

# Identifying Outliers via Local Granular-Ball Density

Xinyu Su, Xiwen Wang, Dezhong Peng<sup>✉</sup>, Xiaomin Song, Huiming Zheng, and Zhong Yuan<sup>✉</sup>

**Abstract**—Existing density-based outlier detection methods process data at the single-granularity level of individual samples, requiring pairwise distance calculations between all samples and exhibiting high sensitivity to noise. The single-granularity-based processing paradigm fails to mine the information at multiple levels of granularity in data, and most of these methods ignore the potential uncertainty information in data, such as fuzziness, resulting in an inability to effectively detect potential outliers in data. As a novel granular computing method, Granular-Ball Computing (GBC) is characterized by its multi-granularity and robustness, which makes it able to make up for the above drawbacks well. In this study, we propose local Granular-Ball Density-based Outlier (GBDO) detection to improve the performance of the density-based methods. In GBDO, we first identify the  $k$ -similarity Granular-Ball (GB) neighborhoods of each GB via the fuzzy relations among them. Subsequently, the local reachability similarity density of the GBs is calculated through the reachability similarity we defined. Finally, the local GB outlier factors of the samples are calculated based on the local reachability similarity density of the GBs. We adopt a multi-granularity processing paradigm using GBs as the basic units, which reduces computational complexity and improves robustness to noisy data by leveraging the multi-granularity nature of GBs. The experimental results demonstrate the effectiveness of GBDO by comparing it with state-of-the-art methods. The source code and datasets are publicly available at <https://github.com/Mxeron/GBDO>.

**Index Terms**—Fuzzy set theory (FST), granular computing (GrC), granular-ball computing (GBC), outlier detection, unsupervised learning.

## I. INTRODUCTION

OUTLIERS refer to individual instances or clusters of instances whose attributes significantly differ from those of the overall dataset, either globally or locally. Outliers typically consist of measurement, recording, sampling errors,

Received 14 May 2024; revised 11 March 2025 and 24 April 2025; accepted 3 June 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62306196 and Grant 62372315; in part by the Open Research Project of National Key Laboratory of Fundamental Algorithms and Models for Engineering Numerical Simulation, Sichuan Science and Technology Program under Grant 2024YFHZ0089, Grant 2024NSFTD0049, and Grant 2024ZDZX0004; in part by Chengdu Science and Technology Project under Grant 2023-XT00-00004-GX; and in part by the Fundamental Research Funds for the Central Universities under Grant YJ202245. (Corresponding author: Zhong Yuan.)

Xinyu Su, Xiwen Wang, and Zhong Yuan are with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: [suxinyu@stu.scu.edu.cn](mailto:suxinyu@stu.scu.edu.cn); [wangxiwen124@stu.scu.edu.cn](mailto:wangxiwen124@stu.scu.edu.cn); [yuanzhong@scu.edu.cn](mailto:yuanzhong@scu.edu.cn)).

Dezhong Peng is with the College of Computer Science, Sichuan University, Chengdu 610065, China, and also with Tianfu Jincheng Laboratory, Chengdu 610093, China (e-mail: [pengdz@scu.edu.cn](mailto:pengdz@scu.edu.cn)).

Xiaomin Song and Huiming Zheng are with Sichuan National Innovation New Vision UHD Video Technology Company Ltd., Chengdu 610095, China (e-mail: [songxiaomin@uptcsc.com](mailto:songxiaomin@uptcsc.com); [michaelzheng@uptcsc.com](mailto:michaelzheng@uptcsc.com)).

Digital Object Identifier 10.1109/TNNLS.2025.3578074

or exceptional values, among other factors [1]. Outlier detection refers to discovering, mining, and detecting outliers present in a dataset, and an outlier is also referred to as an anomaly. In some fields, it is often more interesting and valuable to mine the outliers in the data than to analyze the normal samples in it. As a basic and critical task in data mining, outlier detection has been applied to many real-world tasks, such as medical diagnostic systems [2] and fraud detection [3].

