D\}$.

1) **SOLVING PARAMETERS**

Genetic algorithm is used to solve the likelihood function. As well known, the algorithm can get global optimal solution. That is, finding the optimal solution $\bar{W}^*$ and $\bar{Q}^*$ to maximize $L(W, \bar{Q})$ is our goal. We will concatenate $\bar{W}$ and $\bar{Q}$ into one vector and use equation (24) as fitness function to select evolution solution in each iteration. When iterative time reaches maximum value or the standard deviation of a set of fitness function values does not change obviously, the algorithm will end. The optimal parameters are selected from a series of parameters which can maximize the fitness function value.

$$F(\bar{W}_j, \bar{Q}_j) = \frac{L(W_j, \bar{Q}_j)}{\sum_{i=1}^{Z} L(W_i, \bar{Q}_i)} \quad (24)$$

where $Z$ is the number of solution.

The stage is as follows:

**Step 1.** Delete the samples missing attribute values.

**Step 2.** Calculate the cluster center set (see Table 1).

**Step 3.** Parallel distributed the cluster center set.

**Step 4.** Set the number of solution $Z$, the current iteration $I \leftarrow 0$, precision $\epsilon$, maximum iteration $\text{Max}$, and consecutive iteration $t$ (after several consecutive iterations, the maximum fitness function value has no longer obvious change).

**Step 5.** Randomly generate $Z$ pairs of solutions, namely $NP \leftarrow \{(\bar{W}_1, \bar{Q}_1), (\bar{W}_2, \bar{Q}_2), \ldots, (\bar{W}_Z, \bar{Q}_Z)\}$.

**Step 6.** Calculate the fitness of each pair of solutions in $NP$ and they can be written as

$$F(\bar{W}_1, \bar{Q}_1), F(\bar{W}_2, \bar{Q}_2), \ldots, F(\bar{W}_Z, \bar{Q}_Z)$$

**Step 7.** Transform the solutions into binary code.

**Step 8.** Given a series of probabilities $p_i, i = 1, 2, \ldots, Z$, if $p_i < F(\bar{W}_j, \bar{Q}_j)$, then the solution $(\bar{W}_j, \bar{Q}_j)$ will be selected and duplicated. Then, the process is ended until $Z$ pairs of solutions are selected and duplicated from $NP$, and these solutions can form a new group $NP_1$.

**Step 9.** According to the number $c$ of solutions participating in the crossover determined by the crossover probability $P_c$, $c$ pairs of solutions are randomly selected from $NP_1$ and are paired for crossover operation. The original solutions are replaced with the new solutions generated by crossover operation to obtain population $NP_2$.

**Step 10.** According to the number $m$ of mutations determined by the mutation probability $P_m$, $m$ pairs of solutions are selected randomly from $NP_2$ and performed mutation operation respectively. The original solutions are replaced with the new solutions generated by mutation operation to obtain population $NP_3$.

**Step 11.** Update the solution set $NP$ with $NP_3$, namely $NP \leftarrow NP_3$.

**Step 12.** Select the maximum solution of $F(\bar{W}, \bar{Q})$ from $NP$

$$F(j) \leftarrow F(\bar{W}^*, \bar{Q}^*)$$

**Step 13.** Calculate the standard deviation of sequence $F_1, F_{t-1}, \ldots, F_{t-1}$ by

$$\sigma \leftarrow \text{Var}(F_1, F_{t-1}, \ldots, F_{t-1})$$

**Step 14.** Update the current iteration $I \leftarrow I + 1$.

**Step 15.** If $\sigma < \epsilon$ or $I > \text{Max}$, go Step 16; Otherwise, go Step 6.

**Step 16.** Return $(\bar{W}^*, \bar{Q}^*) \leftarrow \arg \max_{(\bar{W}, \bar{Q}) \in NP}(F(\bar{W}, \bar{Q}))$

2) **PREDICATION**

After performing parameter learning, the random fuzzy clustering granular vector classification model is designed. When a test sample is given, it is first fuzzy granulated, and then the trained classification model is used to predict. The specific algorithm is shown in Table 2.

