Boosted stochastic fuzzy granular hypersurface classifier

Boosted stochastic fuzzy granular hypersurface classifier

Wei Li [a], Huosheng Hu [b], Yumin Chen [a], Yuping Song [c,]
[a] School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China
[b] School of Computer Science and Electronic Engineering, Colchester CO4 3SQ, United Kingdom
[c] School of Mathematical Sciences, Xiamen University, Xiamen 361005, China

A B S T R A C T

In this work, we design a boosted stochastic fuzzy granular hypersurface classifier (BSFGHC) to resolve the classification issue of numerical data and non-numerical data (such as information granules) from the standpoint of granular computing. The scheme is divided into three parts: first, we present an adaptive cluster center clustering (ACCC) algorithm to achieve cluster centers of the data and to realize the fuzzy granulation of data parallelly based on Spark, which dramatically improves the granulation efficiency; second, we build a fuzzy granular space, design various fuzzy granular operators and measurement in the space to construct fuzzy granular hypersurfaces, create the loss function, and employ Particle Swarm Optimization (PSO) to resolve the optimal fuzzy granular hypersurfaces; third, we randomly divide the fuzzy granules to train multiple optimal fuzzy granular hypersurfaces and combine with the classification accuracy of fuzzy hypersurfaces and the difficulty of fuzzy granule subset classification to form a boosted fuzzy hypersurface to predict the data comprehensively. Experimental results and theoretical analysis demonstrate the outstanding performance of the method.

1. Introduction

1.1. Motivation

Support vector machine (SVM) is an effective classifier that avoids the traditional process from induction to deduction and realizes efficient "transduction reasoning" from training samples to forecast samples, which greatly simplifies the usual classification and regression problems. Ensemble learning is a great method to improve the performance of classification. Granular computing has the advantage of knowledge discovery. If these ideas can be combined together, the algorithm can improve the performance of classification. So, the fuzzy granular classifier is designed in the granular space based on these ideas.

1.2. Related work

For many years, humans have been trying to find inspiration from human thinking and lots of laws of nature to create corresponding computing models and apply them to life, which has prompted the budding and development of artificial intelligence and immense data computing intelligence [1]. In this process, the research and application of neural networks, evolutionary computing, genetic computing, and swarm intelligence are all very successful examples. Granular computing can simulate human thinking at a higher level [2]. In recent years, especially in responding to the challenges of big data mining, granular computing has received extensive attention from scholars. Granular computing is considered an emerging computing exemplar in artificial intelligence. It separates complex problems into more straightforward issues, which helps researchers to analyze better and resolve issues [3]. Granular computing was first presented by Zadeh and Lin et al. [4,5]. In 1979, Zadeh [6] first believed that fuzzy information granules play an essential role in human reasoning. Zhang Ba and Zhang Ling proposed that humans can discover and analyze the same issue from different granular perspectives, which is the most extensive cognitive feature in human intelligence in 1992, [7]. In 1998, Yager and Filev claimed that the human ability to observe, measure, conceptualize, and reason can be achieved by considering information granulation [8]. In fact, Zadeh presented that granular computing can be regarded as a novel independent research field in the process of intelligent information in 1997 [9]. In the subsequent development, granularity measurement, granularity uncertainty [10–14] and multi-granularity perspective [15–22] have been extensively adopted in different fields such as rough sets, machine learning, complex networks, data mining and knowledge engineering [23–27].

In human granule intelligence, a granule is thought to be a set of things with indistinguishable, connected, or similar functions [9,28–30]. Granulation of the data produces a set of granules. Things with the same properties cannot be separated in classification problems,

and these objects are grouped into a granule due to indistinguishability. In social network studies, latent communities are found by analyzing the connections between nodes, and these communities can be thought of as primary granules in social networks. The granular structure forms the basic computing unit in intelligent information processing based on granularity.

The research on many different theoretical systems, such as fuzzy sets, rough sets, quotient spaces, shadow sets, and conceptual lattices, has been continuously established and improved and has been rapidly developed and applied. The theory of information granulation and information granule construction based on fuzzy sets plays an indispensable or intermediate bridge role in establishing and running through the representation of human language knowledge and the representation of computer digital technology [31]. In fuzzy reasoning and fuzzy control of complex information systems, information granulation based on fuzzy sets shows the superiority and flexibility that other models cannot match. Therefore, granular computing on the basis of fuzzy sets may become an essential tool for intelligent human–computer interaction and ubiquitous computing in artificial intelligence and computational intelligence. It is generally believed that the structure of granular computing on the basis of rough sets theory is a division, and the concept of division here primarily concentrates on the point of view of set theory. The degree to which objects in a collection belong to the group is related to attribute granularity. To characterize the boundary (fuzzy, uncertain) characteristics of sets, Polish scientist Pawlak proposed rough set theory in 1982 [32]. The critical idea is to create a division of the universe by using indistinguishable relations (or equivalence relations), thereby forming concept granules (or indistinguishable equivalence classes) under different attribute granularities, and these concept granules create an approximate space. Based on this approximation space, two exact sets are defined to approximate the target (the set with a fuzzy boundary). When dealing with problems, humans can discover and explore the same issue from various granular worlds and can flexibly jump between different granularities. Inspired by this phenomenon, Zhang Bo and Zhang Ling combined the diverse granular world with the quotient set theory in mathematics and unified them to create a granular world system of quotient space. The quotient space theory uses quotient sets to represent various levels of granularity and establishes the "falsification principle" and "fidelity principle" in the world of various granularities [33]. The shadow set originates from the fuzzy sets, is induced by the fuzzy sets, and is isomorphic to the ternary logic; its quantification levels are 0, 1, [0,1], respectively [31]. The fuzzy relation is simplified by using these three quantization levels to transform the classical fuzzy sets into a three-valued logic of shadow sets.

Besides those mentioned above main granular computing models, researchers have also proposed many other granular computing models, such as the conceptual granular computing model created by Wille et al. combining concept lattice and fuzzy sets theory [34], Lin's covering granular computing model based on covering theory [35], a granular computing model designed by Yao by using set theory [36] and some other new granular computing frameworks based on these granular computing models, etc.

Ensemble learning is a very efficient method to enhance the accuracy of machine learning. Ensemble learning is also a commonly used algorithm in machine learning. In general, ensemble learning usually has better performance than a single classical supervised machine learning method because ensemble learning improves the overall model performance by combining multiple individual learner models. How to combine granular computing with ensemble learning and give full play to their respective advantages to improve machine learning performance has become an important research direction.

Boosting [37] is a typical sequence ensemble learning method, which realizes implicit ensemble learning by building different individual learners and sequentially weighting the samples in the training dataset. The primary thought of boosting is to apply individual learners to the training dataset repeatedly and generate a series of individual learners through a predetermined number of iterations. In 1996, Breiman proposed the Bagging algorithm [38], which introduced random sampling of samples into ensemble learning. In 1997, Adaboost, which has been attracting the attention of the world, appeared [39]. The Adaboost method is very easy and competent. It employs a simple and clear approach to determine the weight of classifiers and the weight of instances and trains multiple classifiers in the process of continuously updating the weight. In 1998, Ho proposed a stochastic subspace classification ensemble [40]. In 2001, Breiman proposed the famous random forest. The algorithm uses random sampling with replacement to collect samples, collects the same number of instances as the initial dataset, and then adopts these samples to train a random tree and repeats this to train different random trees. An ensemble classifier is composed of these random trees [41].

In addition, there are ensemble learning methods based on deep learning. Deep learning has become an essential power in the machine-learning community. Recently, deep learning has greatly enhanced state of the art in speech recognition, object detection, visual object recognition, and other fields [42–46]. Many studies have tried to combine ensemble and deep learning, and many ensemble learning approaches on the basis of deep learning have been presented. For instance, Deep neural decision forests proposed by Kontschieder et al. [47] is a learning method that combines convolutional neural networks and decision forests. Wen et al. [48] designed an ensemble learning approach on the basis of CNN, which was used for facial expression recognition tasks with good results. Qiu et al. presented an ensemble learning approach on the basis of DNNs (Deep Neural Networks), in which the output of DNNs is used as the input of the support vector regression (SVR) that generates the final output [49]. Deng et al. [50] proposed an ensemble learning method based on deep learning, which combines recurrent neural networks, convolutional neural networks, and fully connected neural networks for speech recognition tasks. The gsForest [51] proposed by Zhou et al. is a new method combining DNNs and ensemble learning, which replaces the DNN neurons with a random forest model, in which the output vector of every random forest is used as the output vector of the next layer enter.

Although many ensemble learning methods have been proposed and verified to have good results in many specific application fields, there are still few studies on granular ensemble learning, so it is necessary to propose new granular ensemble learning to improve the effect.

1.3. Organization

The organization of this paper is as follows: introduction is in the first section; next, in the second section, problems are proposed and the flowchart of the algorithm is shown; how to convert data to fuzzy granules is described in the third section; in the fourth and the fifth section, stochastic fuzzy granular hypersurface classifier and boosted stochastic fuzzy granular hypersurface classifier are designed respectively; experimental evaluation is demonstrated and explained in the sixth section; the final part is the conclusion.

2. Problems

There exists a classified system $\mathbb{S} = (X, Y, A, V, h)$. Here, $X = \{x_1, x_2, \ldots, x_n\}$ represents an instance set and $x_i \in \mathbb{R}^m$; $y_i \in \{0, 1, \ldots, l\}$ is the corresponding category of x_i; $Y = \{y_1, y_2, \ldots, y_n\}$ is a category set; $A = \{a_1, a_2, \ldots, a_m\}$ is a non-empty finite set of attributes. V_a is the value range of attribute a and $V = \cup_{a \in A} V_a$; $\tau : X \times A \longrightarrow V$ is an information function that gives an information value to every attribute of objects, namely $\forall a \in A, x \in X$, $\tau(x, a) \in V_a$. The system consists of two parts: learning system and prediction system. The learning system builds a model on the basis of the training data, that is, for the unknown instance x_{n+1}, the prediction system determines the corresponding output y_{n+1} according to the learned model $Y = F(X)$.

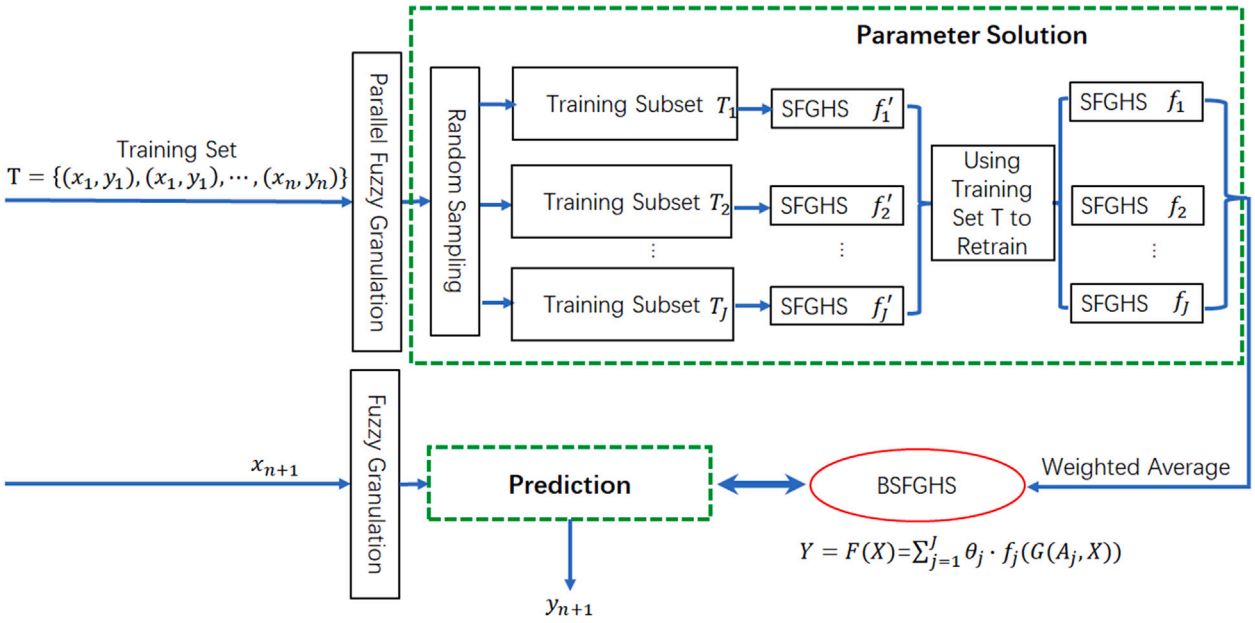

Fig. 1. The overflow of boosted stochastic fuzzy granular hypersurface classifier.