Based on different assumptions about the existence of outliers, outlier detection methods can be categorized into statistical-based, distance-based, density-based, and clustering-based methods. Among them, density-based methods start from the idea of region density and assume the density of the location where the outlier should be small [4], [5], [6]. The best known of these is the Local Outlier Factor (LOF) [4]. The idea of LOF deeply influences subsequent scholars, and many improved LOF methods have been proposed [7], [8]. In place of focusing on the LOF, there are several novel density-based outlier detection methods. Yuan et al. [9] defined the fuzzy-rough density, and based on this, the outlier factor of the samples is calculated. Naghavi Nozad et al. [10] proposed a batchwise density-based clustering method for local outlier detection. Li et al. [11] proposed a robust outlier detection method based on the changing rate of directed density ratio.

Most of the existing density-based methods generally rely on fine-grained single processing units (i.e., individual data samples as basic processing units). To calculate local density, the Euclidean distance or other distance metrics of all the sample pairs must be calculated and traversed, resulting in an algorithm complexity of  $O(n^2)$ , which is not efficient enough. This fine-grained processing also makes the methods susceptible to noise. Furthermore, these density-based methods often ignore the potential multi-granularity information and the uncertainty information such as fuzziness in the data to assist in constructing more discriminative outlier scores or outlier factors, so further exploration and research are needed. The idea of multi-granularity has been proven to help improve the performance of the methods and has been applied to fields such as scene text recognition [12],  $k$  Nearest Neighbor ( $k$ NN) classification [13], person reidentification [14], and video action recognition [15].

Recently, Xia et al. [16] proposed a novel data granulation method, namely, Granular-Ball Computing (GBC), and it has gained great attention in the area of data mining. In particular, a Granular-Ball (GB) is a simple data structure derived from raw data. A GB contains one or more samples, thus inherently possessing multi-granularity characteristics. Using multi-granularity GBs as the basic processing unit of the model rather than individual samples can make the

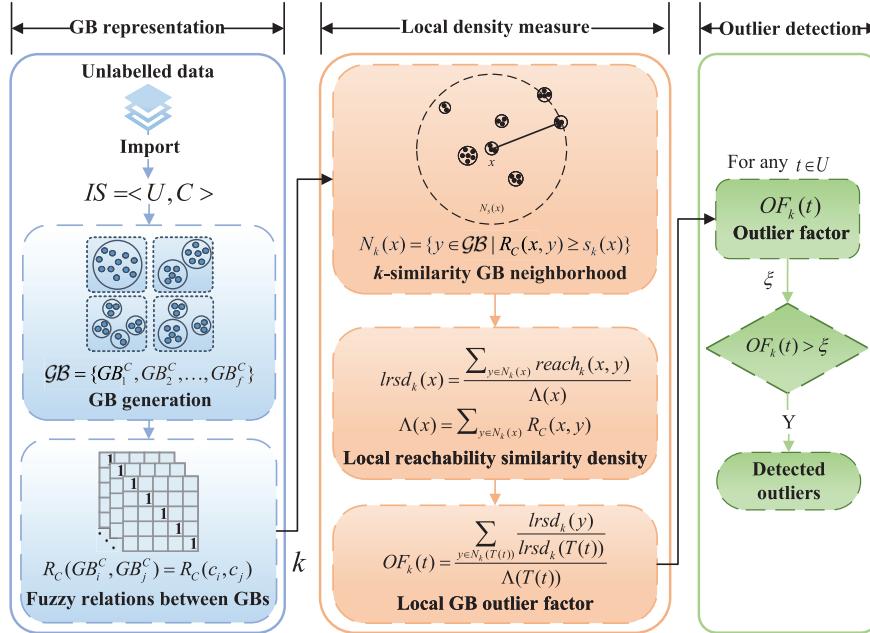


Fig. 1. Framework diagram of GBDO.