### V. EXPERIMENTAL ANALYSIS

In the experiments, we employ six data sets for the experimental test, where three of them are from UCI and others are constructed with 1% noise based on the three data set, as demonstrated in Table 3. 10-fold cross-validation is adopted to experiment. That is, we randomly sample data of 80% as training set, and data of 20% as the test set, and then perform one verification. The operation is repeated 10 times.
The average accuracy, and the average execution time are calculated as performance evaluation parameters. In terms of efficiency, we compared the parallel distributed clustering fuzzy granulation method proposed in this paper, serial clustering fuzzy granulation method, and traditional serial granulation method, as shown in Figure 2. In terms of average accuracy, we compared Long Short-Term Memory (LSTM), Support Vector Machines (SVMs), Back Propagation Neural Network (BPNN), and RFCGHC on these data sets.

In granular computing, granulation is an important and time-consuming task. At present, most fuzzy granulation methods are serial calculations, and the efficiency is so low that it can not meet the demand. If the tasks can be executed parallelly, the efficiency will be improved to a certain extent. For a computing task, to convert from serial computing to parallel computing, the prerequisite that needs to be met is that the internal logic of the computing task is parallelizable. We adopt the method of clustering first and then granulation, which can make fuzzy granulation parallelizable.

The single-node multi-process concurrent mode can enhance the operating efficiency of computing tasks to some extent, but due to the limitation of single-node computing resources, the improvement of computing efficiency is limited. The traditional multi-node data parallel computing model can solve the problem of limited computing resources, but due to the lack of a distributed file system, large IO problems will occur when sharing data sources, which affects read and write efficiency. MapReduce is usually used for large-scale data parallel processing tasks. By abstracting

---

**TABLE 2. Predicted algorithm.**

| Step | Description |
|------|-------------|
| 1    | Fuzzy granulation to get fuzzy clustering granular vector $G_A(x)$. |
| 2    | Calculate the probability of each category $P_1, P_2, \ldots, P_D$ according to equation (22). |
| 3    | Return $y_x = \arg\max_{i \in D} \{P_i\}$. |

**TABLE 3. Data sets from UCI.**

| Index | Data sets                        | Number of Sample | Number of Attributes | Number of Classification |
|-------|----------------------------------|------------------|----------------------|-------------------------|
| 1     | Mushroom                         | 8124             | 22                   | 2                       |
| 2     | Mushroom with Noise              | 8124             | 22                   | 2                       |
| 3     | EEG Eye State                    | 14980            | 14                   | 2                       |
| 4     | EEG Eye State with Noise         | 14980            | 14                   | 2                       |
| 5     | Activity recognition             | 75128            | 8                    | 4                       |
| 6     | Activity recognition with Noise  | 75128            | 8                    | 4                       |

**FIGURE 2.** Comparison of executing time.
a hierarchical computing model, large data sets are fragmented into sub-computing tasks and assigned to computing nodes. The MapReduce model draws on the built-in functions Map and Reduce in the functional programming language. Conceptually, Map is called mapping, and Reduce is called reduction. The main idea is: split the sequence file job into multiple independently runnable Map tasks, distribute them to multiple processors to run, generate intermediate calculation results, and then combine the reduce task operations to produce the final output results. Both Map tasks and Reduce tasks can be highly parallelized. MapReduce is a programming model suitable for distributed parallel computing, and the Hadoop Distributed File System (HDFS) can solve IO problems well.