The boosted stochastic fuzzy granular hypersurface classifier consists of two parts: solving parameters and predicting category. In the parameter-solving stage, the cluster centers of the dataset are first calculated, and then the instances are randomly divided into J subsets. The instances of subsets are parallelized into fuzzy granular vectors. In the parameter-solving stage, the cluster centers of the dataset are first calculated, and then the instances are randomly divided into J subsets, and the parallel fuzzy granulation is made into fuzzy granular vectors. Train J fuzzy granular hypersurfaces $f_1', f_2', \ldots, f_J'$ in each fuzzy granular vector subset. Then, the fuzzy granular hypersurface classifiers are further trained with the entire fuzzy granular vector, and the weight of each fuzzy granular vector is corrected according to the classification accuracy of the fuzzy granular hypersurface. After several iterations, according to these fuzzy granular hypersurfaces and their corresponding weights, linear superposition forms an enhanced stochastic fuzzy granular hypersurface classifier. In the prediction stage, given an instance, fuzzy granulation is first performed on it, and then the trained model $F(X)$ is used to calculate its category (see Fig. 1). Here, "stochastic" refers to two factors as follows. First, the BSFGHC needs a cluster approach Adaptive Cluster Center Clustering Algorithm (ACCC) as the basis of fuzzy granulation. When ACCC is executing, initialization seed instances are random. Second, when solving the optimum fuzzy granular hypersurface, the initialization of parameters is also random. So, to describe the process, the term "stochastic" is employed in the title.

3. Convert data to Fuzzy granules

The granularity of human ratiocination and conceptual building is vague. The classical approach of fuzzy information granulation is sourced from calculating the binary relation of all instances, and its computational complexity is high. We propose to first obtain a series of cluster centers of the instance by clustering and then divide the instance set into several subsets. Next, in each instance subset, based on these cluster centers, fuzzy granulation is performed to form fuzzy atomic granules based on atomic attributes. On the basis of these, fuzzy atomic granules with different attributes are constructed to form fuzzy granular vectors so that the fuzzy granulation of each subset can be executed in parallel, and the computational complexity is greatly reduced.

3.1. Adaptive cluster center clustering algorithm

Some classical clustering algorithms (such as k-means) require initializing the cluster centers and pre-specifying the number of cluster centers to achieve results. These approaches rely on the initial parameters, and when the number of cluster centers is inappropriate, it is easy to fall into a locally optimal solution. We present a random cluster center adaptive clustering algorithm, which adaptively chooses the cluster center and the cluster number and optimizes globally. Note that if the standard deviation between clusters is significant, but the standard deviation within groups is slight, the clustering performance is excellent. Therefore, using the objective function $O(C, X) = log\delta^2 - log(\sum_{i=1}^{k} \sigma_i^2) + \sum_{i=1}^{k} \sum_{x \in c_i} \cos(c_i, x)$ as the evaluation can achieve the aim (here, δ represents the standard deviation of a series of cluster centers, σ_i denotes the standard deviation of the instance points in the ith cluster, and k expresses the number of instance clusters). The aim is to continuously adjust the cluster center to make the objective function increase continuously until the maximum number of iterations gets achieved, or the objective function value for several consecutive iterations no longer changes significantly. A set of cluster centers can be obtained in each iteration, and an evaluation set consists of the cluster centers and objective function values of each iteration. If the termination condition is reached, a series of cluster centers corresponding to the maximum objective function value is found from this evaluation set. That is what is asked for. The algorithm flow is listed below.

(1): Remove instances with default attribute values.

(2): Standardize each attribute value of the instance.

(3): Initialise the maximum number of iterations as MaxIter, and the evaluation set E as an empty set, namely $E \leftarrow \phi$ (It consists of the cluster center set and the corresponding objective function value.), and Set the current iteration number $iter = 1$.

(4): Initialise the current cluster center set $C_{iter} = \phi$, select k as the number of cluster centers and an instance x_j as the cluster center randomly, and assign the current number of cluster centers t to 1, i.e., $k \leftarrow 1$, $c_t \leftarrow x_j$, $C_{iter} \leftarrow C_{iter} \cup \{c_t\}$.

(5): The maximum similarity between each instance and the currently existing cluster center can be calculated by $s(x_j) = \max_{c_i \in C_{iter}} \{cos(c_i, x_j)\}$,

Table 1
Adaptive cluster center clustering algorithm.

Input:	the instance set X,
	the maximum iterations $MaxIter$,
	the threshold value ϵ and N
Output:	the optimal cluster center set C^*
1:	Remove instances with missing attribute values.
2:	Normalize instance attribute values to $[0, 1]$.
3:	$E \leftarrow \phi$\\Set the evaluation set (consists of the cluster center set and the objective function value) to the empty set.
4:	$iter \leftarrow 0$\\Initialize the current iteration to zero.
5:	WHILE $(iter \leq MaxIter)$ AND $(\|\|O(C_{iter}, X) - O(C_{iter+N}, X)\|\|^2 > \epsilon)$
6:	$iter \leftarrow iter + 1$.\\Update the current iteration $iter$
7:	$C_{iter} \leftarrow \phi$\\The empty set is assigned to the current cluster center set.
8:	$t \leftarrow 1$\\Assign 1 to the cluster center number.
9:	$c_t \leftarrow RandSelect(X, 1)$\\Select an instance randomly as a cluster center.
10:	$k \leftarrow RandGenClusterNum()$;\\Select a random number k as the cluster center count.
11:	$C_{iter} \leftarrow C_{iter} \cup \{c_t\}$
12:	WHILE $t \leq k$
13:	$\forall j \in \|X\|, s(x_j) = \max_{c_i \in C_{iter}} \{cos(c_i, x_j)\}$;
14:	$p(x_j) = 1 - \frac{s(x_j)^2}{\sum_{i=1}^{n} s(x_i)^2}, j = 1, 2, \ldots, n.$\\ the selected probability of x_j as next cluster center.
15:	p=GenProb();\\Give a probability randomly.
16:	IF $p(x_j) > p$ THEN $\{t \leftarrow t + 1; c_t \leftarrow x_j; C_{iter} \leftarrow C_{iter} \cup \{c_t\};\}$
17:	END WHILE\\ $k \leq K$
18:	$E \leftarrow E \cup \{(C_{iter}, O(C_{iter}, X))\}$\\Update the evaluation set.
19:	END WHILE
20:	$C^* = \arg\max_{C_j \in E} \{O(C_j, X)\}$\\Select the cluster center set with regard to the maximum objective function value in the evaluation set E.

and the probability of being chosen as the next cluster center for each instance can be $p(x_j) = 1 - \frac{s(x_j)^2}{\sum_{i=1}^{n} s(x_i)^2}, j = 1, 2, \ldots, n$.

(6): If x_j is chosen, then $t \leftarrow t + 1$, $c_t \leftarrow x_j$, $C_{iter} \leftarrow C_{iter} \cup \{c_t\}$.

(7): If $t \leq k$, go to **(5)**; otherwise go to **(8)**.

(8): Calculate the objective function $O(C_{iter}, X)$ of this iteration and update the evaluation set $E \leftarrow E \cup \{(C_{iter}, O(C_{iter}, X))\}$.

(9): Modify the current iteration count $iter \leftarrow iter + 1$.

(10): If the number of iterations $iter > MaxIter$ or $\forall N \in (1, ab)$, $\|O(C_a, X) - O(C_b, X)\|^2 < \epsilon$ go to **(11)** (where $b + N < a$ and ϵ is a small positive number), otherwise go to **(4)**.

(11): In the evaluation set E, according to the objective function value in each iteration, the optimal cluster center can be selected, i.e., $C^* = \arg\max_{1 \leq iter \leq MaxIter} O(C_{iter}, X)$, $C^* = \{c_1, c_2, \ldots, c_{|C^*|}\}$ (where $|\cdot|$ expresses the number of elements in the set).

The adaptive clustering principle of global optimization mentioned above has been given. Table 1 gives the pseudo-code of this principle as follows.

3.2. Parallel Fuzzy granulation

On data granulation, the classical method is to serially calculate the similarity between each instance and the rest of the instances to construct a similarity matrix, which does not have parallel granulation conditions. If the cluster center is obtained by clustering first, the similarity between each instance in the subset and each cluster center will be easy to compute. Therefore, parallel granulation can be implemented. Further, the scheme can reduce computational complexity. We can adopt the Spark framework to execute the parallel fuzzy granulation based on the principle below. Suppose that $C = \{c_1, c_2, \ldots, c_k\}$ the set of cluster centers. For $\forall x_i \in X$, $\forall a_t \in A$ and $\forall c_j \in C^*$, the similarity between the instance x_i and the cluster center c_j is computed by

$$h(x_i, c_j, a_t) = 1 - |\tau(x_i, a_t) - \tau(c_j, a_t)| \tag{1}$$

where $0 \leq \tau(x_i, a_t) \leq 1$, $0 \leq \tau(c_j, a_t) \leq 1$. Obviously, $0 \leq h(x_i, c_j, a_t) \leq 1$. The instance x_i and the cluster center c_j induce a fuzzy granule on the attribute a_t which is written by

$$g(x_i, a_t) = \frac{h(x_i, c_1, a_t)}{c_1} + \frac{s(x_i, c_2, a_t)}{c_2} + \cdots + \frac{h(x_i, c_k, a_t)}{c_k} \tag{2}$$

It is also written as follows:

$$g(x_i, a_t) = \int_{c_j \in C} \frac{h(x_i, c_j, a_t)}{c_j} \tag{3}$$

Where "$\int$" is not an integral operation but a union of sets, and "-" represents a separator. In other words, the fuzzy granule $g(x_i, a_t)$ expresses the similarity set between the instance x_i and the cluster centers on the attribute a_t. Its module can be computed by

$$|g(x_i, a_t)| = \sum_{j=1}^{k} h(x_i, c_j, a_t) \tag{4}$$

Four operators of fuzzy granules are designed as follows. $\forall e, f \in R$, the operator $\cup$ and $\cap$ can be computed by

$$e \cup f = \lambda ef + (1 - \lambda)(e + f - ef) \tag{5}$$

$$e \cap f = (ef)^{1-\gamma}(e - ef)^{\gamma} \tag{6}$$

where $\lambda, \gamma \in [0, 1]$ are parameters. Next, according to the two operators, other operators of fuzzy granules can be computed. For $\forall x, x' \in X$, $\forall a \in A$, the operations on two fuzzy granules induced by the instance x and x' respectively are written as below:

$$g(x, a) \vee g(x', a) = \int_{c_j \in C} \frac{h(x, c_j, a) \cup h(x', c_j, a)}{c_j} \tag{7}$$

$$g(x, a) \wedge g(x', a) = \int_{c_j \in C} \frac{h(x, c_j, a) \cap h(x', c_j, a)}{c_j} \tag{8}$$

$$g(x, a) - g(x', a) = \int_{c_j \in C} \frac{h(x, c_j, a) - h(x', c_j, a)}{c_j} \tag{9}$$

$$g(x, a) \oplus g(x', a) = g(x, a) \vee g(x', a) - g(x, a) \wedge g(x', a) \tag{10}$$