GBC-based methods more efficient and robust [16], [17], [18]. In terms of data representation, GBC is different from traditional sample point-based methods. It uses GBs to cover and represent data, thus establishing a novel and efficient computing framework. This new multi-granularity computing paradigm has been extensively studied in areas such as classifiers [19], [20], rough sets [16], [21], outlier detection [5], [22], [23], [24], [25], clustering [26], [27], reinforcement learning [28], and graph coarsening [29]. Before model learning, GBs are first generated on raw data, enabling a comprehensive coverage of the dataset. The coarse-grained nature of GBs makes the model resistant to noise and more efficient. As a novel research field, current research on GBs mainly focuses on two aspects, one is the GBs' generation [18], [20], [27], i.e., how to generate high-quality GBs efficiently. On the other hand, the application of GBs, i.e., how to use GBs to solve real-world tasks [5], [23], [24], [27].

Fuzzy Set Theory (FST) constitutes a crucial tool for handling uncertainty information inherent in data, with its foundation centered on the membership function. Through the mapping of the membership function, an object within a set no longer belongs or does not belong to that set. Instead, it quantifies the membership degree for each object, describing the degree to which an object belongs to the current set. Recently, FST has been widely used in tasks such as clustering [30], [31], rule induction [32], feature selection [33], [34], and outlier detection [9], [24], [35], yielding commendable results even with limited data. Furthermore, FST is applied to rough sets to construct a novel Granular Computing (GrC) method, i.e., Fuzzy Rough Sets (FRSs) [36]. This method approaches the data from a multi-granularity perspective and also considers the uncertainty information. Nowadays, it has been applied to many fields and many desirable methods have been developed based on it [34], [37]. However, the application of FST often involves calculating fuzzy relations between sample pairs, resulting in computationally expensive models. Moreover, the finest grained processing mode in the

model makes it unable to effectively use the multi-granularity information in the data and is easily affected by noisy samples.

Based on the above discussion, we propose a novel density-based outlier detection method called local Granular-Ball Density-based Outlier (GBDO) detection. In GBDO, we use FST to efficiently process the uncertainty information in the data and use multi-granularity GBs as the basic processing unit to further improve the efficiency and robustness of the method. The framework diagram of GBDO is illustrated in Fig. 1, which includes GB representation, local density measure, and outlier detection. First, the normalized unlabeled data are input into an Information System (IS), where GB generation is performed to create multi-granularity GBs as the basic units for subsequent calculations. Then, we calculate the fuzzy relations between GBs. Based on these fuzzy relations, several critical measures in GBDO are calculated including  $k$ -similarity and  $k$ -similarity GB neighborhood. The local GB outlier factor of each sample is calculated by the relative density between the GB belonging to the sample and the  $k$ -similarity GB neighbors around that GB. Finally, the outliers are identified by threshold judgment.

The main contributions of this study are summarized in the following.

- 1) The existing density-based outlier detection methods process data at the individual sample level, requiring  $O(n^2)$  pairwise distance calculations and making them computationally expensive and sensitive to noise. We use GBs as the basic units instead of individual samples for subsequent calculations, significantly improving the performance of outlier detection.
- 2) We propose a multi-granularity fuzzy density estimation method that integrates GBC and FST through  $k$ -similarity GB neighborhoods, enabling simultaneous handling of multi-granularity information and data uncertainty.
- 3) We propose a novel density-based method for unsupervised outlier detection, named GBDO. By leveraging

FST and GBC, GBDO exploits uncertainty and multi-granularity information to measure local density more efficiently, thereby constructing highly discriminative outlier factors.

- 4) Compared with 16 outlier detection methods across 24 datasets, GBDO achieves superior performance, while maintaining a competitive computational efficiency.