The MapReduce calculation process is mainly summarized into two parts: Map and Reduce. Map receives input in the form of \(<key, value>\) (see Table 4), and then outputs intermediate results in the form of \(<key, value>\). Here, \(key = x_i, value = f(a_1, x_i), ..., f(a_M, x_i), C >\) (as shown in the Table 5). The output of Reduce is detailed in Table 6. Without clustering, the granularity time complexity is \(O(N^2 \times M)\). In contrast, with clustering, its complexity is \(O(N \times K \times M)\) \((K < N)\). After using MapReduce parallelization, the time complexity is \(\widetilde{O}\left(\frac{N \times K \times M}{B}\right)\), where \(B\) is the number of subsets of the data set, and each subset corresponds to a map. As shown in Figure 2, the abscissa denotes the amount of cluster centers, and the ordinate is the average accuracy. When the number of samples is \(N = 10000\) and the number of clustering centers is \(K = 6074\), the average time consumed by the serial granulation is 100 minutes, the average execution time of the serial clustering granulation is 54 minutes, and the average execution time of parallel distributed granulation is 14 minutes. Parallel granulation with clustering reduced about 46% and 86%, respectively. When \(N = 50,000\) and \(K = 31966\), the serial granulation consumes 2500 minutes, and the serial clustering granulation execution time is 1598 minutes, which is a reduction of 36%. The parallel cluster granulation execution time is 400 minutes, which is a reduction of 84% compared with the serial granulation. When \(N = 60,000\) and \(K = 32,683\), the parallel cluster granulation execution time is reduced by 46% and 86% compared with the serial granulation and the serial clustering granulation. It can be concluded that as the amount of samples increases, the execution efficiency of serial clustering granulation and parallel clustering granulation methods has improved greatly compared with serial granulation.

In the RFCGHC model, the amount of cluster centers is an important parameter that affects the accuracy of classification. We analyzed the relationship between the amount of cluster centers and the accuracy, as illustrated in Figure 3-5. As demonstrated in Figure 3, in the noise-free data set Mushroom, the accuracy of SVMs, BP, and LSTM reached 0.952, 0.947, and 0.931, respectively, while RFCGHC reached the highest value of 0.989 when the amount of cluster centers reached 5600. The average accuracy increased by 3.89%, 4.44%, and 6.23%. After adding noise to the data set, the amount of center clusters is 5600, RFCGHC increases 7.97%, 8.21%, and 10.17% respectively compared with SVMs, BP and LSTM. In the two data sets, SVMs, BP, LSTM, and RFCGHC decreased by 5.15%, 4.86%, 4.94%, and 1.02%, respectively. It can be seen that the RFCGHC has the smallest drop and is less affected by noise than other three approaches.

In the noise-free data set EEG, as shown in Figure 4-a, when \(K = 10500\), the average accuracy of RFCGHC reaches the highest value of 0.965, while SVMs, BP and LSTM are 0.934, 0.928, and 0.933, respectively, while RFCGHC reached the highest value of 0.989 when the amount of cluster centers reached 5600. The average accuracy increased by 3.89%, 4.44%, and 6.23%. After adding noise to the data set, when the amount of cluster centers is 5600, RFCGHC increases 7.97%, 8.21%, and 10.17% respectively compared with SVMs, BP and LSTM. In the two data sets, SVMs, BP, LSTM, and RFCGHC decreased by 5.15%, 4.86%, 4.94%, and 1.02%, respectively. It can be seen that the RFCGHC has the smallest drop and is less affected by noise than other three approaches.

In the noise-free data set EEG, as shown in Figure 4-a, when \(K = 10500\), the average accuracy of RFCGHC reaches the highest value of 0.965, while SVMs, BP and LSTM are 0.934, 0.928, and 0.933. RFCGHC improved by 3.32%, 3.99% and 3.43% respectively. In the noise-containing EEG data set, as shown in Figure 4-b, RFCGHC reached a peak accuracy of 0.955 at \(K = 10500\). Compared with SVMs, BP and LSTM, it increased by 7.42%, 8.03% and 7.67%.
respectively. It can be seen from this that RFCGHC is not sensitive to noise, and the decrease is only 1.04%, while the decrease of SVMs, BP and LSTM reaches 4.82%, 4.74%, and 4.93%.