The distance between the two fuzzy granules can be computed by

$$d(g(x,a),g(x',a)) = \frac{1}{|C|}\sum_{c_j\in C}\frac{|g(x,a)\oplus g(x',a)|}{|g(x,a)\vee g(x',a)|} \tag{11}$$

For $\forall T \subseteq A$ and $T = \{a_1, a_2, \ldots, a_{|T|}\}$ with $|T| \le |A|$, the fuzzy granular vector formed by x on T is written by

$$\begin{aligned} &G(x,T)\\ &= \frac{g(x,a_1)}{a_1} + \frac{g(x,a_2)}{a_2} + \cdots + \frac{g(x,a_{|T|})}{a_{|T|}}\\ &= \int_{a_t\in T}\frac{g(x,a_t)}{a_t} = \int_{a_t\in T}\int_{c_j\in C}\frac{h(x,c_j,a_t)}{c_j} \end{aligned} \tag{12}$$

Its module is computed by

$$|G(x,T)| = \sum_{t=1}^{|T|}|g(x,a_t)| \tag{13}$$

The operations on two fuzzy granular vectors are calculated as follows:

$$G(x,T)\vee G(x',T) = \int_{a_t\in T}\frac{g(x,a_t)\vee g(x',a_t)}{a_t} \tag{14}$$

$$G(x,T)\wedge G(x',T) = \int_{a_t\in T}\frac{g(x,a_t)\wedge g(x',a_t)}{a_t} \tag{15}$$

$$G(x,T) - G(x',T) = \int_{a_t\in T}\frac{g(x,a_t) - g(x',a_t)}{a_t} \tag{16}$$

$$G(x,T)\oplus G(x',T) = \int_{a_t\in T}\frac{g(x,a_t)\oplus g(x',a_t)}{a_t} \tag{17}$$

Similar to the distance between two fuzzy granules, the distance on two fuzzy granular vectors is defined by

$$d(G(x,T),G(x',T)) = \frac{1}{|T|*|C|}\sum_{a\in T}\frac{|G(x,a)\oplus G(x',a)|}{|G(x,a)\vee G(x',a)|}. \tag{18}$$

Theorem 1. *For $\forall x_1, x_2 \in X$, $\forall T \subseteq A$, $\forall a \in T$, and $\forall$ the cluster $c_j \in C$, the distance of the two fuzzy granular vector assures*

$$0 \le d(G(x_1,T),G(x_2,T)) \le 1. \tag{19}$$

Proof. According as the definition of fuzzy granules, we can give $g(x_1,a) = \int_{c_j\in C}\frac{h(x_1,c_t,a)}{c_j}$ and $g(x_2,a) = \int_{c_j\in C}\frac{h(x_2,c_t,a)}{c_j}$. Thanks to Eq. (1), we have $0 \le h(x_1,c_j,a) \le 1$ and $0 \le h(x_2,c_j,a) \le 1$. Because of $|g(x_1,a)| = \sum_{c_j\in C} h(x_1,c_j,a)$, $0 \le |g(x_1,a)| \le |C|$ is established. Similarly, we can also obtain $0 \le |g(x_2,a)| \le |C|$. Furthermore, according to the definition of the fuzzy granular vector $G(x_1,T) = \int_{a_t\in T}\frac{g(x,a_t)}{a_t}$ and $|G(x_1,T)| = \sum_{t=1}^{|T|}|g(x_1,a_t)|$, it is easy to obtain $0 \le |G(x_1,T)| \le |T| * |C|$. Similarly, we have $0 \le |G(x_2,T)| \le |T| * |C|$. Because of $g(x_1,a)\oplus g(x_2,a) = g(x_1,a)\vee g(x_2,a) - g(x_1,a)\wedge g(x_2,a)$, we can obtain $0 \le \sum_{a\in T}\frac{|g(x_1,a)\oplus g(x_2,a)|}{|g(x_1,a)\vee g(x_2,a)|} \le |T| * |C|$. Suppose we want to make a left-inverse $|T|^{-1} * |C|^{-1}$ of $|T| * |C|$,so that we can solve a inequality by left-multiplying each side to obtain $0 \le \frac{1}{|T|*|C|}\sum_{a\in T}\frac{|g(x_1,a)\oplus g(x_2,a)|}{|g(x_1,a)\vee g(x_2,a)|} \le 1$, i.e., $0 \le d(G(x_1,T),G(x_2,T)) \le 1$. □

Theorem 2. *For $\forall x \in X$, the subset T of the attribute Q and T satisfy $Q \subseteq T \subseteq A$. Suppose we have two fuzzy granular vectors $G(x,Q)$ and $G(x,T)$, which are induced by x and defined as Q and T, respectively. Then we can give $|G(x,Q)| \le |G(x,T)|$.*

Proof. For $\forall a_t \in Q$ and $G(x,Q) = \int_{a_t\in Q}\frac{g(x,a_t)}{a_t}$, $G(x,T) = \int_{a_t\in T}\frac{g(x,a_t)}{a_t}$ can be obtained thanks to $Q \subseteq T$ and $a_t \in T$. Because $T \subseteq Q \subseteq A$ is established, for $a \in Q$, we can obtain $a \in T$ and $|Q| \le |T|$. In that way, if $g(x,a) \in G(x,Q)$, then $g(x,a) \in G(x,T)$. In sum, the inequality $|G(x,Q)| \le |G(x,T)|$ is established. □

Table 2
Metric and granulation of the instance set.

$X,C\|A,Y,\nu$	a_1	a_2	a_3	y	ν
x_1	0.30	0.20	0.10	0.20	0.50
x_2	0.10	0.20	0.30	0.20	0.50
x_3	0.30	0.40	0.20	0.20	0.50
x_4	0.30	0.20	0.40	0.40	0.50
c_1	0.25	0.35	0.15	–	–
c_2	0.15	0.25	0.25	–	–

This is an instance of granulation exhibited in Table 2. Given an instance set $X = \{x_1, x_2, x_3, x_4\}$, an attribute set $A = \{a_1, a_2, a_3\}$, a category set $Y = \{y_1, y_2, y_3, y_4\}$, a cluster center set $C = \{c_1, c_2\}$, and the parameter $\nu = 0.5$, the granulation computation is as follows.

We take the instance x_1 as an instance. The similarities between the instance x_1 and the cluster center c_1 and c_2 respectively on the attribute a_1, a_2, a_3 are as below:

$$\begin{aligned} &h(x_1,c_1,a_1) = 1 - |\tau(x_1,a_1) - \tau(c_1,a_1)| = 1 - |0.30 - 0.25| = 0.95,\\ &h(x_1,c_2,a_1) = 1 - |\tau(x_1,a_1) - \tau(c_2,a_1)| = 1 - |0.30 - 0.15| = 0.85,\\ &h(x_1,c_1,a_2) = 1 - |\tau(x_1,a_2) - \tau(c_1,a_2)| = 1 - |0.20 - 0.35| = 0.85,\\ &h(x_1,c_2,a_2) = 1 - |\tau(x_1,a_2) - \tau(c_2,a_2)| = 1 - |0.20 - 0.25| = 0.95,\\ &h(x_1,c_1,a_3) = 1 - |\tau(x_1,a_3) - \tau(c_1,a_3)| = 1 - |0.10 - 0.15| = 0.95,\\ &h(x_1,c_2,a_3) = 1 - |\tau(x_1,a_3) - \tau(c_2,a_3)| = 1 - |0.10 - 0.25| = 0.85, \end{aligned}$$

According to Eq. (3), the fuzzy granules induced by x_1 on the attribute a_1, a_2, and a_3 are as below:

$$\begin{aligned} &g(x_1,a_1) = \frac{0.95}{c_1} + \frac{0.85}{c_2},\\ &g(x_1,a_2) = \frac{0.85}{c_1} + \frac{0.95}{c_2},\\ &g(x_1,a_3) = \frac{0.95}{c_1} + \frac{0.85}{c_2}. \end{aligned}$$

In the same way, we can obtain the fuzzy granules induced by the instance x_2 on the attribute set A as follows:

$$\begin{aligned} &g(x_2,a_1) = \frac{0.85}{c_1} + \frac{0.95}{c_2},\\ &g(x_2,a_2) = \frac{0.85}{c_1} + \frac{0.95}{c_2},\\ &g(x_2,a_3) = \frac{0.85}{c_1} + \frac{0.95}{c_2}. \end{aligned}$$

According to $e\cup f = \lambda ef + (1-\lambda)(e+f-ef)$ and Eq. (7) (here, $\lambda = 0.5$), we can give (see Box I). According to $e\cap f = (ef)^{1-\gamma}(e-ef)^{\gamma}$ (here, $\gamma = 0.5$) and Eq. (8), we have

$$\begin{aligned} &g(x_1,a_1)\wedge g(x_2,a_1)\\ &= \sum_{j=1}^{2}\frac{(h(x_1,c_j,a_1)h(x_2,c_j,a_1))^{1-\gamma}(h(x_1,c_j,a_1) - h(x_1,c_j,a_1)h(x_2,c_j,a_1))^{\gamma}}{c_j}\\ &= \frac{(0.95*0.85)^{1-0.5}(0.95 - 0.95*0.85)^{0.5}}{c_1}\\ &\quad + \frac{(0.85*0.95)^{1-0.5}(0.85 - 0.85*0.95)^{0.5}}{c_2}\\ &\approx \frac{0.34}{c_1} + \frac{0.34}{c_2} \end{aligned}$$

$$\begin{aligned} &g(x_1,a_1)\oplus g(x_2,a_1)\\ &= g(x_1,a_1)\vee g(x_2,a_1) - g(x_1,a_1)\wedge g(x_2,a_1)\\ &= \frac{0.90 - 0.34}{c_1} + \frac{0.90 - 0.34}{c_2} = \frac{0.56}{c_1} + \frac{0.56}{c_2} \end{aligned}$$

$$\begin{aligned} &|g(x_1,a_1)\oplus g(x_2,a_1)|\\ &= |g(x_1,a_1)\vee g(x_2,a_1) - g(x_1,a_1)\wedge g(x_2,a_1)|\\ &= |\frac{0.56}{c_1} + \frac{0.56}{c_2}| = 1.12. \end{aligned}$$

In the same way, we have

$$\begin{aligned} &|g(x_1,a_2)\oplus g(x_2,a_2)| = |\frac{0.56}{c_1} + \frac{0.56}{c_2}| = 1.12.\\ &|g(x_1,a_3)\oplus g(x_2,a_3)| = |\frac{0.56}{c_1} + \frac{0.56}{c_2}| = 1.12. \end{aligned}$$

$$
\begin{aligned}
& g(x_1, a_1) \vee g(x_2, a_1) \\
&= \sum_{j=1}^{2} \frac{\lambda h(x_1, c_j, a_1) h(x_2, c_j, a_1) + (1-\lambda)(h(x_1, c_j, a_1) + h(x_2, c_j, a_1) - h(x_1, c_j, a_1) h(x_2, c_j, a_1))}{c_j} \\
&= \frac{0.5 * 0.95 * 0.85 + (1-0.5)(0.95 + 0.85 - 0.95 * 0.85)}{c_1} \\
&+ \frac{0.5 * 0.85 * 0.95 + (1-0.5)(0.85 + 0.95 - 0.85 * 0.95)}{c_2} \\
&= \frac{0.90}{c_1} + \frac{0.90}{c_2}
\end{aligned}
$$

Box I.