The rest of this study is organized as follows. Section II reviews related works on density-based outlier detection and GBC. Section III introduces fundamental knowledge of FST and GBC, laying the foundation for subsequent method descriptions. Section IV proposes an effective unsupervised outlier detection method. Section V presents the corresponding algorithm for the proposed method. Section VI validates the effectiveness of our method through extensive experiments. Section VII concludes the study.

## II. RELATED WORKS

In this section, several density-based outlier detection methods and GBC are reviewed. The related works of GBC include GB generation and the applications of GBC.

### A. Density-Based Outlier Detection

The density-based concept is quite popular for outlier detection. LOF [4] is the earliest and most well-known method, which is inspired by the density-based clustering method. LOF defines the local reachability density of a sample by its  $k$ -distance. The  $k$ -distance neighborhood of a sample and the reachability distance between the samples are defined by  $k$ -distance. Based on the neighborhood of a sample, the local reachability density of the sample is defined. Finally, the relative density between each sample and its surrounding neighborhood is used to calculate the outlier factor. LOF measures the outlier degree of a sample through its outlier factor.

LOF's excellent ability to detect outliers has attracted many scholars to study it further. LOF assumes that the samples in the dataset are distributed in a spherical pattern. This assumption is difficult to hold in high-dimensional or complex data, rendering LOF ineffective in these data. To address this, the Connectivity-based Outlier Factor (COF) assumes a linear distribution of samples in the data space. Integrating both the density and isolation characteristics of samples enhances the performance of LOF [38]. LOF requires searching the neighborhoods three times, making the method not efficient enough. The Relative Density Factor (RDF) uses a vertical data model P-trees and calculates RDF for each sample [39]. Kriegel et al. [7] argued that the scores calculated by the existing methods are not uniform and are often difficult to interpret. They proposed a new local density-based outlier detection method and scored the samples using statistical probability. Hu et al. [40] proposed a new kernel function to estimate the local density of the samples to improve the detection accuracy and combine it with a weighted domain density to improve the robustness. Wahid and Annavarapu [41] introduced the natural neighbor concept and estimated the density of the region where the sample is located by a weighted kernel density estimation method. In addition,  $k$ NN and reverse nearest neighbors are introduced to extend the

application scenario of the method. Yuan et al. [9] extended the idea of weighted density to FRS. The method considers the weights of different attributes and then combines the weight with fuzzy-rough density with that weight to calculate the outlier factor. A novel density-based outlier detection method is proposed by fusing the fuzzy neighborhood information in the data [6].

The density-based methods reviewed above can effectively detect outliers in most cases. Nevertheless, these methods require calculating the distance metrics between samples one by one, which are highly susceptible to noise and not efficient enough. In addition, the different information at multiple levels of granularity and uncertainty in the data deserve further consideration. Gao et al. [5] enhanced local outlier detection using relative fuzzy granule density with multiscale GBs and FRS. While we all focus on GB density-based outlier detection, their method only relies on simple neighborhood information and directly calculates the average fuzzy similarity between GBs to estimate the local density instead of reflecting the density through important neighbors, which inaccurately captures the neighborhood information. In contrast, our method uses GB neighborhoods to calculate the multi-granularity local density of each GB, capturing nuanced density variations and achieving superior experimental performance. Therefore, the application of GB in density-based outlier detection deserves further study.