The number of data set activity recognition samples has increased significantly compared with the previous two data sets, reaching 75128, and the number of categories has also increased to 4. As shown in Figure 5-a, without noise, when $K = 60,000$, the average accuracy of RFCGHC reaches the highest value of 0.958, SVMs is 0.921, BP is 0.905, and LSTM is 0.951 (improvement by 4.02%, 5.86%, and 0.74%, respectively). Compared with the first two data sets, LSTM performs better than SVMs and BP, and RFCGHC is slightly better than LSTM. As shown in Figure 5-b, in the activity recognition data set with noise, RFCGHC increased by 7.62%, 8.36%, and 0.96% compared with SVMs, BP and LSTM, respectively. In this data set, RFCGHC and LSTM are less sensitive to SVMs and BP.

From the above analysis, we can see that with appropriate parameters, RFCGHC performs better than LSTM, BP, and SVMs in multiple data sets. We also draw conclusion that the accuracy fluctuates with the K value, namely not
monotonously increasing or decreasing. The main reason lies in 10-fold cross-validation. That is, the algorithm is evaluated on a different training set and test set each time, which has a certain degree of randomness, but it also measures the performance of the algorithm more objectively. In addition, global optimal solution can be found by genetic algorithm in solving parameters. The accuracy of algorithm is different for different data sets. In general, data sets with more samples, fewer attributes, and fewer categories have higher accuracy of the algorithm; On the contrary, a dataset with a small number of samples, many attributes, and many categories has lower accuracy of the algorithm. The imbalance of the data set may also decrease the performance. For noisy data sets, we found that RFCGHC are more robust than any other three algorithms. The reason is that in the process of fuzzy granulation, RFCGHC contains a global comparison idea, which overcomes the interference of noise to a certain extent. If $K$ is too small, the classification accuracy may be reduced; If $K$ is too large, noise may be included and it may also be on the decline. Thereby we can adjust $K$ to enhance the performance.

VI. CONCLUSION

We present a random fuzzy clustering granular hyperplane classifier in the paper. This classifier can be used for binary classification or multi-classification problems. In this model, we introduce a parallel distributed fuzzy granulation to obtain fuzzy granules and fuzzy granular vectors, and convert the classification problem of samples into the classification problem of fuzzy clustering granular space to solve. In fuzzy clustering granular space, we gave definitions of fuzzy clustering granule, fuzzy clustering granular vector, and fuzzy clustering hyperplane. We also designed their operators and loss function in fuzzy clustering granular space, of which solution can be obtained through genetic algorithm. The binary classification or multi-classification can be solved by the classifier. In the future, we will combine this algorithm with 5G edge computing to apply in internet of things.

REFERENCES

[1] C. Cortes and V. Vapnik, “Support-vector networks,” Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995.
[2] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees. Monterey, CA, USA: Wadsworth & Brooks/Cole Advanced Books & Software, 1984.
[3] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[4] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2962–2971.
[5] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean, “Zero-shot learning by convex combination of semantic embeddings,” in Proc. 2nd Int. Conf. Learn. Represent. (ICLR), 2014, pp. 1–9.
[6] R. S. Sutton and A. G. Barto, “Reinforcement learning: An introduction,” IEEE Trans. Neural Netw., vol. 9, no. 5, p. 1054, Sep. 1998.
[7] D. Silver, R. Sutton, and M. Müller, “Reinforcement learning of local shape in the game of go,” in Proc. 20th Int. Joint Conf. Artif. Intell. (IJCAI), Banff, AB, Canada, 2007, pp. 1053–1058.
[8] L. M. D. Da Silva, M. F. Torquato, and M. A. C. Fernandes, “Parallel implementation of reinforcement learning Q-learning technique for FPGA,” IEEE Access, vol. 7, pp. 2782–2798, 2019.
[9] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation,” in Proc. Adv. Neural Inf. Process. Syst. Conf. (NIPS), 1999, pp. 1057–1063.
[10] C. Zhao, X. Gao, Y. Wang, and J. Li, “Efficient multiple-feature learning-based hyperspectral image classification with limited training samples,” IEEE Trans. Geosci. Remote Sens., vol. 54, no. 7, pp. 4052–4062, Jul. 2016.
[11] S. Tadjudin and D. A. Landgrebe, “Covariance estimation with limited training samples,” IEEE Trans. Geosci. Remote Sens., vol. 37, no. 4, pp. 2113–2118, Jul. 1999.
WEI LI received the Ph.D. degree from Xiamen University, China, in 2013. From June 2018 to June 2019, he was a Visiting Scholar with the Department of Computer Science, University of Massachusetts, Boston, MA, USA. He is currently an Associate Professor with the School of Computer and Information Engineering, Xiamen University of Technology, China. He has published over 20 papers in journals and conferences in these areas. His research interests include machine learning, fuzzy modeling and granular computing, big data, and information security. He is a Committee Member of the China Computer Federation.