So far we can obtain the distance between x_1 and x_2 on the attribute set A with parameter $\lambda = \gamma = 0.5$ as follows:

$$d(G(x_1, A), G(x_2, A)) = \frac{1}{|A| * |C|} \sum_{a \in A} \frac{|g(x_1, a) \oplus g(x_2, a)|}{|g(x_1, a) \vee g(x_2, a)|} \approx 0.62.$$

4. Stochastic Fuzzy granular hypersurface classifier

Recall the problem that we can suppose there are n instances that are nonlinearly classified. Which decision surface is the best in infinitely decision surfaces with zero error in the space? The classification interval should be as large as possible for the decision surface with zero error. We transfer the problem to fuzzy granular space to solve it. According to the atom attribute, the classification system converts each instance into a fuzzy granule. The fuzzy granules build a fuzzy granular vector. Then, category and fuzzy granular vector make up the fuzzy granular vector instance set. As well known, Support Vector Machine (SVM) is a good algorithm for classification and regression. In SVM, the researcher designed the decision hyperplane function y=w $\cdot$ x + b to classify where w and b can be obtained by training samples. Inspired by the SVM, we introduced the vectors α and β to construct a fuzzy granular hypersphere equation on fuzzy granular space. Therefore, the classification process can be converted to finding the fuzzy granular vectors α and β in Eq. (20) in the fuzzy granular space. Then the decision function (Eq. (21)) computes the category to which the fuzzy granular vector belongs, i.e., the instance category. The computational process follows: First, the instance is fuzzily granulated. Next, the problem is converted into solving Eq. (20) to find the fuzzy granular vector α and β. A fuzzy granular hypersurface Equation (see Eq. (20)) is introduced and solves the minimum value of the loss function to obtain the maximum interval hypersurface, i.e., the optimal solution of α and β. Before solving the problem, we need to make the following assumptions:

(1) The error between prediction and ground truth satisfies Gaussian distribution. (2)There is a linearly classified solution. It can be found that α and β make the fuzzy granular hypersurface become the fuzzy granular hypersurface with the most significant interval. The solution process is implemented using PSO in granular space. The above process is defined as:

Definition 1. Suppose $\mathbb{S} = (X, Y, A, V, h)$ is a classified system. Here $X = \{x_1, x_2, \ldots, x_n\}$ represents an instance set and $x_i \in \mathbb{R}^m$. $Y = \{y_1, y_2, \ldots, y_n\}$ denotes a category set and $y_i \in \{0, 1, \ldots, l\}$ is the output corresponding to x_i. $A = \{a_1, a_2, \ldots, a_m\}$ expresses a nonempty infinite set of the attribute. $V = \cup_{a \in A} V_a$ and V_a represents the value field of the attribute a. $\tau : X \times A \longrightarrow V$ is an information function and it gives a value for each attribute of each object, namely, for $\forall a \in A, x \in X$, $\tau(x, a) \in V_a$, we have a pair $\langle G(x_i, A), y_i \rangle (i = 1, 2, \ldots, n.)$ consists of a fuzzy granular vector on A and a category. The fuzzy granular vector instance set defined by the pairs above is $G(X, A) = \{\langle G(x_i, A), y_i \rangle | \forall x_i \in X\}$. A fuzzy granular hypersphere equation on fuzzy granular space can be defined by

$$L(\alpha, \beta) = \min_{\alpha, \beta} \{ \sum_{x_i \in X} (log|\alpha \wedge G(A, x_i) \vee \beta| - y_i)^2 + \lambda \cdot |\alpha|^2 \}. \tag{20}$$

Here, α and β denote fuzzy granular vectors and λ represents a real number. Let α^* and β^* be the optimization solution of Eq. (20), then the classified decision function is written by

$$f(x) = \arg\min_{0 \le t \le l} \{ |log|\alpha^* \wedge G(A, x) \vee \beta^*| - t| \} \tag{21}$$

4.1. Solving parameters

We employ PSO to solve Eq. (20). The principle is to discover the approximate optimal solution through information sharing and cooperation among individuals in the group. The superiority is that it is effortless and does not have too many parameters to alter. During the process, every particle is a fuzzy granular vector with two characteristics: speed and position (speed expresses the speed of movement, and position denotes the direction of movement). Every particle explores the optimal solution freely in the space, registers it as the current individual extremum, and communes the individual extremum with other particles. Then, the optimal individual extremum discovered is the current global optimal solution of the whole particle swarm. The speed and position of particles are altered according to the current individual extreme value discovered by themselves and the current global optimal solution communicated by the whole particle swarm.

The process can be as follows:

(1): Initialize a set of random fuzzy granular vectors $\gamma_1, \gamma_2, \ldots, \gamma_N$ (where $\gamma_i = (\alpha_i, \beta_i)$) as random solution and their corresponding to speed $v_1, v_2, \ldots, v_N$. Set the current iteration $Iter = 0$ and the maximum iterations $MaxIter$.

(2): Compute the function value corresponding to each solution $L(\gamma_i)$ and set $pBest_i = \gamma_i$, $i = 1, 2, \ldots, N$.

(3): Select the global optimal position $gBest = \arg\min_{1 \le j \le N} \{L(pBest_j)\}$

(4): According to the historical optimal position of every particle and the global optimal position, the speed and the position of each particle can be updated as follows:

$$v_i = r_0 \cdot v_i + c_1 \cdot r_1 \cdot (pBest_i - \gamma_i) + c_2 \cdot r_2 \cdot (gBest - \gamma_i) \tag{22}$$

$$\gamma_i^* = \gamma_i \tag{23}$$

$$\gamma_i = \gamma_i + v_i \tag{24}$$

(5): Evaluate the fitness function value to update the optimal historical position. If $L(\gamma_i^*) > L(\gamma_i)$, then $pBest_i = \gamma_i$; otherwise, $pBest_i = \gamma_i^*$.

(6): Update the current iteration $Iter = Iter + 1$.

Table 3
The algorithm of solving parameters on stochastic fuzzy granular hypersurface classifier.

Input:	The fuzzy granular vector set $G(A, X)$ and maximum iteration $MaxIter$.
Output:	The solution of fuzzy granular hypersurface α^* and β^*
1:	Initialize a group of fuzzy granular vectors (random solution) $\gamma_1, \gamma_2, \ldots, \gamma_N$. (where $\gamma_i = (\alpha_i, \beta_i)$) and their corresponding to speed $v_1, v_2, \ldots, v_N$. Set the current iteration $Iter = 0$ and maximum iteration $MaxIter$.
2:	Compute the function value corresponding to each solution $L(\gamma_i)$ and set $pBest_i = \gamma_i,\ i = 1, 2, \ldots, N$
3:	Select the global optimal position $gBest = \arg\min_{1 \le j \le N}\{L(pBest_j)\}$
4:	According as the particle's historical optimal position and global optimal position, the velocity and position of each particle are altered. $v_i = r_0 \cdot v_i + c_1 \cdot r_1 \cdot (pBest_i - \gamma_i) + c_2 \cdot r_2 \cdot (gBest - \gamma_i)$ $\gamma_i = \gamma_i + v_i$
5:	Modify the historical optimal position of the particle by evaluating the fitness value. If $L(\gamma_i^*) > L(\gamma_i)$, then $pBest_i = \gamma_i$; Otherwise, $pBest_i = \gamma_i^*$.
6:	Set the current iteration $Iter = Iter + 1$.
7:	If $Iter > MaxIter$ go to 8; Otherwise, go to 2.
8:	Output the global optimal solution $gBest$.

(7): If $Iter > MaxIter$, go to **(8)**, otherwise go to **(2)**.

(8): Output the global optimal result $gBest$.

The pseudocode on solving parameters is as follows (see Table 3).

5. Boosted stochastic Fuzzy granular hypersurface classifier

We propose integrating multiple fuzzy granular hypersurface classifiers into a robust classifier, namely the boosted stochastic fuzzy granular hypersurface classifier. The basic idea is that the classifier's fuzzy granular vector that is wrongly classified will be employed to train the next classifier. The wrongly classified fuzzy granular vector is emphasized to train, and multiple weak classifiers (where the weak classifier is the fuzzy granular hypersurface classifier mentioned above) are combined into a robust classifier.

5.1. The principle

To achieve adaptive enhancement, boosted stochastic fuzzy granular hypersurface consists of fuzzy granular hypersurfaces. It is an iterative algorithm, and a new weak classifier is added in each iteration until the classification error rate is low enough and the program ends. The principle is as follows. Every fuzzy granular vector is given a weight representing its probability of being chosen into the training set by a specific classifier. If a fuzzy granular vector has been distinguished accurately, its probability is decreased when building the next training set. Instead, if a fuzzy granular vector is not accurately distinguished, its weight is enhanced. Thus, the fuzzy granular hypersurface can concentrate on those fuzzy granular vectors that are harder to differentiate and more informative. Specifically, let each fuzzy granular vector have the same weight. In every iteration, the weight of the fuzzy granular vector is adjusted continuously according to the last classification error. For instance, for the kth iteration, we select the fuzzy granular vectors according to these weights and train the fuzzy granular hypersurface classifier. Then, according to this classifier, the weight of the fuzzy granular vector that is wrongly classified is enhanced, and that of the fuzzy granular vectors correctly distinguished is decreased instead. Then, the set of fuzzy granular vectors that are modified weights is adopted to train the next fuzzy granular hypersurface. The entire process goes on repeatedly. Each fuzzy granular hypersurface classifier is divided according to a series of different attributes. When the algorithm ends, on the one hand, the classification result is obtained; on the other hand, according to the weight of each weak classifier, the importance of the corresponding attribute set can be obtained, which provides an essential basis for data dimensionality reduction. The process is as follows:

(1): Compute the cluster centers of the instances.

(2): Randomly divide the instance set into several instance subsets.

(3): Get the fuzzy granular vector subset from the instance and cluster centers.

(4): Summarize the subset of fuzzy granular vectors into the full set of fuzzy granular vectors.

(5): Initialize weights of all fuzzy granular vectors.

(6): In the jth $(j = 1, 2, \ldots, J)$ iteration, the attributes are randomly selected to obtain the attribute subset A_j and the fuzzy granular hypersurface $f_j(G(A_j, X))$ can be learned on the fuzzy granular vector set $G(A_j, X)$.

(7): Compute the classified error rate of $f_j(G(A_j, X))$ on the full fuzzy granular vector set.

(8): Alter the weight of the weak classifier $f_j(G(A_j, X))$ according to the classified error rate.

(9): Alter the weights of the fuzzy granular vectors according to the weight of the weak classifier $f_j(G(A_j, X))$.

(10): Repeat **(6)** to **(9)** until the iteration meets the maximum iteration.

(11): Return all fuzzy granular hypersurfaces and their weights.

The algorithm is designed according to the principle above and is demonstrated in Table 4.

5.2. Prediction

Given an instance x and a cluster center set C^*, a set of $G(A_j, x)$, $j = 1, 2, \ldots, J$ on x can be obtained. After that, calculate the category of x by the trained model $F(x) = \sum_{j=1}^{J} \theta_j \cdot f_j(G(A_j, x))$ (see Table 5).

6. Experimental evaluation

This section carries out an experimental study on the boosted stochastic fuzzy granular hypersurface classifier. Remarkably, the datasets and the experimental setup is described first. Then, several algorithms are compared on accuracy and computational complexity. Meanwhile, the results are demonstrated, and the analysis is also given.