KAIQIANG ZHANG is currently pursuing the degree with the School of Computer and Information Engineering, Xiamen University of Technology, China. His main research interests include artificial intelligence, granular computing, and computer graphics.

YUMIN CHEN (Member, IEEE) received the M.E. degree in computer application technology from Nanchang University, China, in 2005, and the Ph.D. degree in pattern recognition and intelligence system from Tongji University, China, in 2010. From November 2018 to November 2019, he was a Visiting Scholar, supported by the Education Department of Fujian Province, with the Department of Mathematics and Statistics, Mississippi State University, USA. He is currently a Professor with the College of Computer and Information Engineering, Xiamen University of Technology. He has published over 50 papers in refereed journals and conferences. His research interests include machine learning, rough sets, and granular computing.

CHAO TANG was born in Hefei, Anhui, China. He received the M.S. degree from Shanxi University, Taiyuan, China, in 2009, and the Ph.D. degree in artificial intelligence from Xiamen University, China, in 2014. He is currently an Associate Professor with the School of Artificial Intelligence and Big Data, Hefei University, China. His research interests include machine learning, computer vision, and human action recognition.

XIAOYU MA is currently pursuing the degree with the School of Computer and Information Engineering, Xiamen University of Technology, China. His research interests include machine learning, natural language processing, and granular computing.

YOUMENG LUO is currently pursuing the degree with the School of Computer and Information Engineering, Xiamen University of Technology, China. His research interests include image processing, artificial intelligence, and granular computing.

* * *

[12] Y. Yao, “Granular computing: Past, present and future,” in Proc. IEEE Int. Conf. Granular Comput. (GRC), Hangzhou, China, Aug. 2008, pp. 80–85.

[13] D. Miao, D. Li, Y. Yao, G. Wang, Y. Zhang, W. Wu, J. Liang, D. Yu, and R. Wang, Uncertainty and Granular Computing, 1st ed. Beijing, China: Science Press (in Chinese), Oct. 2011.

[14] Y. Yao, “A triarchic theory of granular computing,” Granular Comput., vol. 1, no. 2, pp. 145–157, 2016.

[15] L. Zhang and B. Zhang, Quotient Space Based Problem Solving: A Theoretical Foundation of Granular Computing. Beijing, China: Tsinghua Univ. Press, 2014.

[16] L. A. Zadeh, “Fuzzy sets and information granularity,” in Advances in Fuzzy Set Theory and Applications, vol. 2, M. Gupta, R. Ragade, and R. Yager, Eds. Amsterdam, The Netherlands: North-Holland, 1979, pp. 3–18.

[17] L. A. Zadeh, “Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic,” Fuzzy Sets Syst., vol. 90, no. 2, pp. 111–127, 1997.

[18] J. Hobbs, “Granularity,” in Proc. IJCAI, Los Angeles, CA, USA, 1985, pp. 432–435.

[19] Z. Pawlak. “Granularity of knowledge, indiscernibility and rough sets,” in Proc. IEEE Int. Conf. Fuzzy Syst., May 1998, pp. 106–110.

[20] T. Lin, “Granular computing on binary relations II: Rough set representations and belief functions,” in Rough Sets in Knowledge Discovery, vol. 1. Berlin, Germany: Physica-Verlag, 1998, pp. 121–140.

[21] R. R. Yager and D. Filev, “Operations for granular computing: Mixing words with numbers,” in Proc. IEEE Int. Conf. Fuzzy Syst., May 1998, pp. 123–128.

[22] Y. Yao, “Granular computing: Basic issues and possible solutions,” in Proc. Joint Conf. Inf. Sci., Atlantic City, NJ, USA, 2000, pp. 186–189.

[23] Y. Yao, “Information granulation and rough set approximation,” Int. J. Intell. Syst., vol. 16, no. 1, pp. 87–104, 2001.

[24] Y. Yao, “A partition model of granular computing,” in Transactions on Rough Sets I (Lecture Notes in Computer Science), vol. 3100. Berlin, Germany: Springer, 2004, pp. 232–253.

[25] Y. Yao, “A unified framework of granular computing,” in Handbook of Granular Computing. Hoboken, NJ, USA: Wiley, 2009.

[26] Y. Yao, “Interpreting concept learning in cognitive informatics and granular computing,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 39, no. 4, pp. 855–866, Aug. 2009.

[27] F. Wang, “Computing with words and a framework for computational linguistic dynamic systems,” (in Chinese), Pattern Recognit. Artif. Intell., vol. 14, no. 4, pp. 377–384, 2001.

[28] G. Wang, F. Hu, H. Huang, and Y. Wu, “A granular computing model based on tolerance relation,” J. China Universities Posts Telecommun., vol. 12, no. 3, pp. 86–90, 2005.

[29] Y. Qian, J. Liang, Y. Yao, and C. Dang, “A multi-granulation rough set,” Inf. Sci., vol. 180, no. 6, pp. 949–970, 2010.

[30] W. W. and Y. Leung, “Theory and applications of granular labelled partitions in multi-scale decision tables,” Inf. Sci., vol. 181, no. 18, pp. 3878–3897, 2011.

[31] W. Zhu and F. Wang, “On three types of covering-based rough sets,” IEEE Trans. Knowl. Data Eng., vol. 19, no. 8, pp. 1131–1144, Aug. 2007.

[32] W. Zhu, “Relationship between generalized rough sets based on binary relation and covering,” in Proc. Beijing Int. Conf. Rough Sets, Fuzzy Theory and Algebraic Hyperstructures, Beijing, China, Oct. 2007, pp. 210–225.

[33] Q. Feng, D. Miao, and Y. Cheng, “Hierarchical decision rules mining,” Expert Syst. Appl., vol. 37, no. 3, pp. 2081–2091, 2010.

[34] W. Li, Y. Chen, H. Ha, and C. Tang, “Using granule to search privacy preserving voice in home IoT systems,” IEEE Access, vol. 8, pp. 31957–31969, 2020.

[35] W. Li, Z. Wei, Y. Chen, C. Tang, and Y. Song, “Fuzzy granular hyperplane classifiers,” IEEE Access, vol. 8, pp. 112066–112077, 2020.

[36] W. Li, Y. Chen, and Y. Song, “Boosted K-nearest neighbor classifiers based on fuzzy granules,” Knowl.-Based Syst., vol. 195, no. 3, pp. 1–13, 2020.

[37] C. Lu, P. Xu, and L. Cong, “Fault diagnosis model based on granular computing and echo state network.” Eng. Appl. Artif. Intell., vol. 94, pp. 1–8, Sep. 2020.

[38] X. Zhu, W. Pedrycz, and Z. Li, “Development and analysis of neural networks realized in the presence of granular data,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 9, pp. 3606–3619, Nov. 2020.

[39] F. Colace, V. Loia, W. Pedrycz, and S. Tomasiello, “On a granular functional link network for classification,” Neurocomputing, vol. 398, pp. 108–116, Feb. 2020.