The algorithms are implemented in MATLAB on the workstation with ASUS TUF RTX3080 GPU with 64 GB DRAM. The employed datasets are Dry Bean, Anuran Calls, and BLE RSSI from the University of California Irvine Machine Learning Repository. To evaluate the performance of noise data, we construct another three datasets on the basis of them, i.e., Dry Bean with the noise of 1%, Anuran Calls with a noise of 1%, and BLE RSSI with a noise of 1%. These datasets are exhibited in Table 6. 10-fold cross-validation was employed in the experiments. Namely, data of 70% chosen arbitrarily is used as the training set, and data of 30% left is adopted as the test set. We implement an evaluation of the divided data. The operation is echoed ten times. The average executed time and accuracy is as the performance evaluation. The parallel fuzzy granulation based on Spark proposed in the paper is compared with the serial fuzzy granulation to evaluate the computational complexity. The result is described in Fig. 2. We applied Back Propagation (BP), Support Vector Machine (SVM), Long Short-Term Memory (LSTM), and BSFGHC to the datasets above to get accuracy and the root mean square. The number of cluster centers is an essential parameter affecting classification performance. The relationship between the number of cluster centers, the average accuracy, and the average root mean squared error is analyzed (see Figs. 3–5).

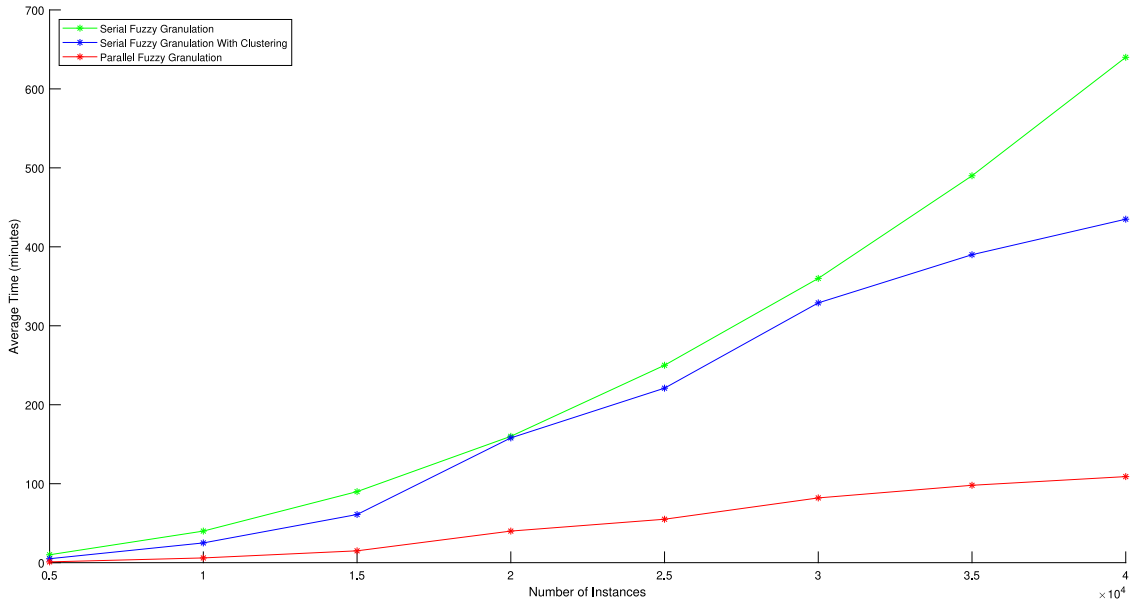

Fig. 2. Comparison of time complexity.

Table 4
Boosted stochastic fuzzy granular hypersurface classified algorithm.

Input:	The instance set X, The maximum iteration J and the threshold value δ
Output:	Boosted stochastic fuzzy granular hypersurface classifier
1:	Compute the cluster center of all instances.
2:	Divide instances set into multi instances subset.
3:	Get the fuzzy granular vector subset by calculating the instance and cluster center in the instance subset.
4:	Summarize the fuzzy granular vector subset to a full fuzzy granular vector set.
5:	Initialize the iteration $j = 1$ and the distributed weights of the fuzzy granular vector $w_{j1} = \frac{1}{n}, i = 1, 2, \ldots, n$.
6:	Select the attributes randomly to make up the attribute subset $A_j \subseteq A, j = 1, 2, \ldots, J.$ and the fuzzy granular hypersurface classifiers $f_j(G(A_j, x_i)) : G \to \{0, 1, \ldots, l\}, j = 1, 2, \ldots, J, i = 1, 2, \ldots, n.$ can be trained on this subset.
7:	Compute the classified error of $f_j(G(A_j, x))$ by $e_j = \sum_{i=1}^{N} w_{ji} I(f_j(G(A_j, x_i)) \neq y_i)$
8:	Update the coefficient of the fuzzy granular hypersurface by $\theta_j = \frac{1}{2} ln \frac{1-e_j}{e_j}$.
9:	The weights of the fuzzy granular vectors are updated as follows: $w_{(j+1)i} = \frac{w_{ji}}{z_j} \cdot h_{ji}$ $h_{ji} = \begin{cases} e^{-\theta_j}, & if \quad \lvert f_j(G(A_j, x_i)) - y_i \rvert < \delta \\ e^{\theta_j}, & if \quad \lvert f_j(G(A_j, x_i)) - y_i \rvert \geq \delta \end{cases}$ $z_j = \sum_{i=1}^{N} (w_{ji} \cdot h_{ji})$, δ is a small positive number.
10:	$j = j + 1$.
11:	If $j \leq J$, repeat 6 to 10; otherwise go to 12.
12:	Return the boosted stochastic fuzzy granular hypersurface classifier $F(x) = \sum_{j=1}^{J} \theta_j \cdot f_j(G(A_j, x))$.

Table 5
Prediction algorithm.

Input: the instance x, the cluster set C^*, and parameters θ_j, $f_j(\cdot)$, $j = 1, 2, \ldots, J$.
Output: the category y of the instance x.
1: Get the fuzzy granular $G(A_j, x)$, $j = 1, 2, \ldots, J$. of x by computing the similarity between x and the cluster centers. 2:Calculate the category of x by $F(x) = \sum_{j=1}^{J} \theta_j \cdot f_j(G(A_j, x))$

The parallel fuzzy granulation approach based on Spark is as follows.

(1): Read data from HDFS datasets, create RDDs, and employ the distributed cluster resource scheduler to allocate computing resources to the created job tasks.

(2): Slice the original data vectorize the data object, and call the cache operation of RDD to place the data in memory.

(3): Input cluster centers and broadcasts to each node.

(4): Compute the similarity between each instance and the cluster center at each node.

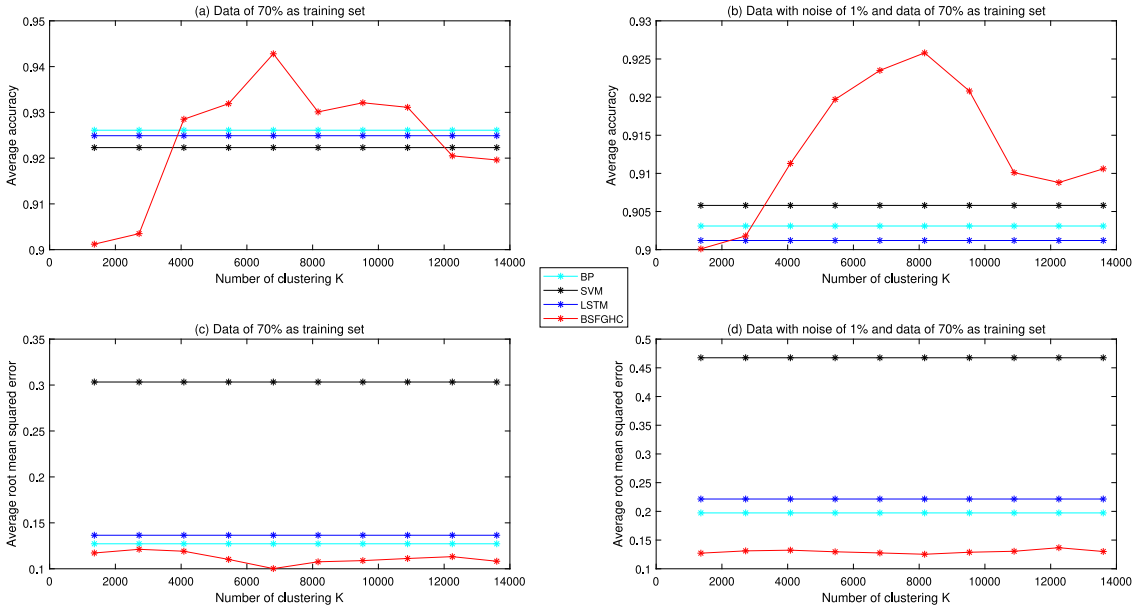

Fig. 3. Evaluation on the dataset DryBean.

Table 6
Datasets from UCI machine learning repository.

Dataset	Number of instances	Number of attributes
Dry Bean	13 611	17
Dry Bean with Noise	13 611	17
Anuran Calls	7195	22
Anuran Calls with Noise	7195	22
BLE RSSI	23 570	5
BLE RSSI with Noise	23 570	5
BLE RSSI	23 570	5
Crowdsourced Mapping	10 546	29
Motion Capture Hand Postures	78 095	38
Internet Firewall Data	65 532	12

(5): Form <key,value > in the form of <cluster center, fuzzy granule> at each node.

(6): Implement local fuzzy granulation by ReduceByKey and broadcast them at each node.

(7): If all instances are fuzzily granulated, go to **(8)**; otherwise, go to **(4)**.

(8): Get fuzzy granular vectors by Catch.

(9): Write the result to HDFS.

As shown in Fig. 2, the green curve denoting the serial fuzzy granulation is getting steeper, the blue curve representing the serial fuzzy granulation with clustering changes more gently, and the red curve expressing parallel fuzzy granulation proposed in the paper changes most gently. In other words, as the number of instances enhances, the time complexity of the parallel fuzzy granulation changes the least, while that of the other two methods is greatly enhanced, especially the traditional serial fuzzy granulation. For instance, when the number of instances is 10187, the serial fuzzy granulation requires about 90 min, the serial fuzzy granulation with clustering needs about 61 min, and the parallel fuzzy granulation is achieved only within 15 min; that is, the serial fuzzy granulation with clustering and the parallel fuzzy granulation reduce time complexity by 32.22% and 83.33% compared with the serial fuzzy granulation, respectively. It can be seen that clustering and parallel processing are essential to improve the efficiency of fuzzy granulation.

Fig. 3 demonstrates the average accuracy of the classification and the average root means squared error for BP, SVM, LSTM, and BSFGHC using the dataset DryBean. As shown in Fig. 3(a), BP, SVM, and LSTM achieved an average accuracy of 92.61%, 92.49%, and 92.23%, respectively, while BSFGHC just got 94.28% when the number of clustering $K = 6805$ (i.e., 1.80%, 1.94%, 2.22% improvement). On the dataset with the noise of 1%, the four algorithms above were reduced by 2.48%, 2.06%, 2.29%, and 1.80%, respectively, as demonstrated in Fig. 3(b). BSFGHC outperforms the other three algorithms. Fig. 3(c) indicates the four algorithms' average root mean squared error. BP, SVM, LSTM got the average root error of 0.1272, 0.3033, and 0.1365, but BSFGHC reached 0.1002 when $K = 6805$. That is, BSFGHC was reduced by 21.23%, 66.96%, and 26.59%, respectively. Compared with the other three methods, the average root mean square error of BSFGHC is the smallest in the noise dataset, as shown in Fig. 3(d).

Next, Anuran Calls is adopted as a test dataset. In this dataset, the feature of the voice from the frog is extracted by MFCC, and Family is an attribute of the category. As shown in Fig. 4(a), BP, SVM, and LSTM got an average accuracy of 96.58%, 97.21%, and 95.78%, respectively, while BSFGHC achieved the maximum accuracy of 98.12% (i.e., improvement 1.59%, 0.94%, and 2.44% respectively). On the noisy data, as exhibited in Fig. 4(b), the maximum accuracy of BSFGHC is 97.11%, which is 2.46%, 2.03%, and 3.43% higher than that of BP, SVM, and LSTM, respectively. On the root mean square error index, as exhibited in Fig. 4(c), BSFGHC got the minimum value of 0.0052, which was 95.94%, 95.29%, and 96.09% lower than BP, SVM, and LSTM, respectively. On the noise dataset, as demonstrated in Fig. 4(d), BSFGHC decreased by 55.39%, 49.69%, and 63.36%, respectively, compared with BP, SVM, and LSTM.

We applied the algorithms to the dataset RSSI, and the results are as follows. On the evaluation of accuracy, as shown in Fig. 5(a), BSFGH improved 1.61%, 2.35%, and 0.92% compared with BP, SVM, and LSTM, respectively, when $K = 11785$. On the noise data, BSFGH was enhanced by 3.28%, 3.92%, and 2.50%, respectively, compared with the three algorithms above. Although the four algorithms decreased, BSFGH is still better than the other three algorithms. On the root mean square error, as exhibited in Fig. 5(c), BSFGH achieved 0.0248, while BP, SVM, and LSTM achieved 0.0465, 0.0515, and 0.0323 respectively (i.e., 46.67%, 51.84%, and 23.22% improvement). Meanwhile, on the noise data, BSFGH still outperforms the other three algorithms; that is, BSFGH improved by 70.99%, 76.44%, and 56.70%, respectively, on the root mean square error (see Fig. 5(d)).

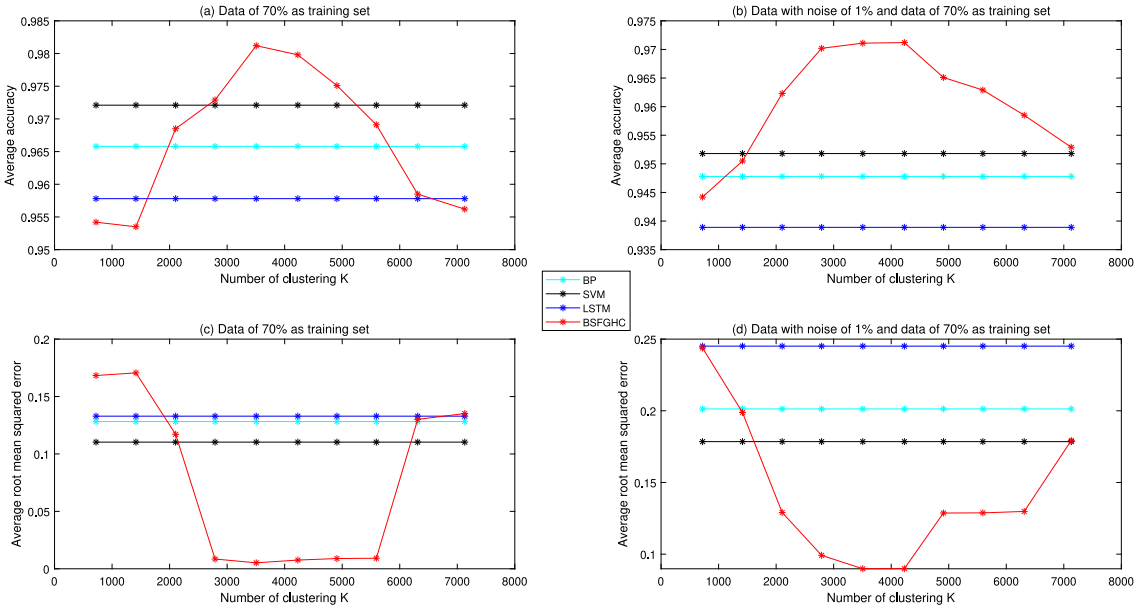

Fig. 4. Evaluation on the dataset Anuran Calls.

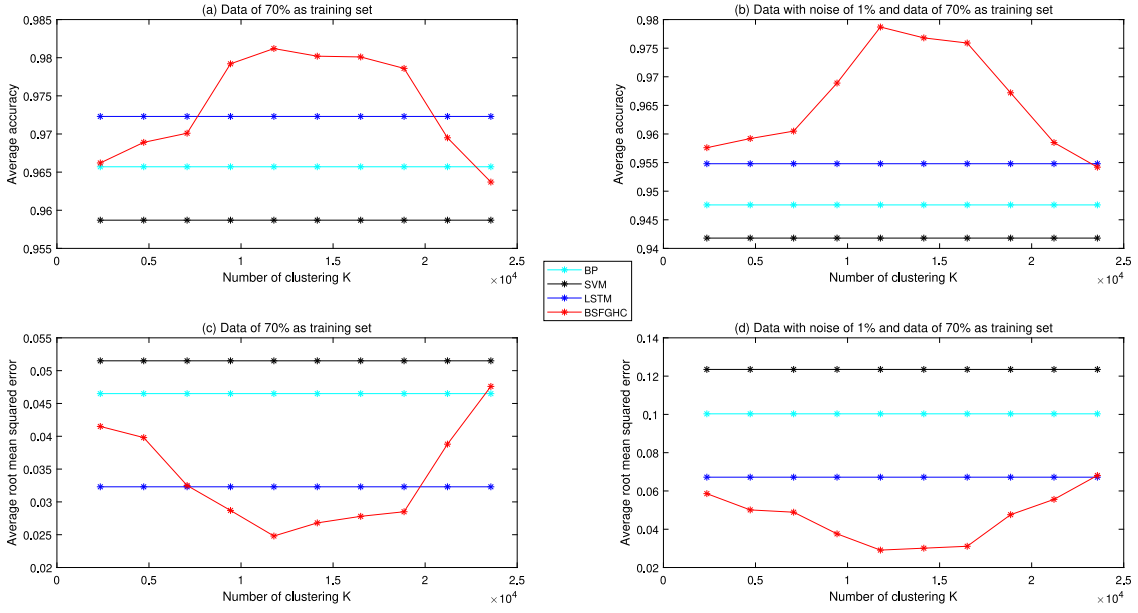

Fig. 5. Evaluation on the dataset RSSI.

We also compared the performance of [52](EEFC),[53](RBGC),[54](FGHC), [55](BKNC),[56](IFC-BD),[57](IGBC), SFGHC, and BSFGHC on the Crowdsourced Mapping dataset which has 10,546 instances and 29 attributes. As demonstrated in Fig. 6, BSFGHC, SFGHC, and IFC-BD got an average accuracy of 93.00%, 92.74%, and 92.71%, respectively, while the other five approaches achieved an average accuracy under 90%; Meanwhile, BSFGHC got the accuracy of 94.8% and its performance is higher than the other methods; in the ROC Area, BSFGHC, SFGHC, and IFC-BD got 0.984, 0.979, and 0.971 respectively, the other methods are under 0.97; the similar situation happens in measuring on PRC Area. In addition, the cost time is evaluated on this dataset. BSFGHC and SFGHC need 321 s and 280 s to execute the training process, while the other algorithms cost about 200 s on average. This is mainly because fuzzy granulation and PSO in our methods cover much time.

As shown in Fig. 7, the performance is compared further in the dataset Motion Capture Hand Postures. As mentioned in Table 6, there are 78095 instances and 38 attributes. BSFGHC, SFGHC, and IFC-BD reached average accuracy of 93.10%, 91.04%, and 92.31%, respectively. The other five algorithms got no more than 90.00%. BSFGHC is slightly better than IFC-BD. SFGHC is slightly lower than IFC-BD and more higher than the other five approaches. The BSFGHC also has the highest average recall, and it enhances by 0.85% 6.27% than IFC-BD and EEFC, respectively. In ROC Area and PRC Area, BSFGHC and SFGHC are also slightly better than the other six schemes.

To verify the performance, these algorithms mentioned above are verified in Internet Firewall Data, as shown in Fig. 8. In the dataset,

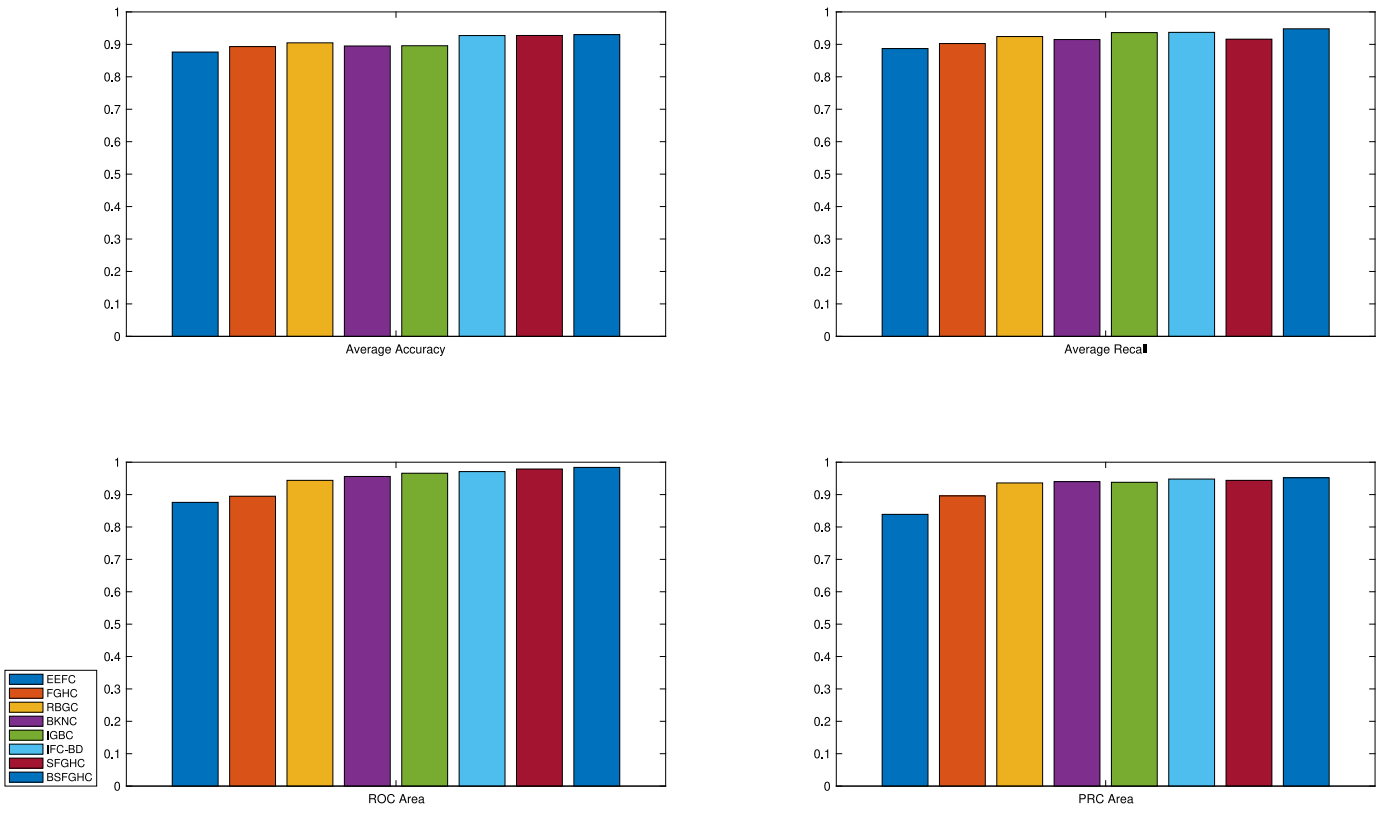

Fig. 6. Evaluation on the dataset Crowdsourced Mapping.

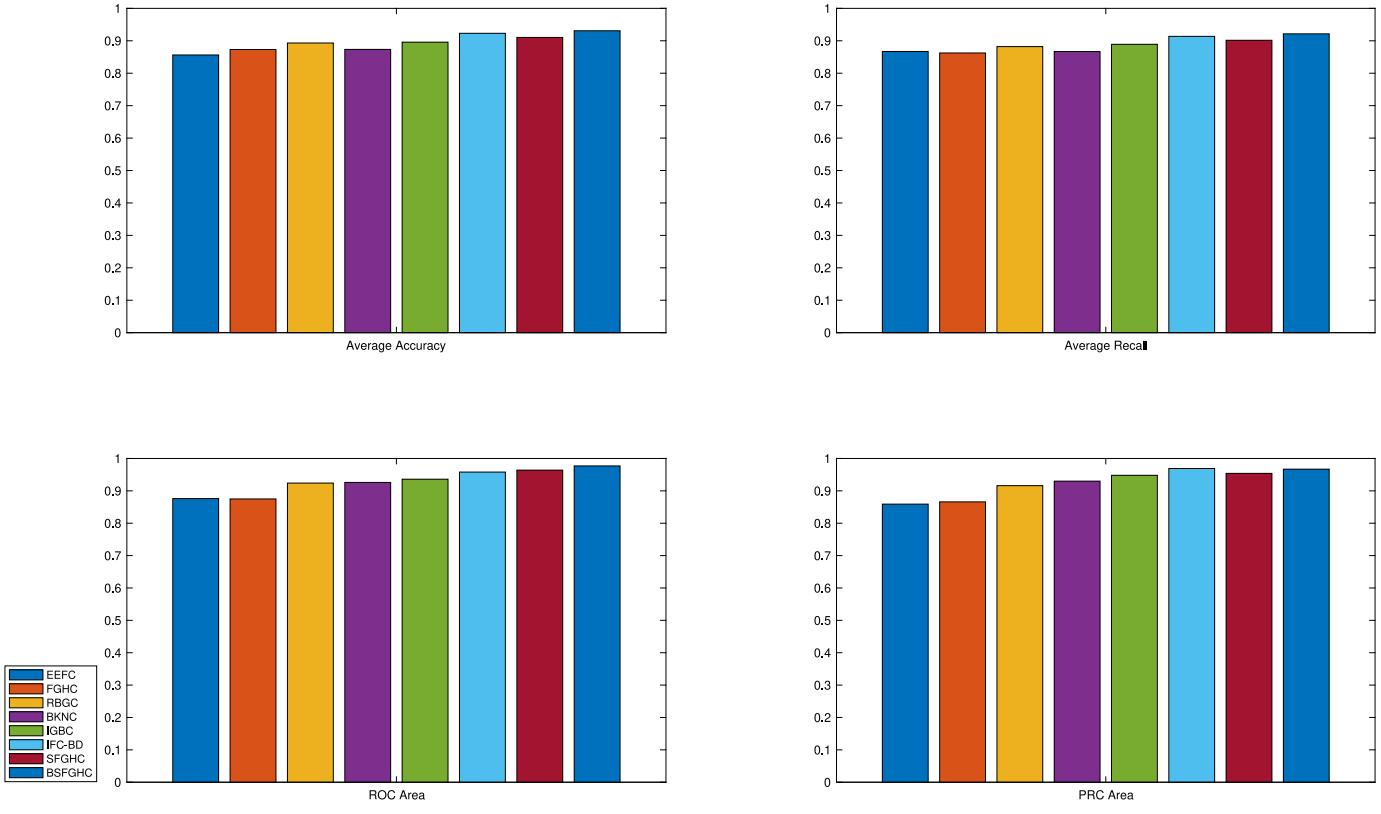

Fig. 7. Evaluation on the dataset Motion Capture Hand Postures.

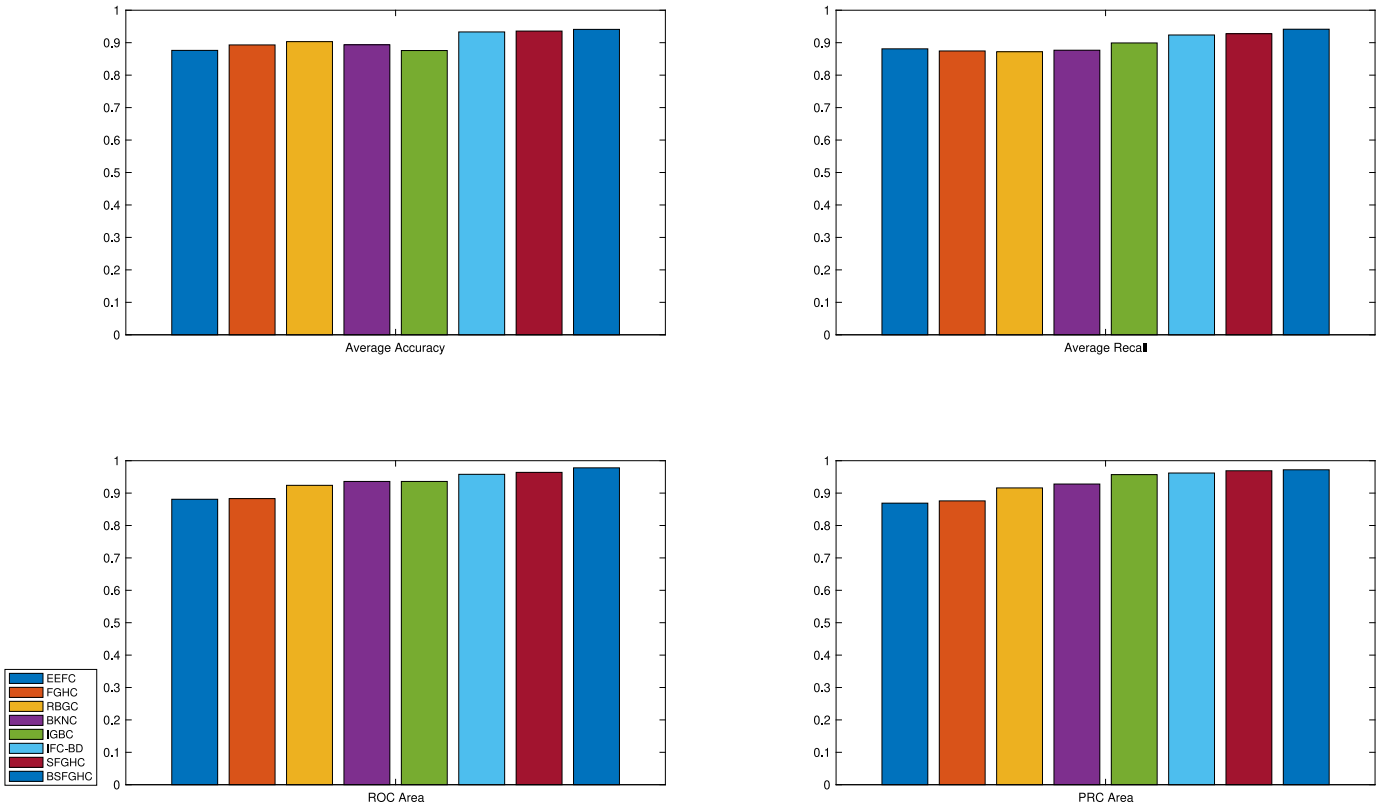

Fig. 8. Evaluation on the dataset Internet Firewall.

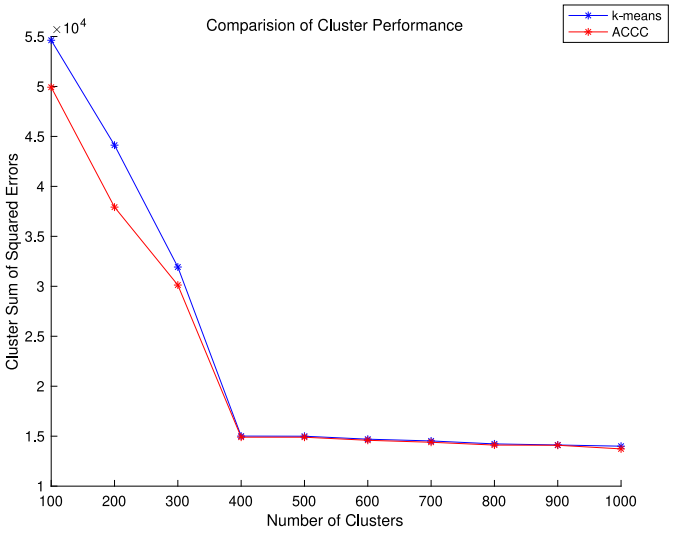

Fig. 9. Cluster Evaluation on the dataset Crowdsourced Mapping.

there are a total of 12 features, with the "action" feature serving as the class label, and the classification task involves predicting one of four classes: "allow", "action", "drop", or "reset-both". BSFGHC and SFGHC got average accuracy of 94.10% and 93.58%, respectively. The two algorithms perform better than the other six algorithms. BSFGHC reached average recall of 94.15% and improved by 1.45% and 1.93% than SFGHC and IFC-BD, respectively. When considering the ROC Area and PRC Area metrics, BSFGHC outperforms the remaining seven methods.

In addition, the cluster performance is compared between k-means and ACCC. As shown in Fig. 9, the sum of squared error is sharply decreased when the number of clusters is no more than 400. After that, the curve dropped slowly. The sum of the squared error of ACCC is less than that of k-means in the experiments, which expresses that ACCC performs better than k-means. Furthermore, ACCC can also automatically select the number of clusters corresponding to the minimum sum of squared error as the optimal value. K-means is a local optimum algorithm, and ACCC is a global optimum one.

The above analysis shows that BSFGH performs better than BP, SVM, and LSTM in many datasets. Especially on noisy data, BSFGH is more robust than the other three algorithms. This is due to two aspects: On the one hand, the fuzzy granulation process contains the idea of global comparison, defeating the interference of noise partly. On the other hand, BSFGHC starts from the weak learning algorithm and repeatedly learns to obtain a series of weak classifiers (also known as basic classifiers) and then unites these weak classifiers to constitute a robust classifier.

7. Conclusion

Classical classifiers are oriented to numerical computation and do not involve set operations. This work proposes a boosted stochastic fuzzy granular hypersurface classifier from a fuzzy set perspective. First, the cluster center global optimization algorithm is presented, which gives the number of clusters adaptively and overcomes the disadvantage of the traditional clustering approach that the number of cluster centers needs to be pre-specified. Second, a fuzzy granulation method is presented, which can realize fuzzy granulation parallelly based on cluster centers and Spark to achieve a fast transformation from data to fuzzy granules. Third, on the basis of these, we integrate multiple fuzzy granular hypersurface classifiers into a boosted

fuzzy granular hypersurface classifier to achieve data classification. The proposed scheme can implement the classification of data and classify information granules. In future, we will regard the local granulation and the geometric structure of the data and combine the manifold idea to improve the classification performance.

CRediT authorship contribution statement

Wei Li: Conceptualization, Methodology, Project administration, Resources, Software, Validation, Writing – original draft, Writing – review & editing. **Huosheng Hu:** Methodology, Supervision, Writing – review & editing. **Yumin Chen:** Data curation, Validation, Visualization. **Yuping Song:** Conceptualization, Data curation, Funding acquisition, Methodology, Resources, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work was supported in part by Fujian Natural Science Foundation of China (2022J011233), in part by Ministry of Education of the People's Republic of China (23YJAZH067), in part by China Scholarship Council (202308350042), in part by Xiamen Municipal Science and Technology Bureau (2023CXY0409), in part by the National Natural Science Foundation of China (61976183).

References

[1] M. Mohadesi, R. Abbas, Biodiesel conversion modeling under several conditions using computational intelligence methods, Environ. Prog. Sustain. Energy 37 (1) (2018) 562–568.
[2] W. Pedrycz, J.V. Oliveira, A development of fuzzy encoding and decoding through fuzzy clustering, IEEE Trans. Instrum. Meas. 57 (4) (2008) 829–837.
[3] W. Pedrycz, Granular Computing: Analysis and Design of Intelligent Systems, CRC Press, 2018.
[4] L.A. Zadeh, Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems, Artif. Intell. Mag. 2 (1) (1998) 23–25.
[5] T. Lin, Granular computing on binary relations II: Rough set representations and belief functions, in: Rough Sets in Knowledge Discovery, 1998, pp. 121–140.
[6] L.A. Zadeh, Fuzzy sets and information granularity, in: M. Gupta, R. Ragade, R. Yager (Eds.), Advances in Fuzzy Set Theory and Applications, North-Holland Publishing Co., Amsterdam, 1979, pp. 3–18.
[7] B. Zhang, L. Zhang, Theory and Applications of Problem Solving, North-Holland, Amsterdam, 1992.
[8] R.R. Yager, D. Filev, Operations for granular computing: Mixing words with numbers, in: Proceedings of 1998 IEEE International Conference on Fuzzy Systems, 1998, pp. 123–128.
[9] L. Zadeh, Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy Sets and Systems 90 (1997) 111–127.
[10] J. Hobbs, Granularity, in: Readings in Qualitative Reasoning About Physical Systems, 1990, pp. 542–545.
[11] Y.H. Qian, J.Y. Liang, W.Z. Wu, C.Y. Dang, Information granularity in fuzzy binary GrC model, IEEE Trans. Fuzzy Syst. 19 (2) (2010) 253–264.
[12] B.W. Xu, Y.M. Zhou, H.M. Lu, An improved accuracy measure for rough sets, J. Comput. System Sci. 71 (2) (2005) 163–173.
[13] Y.Y. Yao, L.Q. Zhao, A measurement theory view on the granularity of partitions, Inform. Sci. 213 (2012) 1–13.
[14] H.Y. Zhang, W.X. Zhang, C.L. Mei, Entropy of interval-valued fuzzy sets based on distance and its relationship with similarity measure, Knowl. Based Syst. 22 (6) (2009) 449–454.
[15] Y.H. Qian, S.Y. Li, J.Y. Liang, Z.Z. Shi, F. Wang, Pessimistic rough set based decisions: A multigranulation fusion strategy, Inform. Sci. 264 (2014) 196–210.
[16] Y. Qian, J.Y. Liang, Y.Y. Yao, C.Y. Dang, MGRS: A multi-granulation rough set, Inform. Sci. 180 (6) (2010) 949–970.
[17] Y.H. She, X.L. He, On the structure of the multigranulation rough set model, Knowl. Based Syst. (2012) 81–92.
[18] W.Z. Wu, Y. Leung, Theory and applications of granular labelled partitions in multi-scale decision tables, Inform. Sci. 181 (18) (2011) 3878–3897.
[19] W.Z. Wu, Y. Leung, Optimal scale selection for multi-scale decision tables, Internat. J. Approx. Reason. 54 (8) (2013) 1107–1129.
[20] W.Z. Wu, Y.H. Qian, T.J. Li, S.M. Gu, On rule acquisition in incomplete multi-scale decision tables, Inform. Sci. 378 (2017) 282–302.
[21] W.H. Xu, Q.R. Wang, S.Q. Luo, Multi-granulation fuzzy rough sets, J. Intell. Fuzzy Systems 26 (3) (2014) 1323–1340.
[22] X.B. Yang, Y.S. Qi, X.N. Song, J.Y. Yang, Test cost sensitive multigranulation rough set: Model and minimal cost selection, Inform. Sci. 250 (2013) 184–199.
[23] S. Calegari, D. Ciucci, Granular computing applied to ontologies, Internat. J. Approx. Reason. 51 (4) (2010) 391–409.
[24] D. Ciucci, D. Dubois, A map of dependencies among three-valued logics, Inform. Sci. 250 (2013) 162–177.
[25] D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets, Int. J. General Syst. 17 (2–3) (1990) 191–209.
[26] W.Z. Wu, L. Yee, J.S. Mi, Granular computing and knowledge reduction in formal contexts, IEEE Trans. Knowl. Data Eng. 21 (10) (2008) 1461–1474.
[27] J.T. Yao, A.V. Vasilakos, W. Pedrycz, Granular computing: Perspectives and challenges, IEEE Trans. Cybern. 43 (6) (2013) 1977–1989.
[28] J.Y. Liang, Y.H. Qian, Information granules and entropy theory in information systems, Sci. China Ser. F: Inf. Sci. 51 (10) (2008) 1427–1444.
[29] W. Pedrycz, Relational and directional aspects in the construction of information granules, IEEE Trans. Syst. Man Cybern. 32 (5) (2002) 605–614.
[30] W. Pedrycz, A. Bargiela, Granular clustering: A granular signature of data, IEEE Trans. Syst. Man Cybern. 32 (2) (2002) 212–224.
[31] W. Pedrycz, Knowledge-Based Clustering: From Data to Information Granules, Wiley Interscience, 2005.
[32] Z. Pawlak, Rough sets, Int. J. Comput. Inf. Sci. 11 (5) (1982) 341–356.
[33] D.Q. Miao, S.D. Hu, Uncertain analysis based on granular computing, J. Northwest Univ. (Nat. Sci. Ed.)(in Chinese) 49 (4) (2019) 487–495.
[34] R. Wille, Restructuring lattice theory: An approach based on hierarchies of concepts, in: International Conference on Formal Concept Analysis, vol. 83, Springer, Berlin, Heidelberg, 2009, pp. 314–339.
[35] T. Lin, Granular computing on binary relations I: Rough set representations and belief functions, in: Rough Sets in Knowledge Discovery, vol. 1, 1998, pp. 107–121.
[36] Y.Y. Yao, A partition model of granular computing, in: Transactions on Rough Sets I. Lecture Notes in Computer Science, vol. 3100, 2004, pp. 232–253.
[37] R. Schapire, The strength of weak learnability, Mach. Learn. 5 (2) (1990) 197–227.
[38] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140.
[39] Y. Freund, R.E. Schapire, A decision–Theoretic generalization of on–Line learning and an application to boosting, J. Comput. System Sci. 55 (1) (1997) 119–139.
[40] T.K. Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 832–844.
[41] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[42] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521 (2015) 436–444.
[43] K. Gopinath, C. Desrosiers, H. Lombaert, Learnable pooling in graph convolutional networks for brain surface analysis, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2) (2022) 864–876.
[44] Q. Ma, S. Li, G.W. Cottrell, Adversarial joint-learning recurrent neural network for incomplete time series classification, IEEE Trans. Pattern Anal. Mach. Intell. 44 (4) (2022) 1765–1776.
[45] C. Yan, B. Gong, Y. Wei, Y. Gao, Deep multi-view enhancement hashing for image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 43 (4) (2021) 1445–1451.
[46] C. Huang, Y. Li, C.C. Loy, X. Tang, Deep imbalanced learning for face recognition and attribute prediction, IEEE Trans. Pattern Anal. Mach. Intell. 42 (11) (2020) 2781–2794.
[47] P. Kontschieder, M. Fiterau, A. Criminisi, S.R. Bulo, Deep neural decision forests, in: Proceedings of the IEEE International Conference on Computer Vision, vol. 83, Santiago, 2015, pp. 1467–1475.
[48] G.H. Wen, Z. Hou, H.H. Li, D.Y. Li, L.J. Jiang, E. Xun, Ensemble of deep neural networks with probability-based fusion for facial expression recognition, Cogn. Comput. 9 (2017) 597–610.
[49] X.H. Qiu, L. Zhang, Y. Ren, P.N. Suganthan, G. Amaratunga, Ensemble deep learning for regression and time series forecasting, in: IEEE Symposium on Computational Intelligence in Ensemble Learning, IEEE, Orlando, FL, USA, 2014, pp. 1–6.

[50] L. Deng, J.C. Platt, Ensemble deep learning for speech recognition, in: 15th Annual Conference of the International Speech Communication Association, Singapore, 2014, pp. 14–18.
[51] Z. Zhou, J. Feng, Deep forest, Natl. Sci. Rev. 6 (1) (2019) 74–86.
[52] M. Pratama, W. Pedrycz, E. Lughofer, Evolving ensemble fuzzy classifier, IEEE Trans. Fuzzy Syst. 26 (5) (2018) 2552–2567.
[53] C. Fu, W. Lu, W. Pedrycz, J. Yang, Rule-based granular classification: A hypersphere information granule-based method, Knowl.-Based Syst. 194 (2020).
[54] W. Li, Z. Wei, Y. Chen, C. Tang, Y. Song, Fuzzy granular hyperplane classifiers, IEEE Access 8 (2020) 112066–112077.
[55] W. Li, Y. Chen, Y. Song, Boosted K-nearest neighbor classifiers based on fuzzy granules, Knowl.-Based Syst. 195 (2020).
[56] F. Aghaeipoor, M.M. Javidi, A. Fernandez, IFC-BD: An interpretable fuzzy classifier for boosting explainable artificial intelligence in big data, IEEE Trans. Fuzzy Syst. 30 (3) (2022) 830–840.
[57] X. Hu, W. Pedrycz, K. Wu, Y. Shen, Information granule-based classifier: A development of granular imputation of missing data, Knowl.-Based Syst. 214 (2021).

Wei Li is an associate professor and master's supervisor with the School of Computer and Information Engineering at the Xiamen University of Technology, Xiamen, China. He is also a member of China Computer Federation and China Granular Computing and Knowledge Discovery Committee. His research interests include Artificial Intelligence, Computer Graphics, and Granular Computing. He received a Ph.D. degree in Basic Theory of Artificial Intelligence at Xiamen University, China, in 2013. He was an academic visitor with the computer science department at the University of Massachusetts Boston, U.S.A, from June 2018 to June 2019. Now he is also an academic visitor with the school of computer science and electronic engineering at the University of Essex. Contact him at drweili@hotmail.com.

Huosheng Hu is a Professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K. His research interests include robotics, mechatronics, human–robot interaction, embedded systems, and pervasive computing. He is a Founding Member of the IEEE Robotics and Automation Society Technical Committee on Networked Robots and a fellow of the Institution of Engineering and Technology and the Institute of Measurement and Control, U.K. He is an Executive Editor of International Journal of Mechatronics and Automation.

Yumin Chen is a professor at the Xiamen University of Technology and a tutor of Ph.D. students at Fuzhou University. He received his Ph.D. from Tongji University, China, in 2010. He was a post-doctor at the Electronic Science and Technology University of China from 2014 to 2017. He is a committee member of Rough Sets and Soft Computing of China, a member of CCF, a CAA member, and a member of CAAI. His research interests include artificial intelligence, machine learning, pattern recognition, and rough sets.

Yuping Song is an associate professor in the School of Mathematical Sciences at Xiamen University. Her research interests include discrete differential geometry, computer graphics, and artificial intelligence. She has a Ph.D. in Pure Mathematics from Peking University. Contact her at ypsong@xmu.edu.cn.