### B. Granular-Ball Computing

Efficient GB generation methods are a prerequisite for realizing efficient methods based on GBC. Early GB generation is carried out using  $k$ -means clustering [16]. Initially, the whole dataset is regarded as a single GB, and then a  $k$ -means clustering is used to split each GB until the qualities of all the GBs satisfy a given criterion. To enhance the speed and minimize unnecessary computing in GB generation, Xia et al. [18] introduced an efficient GB generation method using the  $k$ -division algorithm. This method not only quickens the GB generation process but also maintains classification accuracy. Furthermore, to eliminate the influence of manually set thresholds on the GB generation, a novel and adaptive GB generation method is proposed, which significantly improves the efficiency and achieves similar classification results as the previous ones. Previous strategies for GB generation relied on  $k$ -means or  $k$ -division, which leads to the randomness of the GB generation method. To address this, Xie et al. [20] proposed that when splitting each GB, it is only necessary to calculate the distance from the data-driven center to the undivided samples, without randomly selecting the center and calculating its distance from all the samples. Xia et al. [27] proposed a novel GB generation method to optimize the distribution of GBs. They initially regarded the entire dataset as a single GB and then split it into  $\sqrt{n}$  GBs. For each GB, they calculated the center consistency as the splitting criterion. If this value was low, further splitting was required. The process continued until all the GBs met the requirements, completing the generation. These efficient GB generation methods provide novel perspectives for the theoretical development and application expansion of GBC.

In terms of the application of GBC, considering that GBC is also a GrC method, it is often combined with a variety of

other GrC methods. GB Neighborhood Rough Sets (GBNRSs) [17], a novel Neighborhood Rough Sets (NRSs) method, combines GBC and NRS to adaptively generate different neighborhoods for each sample. Xia et al. [17] proposed a fast and adaptive attribute reduction method in classification based on GBNRS. Inspired by GBNRS, Zhang et al. [42] proposed a new GB Rough Sets (GBRSs) model and introduced incremental learning to avoid recomputing GBs when encountering new samples. Qian et al. [43] proposed a novel GBC-based FRS (GBFRS) for label distribution feature selection. GBFRS explored the fuzzy relations between GBs and samples with characteristics that mimic large-scale priority in human thinking. Based on the structure and information of the GBs, Bai et al. [22] proposed an adaptive outlier detection method. The method detects outliers by the number of samples in a GB, the overlap between GBs, and the change in the neighborhood relationship of each GB.

The multi-granularity characteristic of GBs makes GBC-based methods more efficient and robust in processing data, which is also very beneficial for outlier detection. Su et al. [23] measured the outlier degree of samples using GB fuzzy approximation accuracy. Cheng et al. [44] improved mean-shift-based outlier detection with GBs to enhance detection performance. Wang et al. [45] enhanced random-walk-based outlier detection using GBs to improve performance. Su et al. [24] detected outliers by fusing multi-granularity fuzzy information. Although these GBC-based outlier detection methods have achieved desirable performance, they overlook critical and rich local density information provided by GBs in the data, limiting their ability to effectively detect local outliers.

In this study, we focus on the limitations of the existing density-based outlier detection methods and improve the performance of density-based methods by incorporating the advantages of GBC and FST to construct a more efficient method.

### III. PRELIMINARIES

In this section, we review the fundamental knowledge related to FST and GBC.

#### A. Fuzzy Set Theory

In FST, data are usually imported into a data table for processing, which is called an IS. An IS can be denoted as a two-tuple  $IS = \langle U, A \rangle$ , where  $U = \{t_1, t_2, \dots, t_n\}$  denotes the set of samples, and  $A$  is the attribute set of the samples. For any  $a \in A$  and  $t \in U$ ,  $a(t)$  denotes the attribute value of the sample  $t$  under the attribute  $a$ . In this study, we discuss unsupervised outlier detection methods, so the attribute set  $A$  contains only the condition attribute  $C$ , so the IS can also be denoted as  $IS = \langle U, C \rangle$ .

In an IS,  $R$  is said to be a fuzzy set on  $U$  if  $R$  is a mapping from  $U$  to  $[0, 1]$ , i.e.  $R : U \rightarrow [0, 1]$ . Unlike the classical concept of set, a fuzzy set is defined by mapping, and the objects in the fuzzy set no longer belong to the object completely, but belong to the set to a certain degree. This fuzzy relation-of belonging is the key to the ability of fuzzy sets to characterize the uncertainty information in data.

For any  $t \in U$ , the degree to which it belongs to the fuzzy set  $R$  is called the membership degree, denoted as  $R(t)$ . For

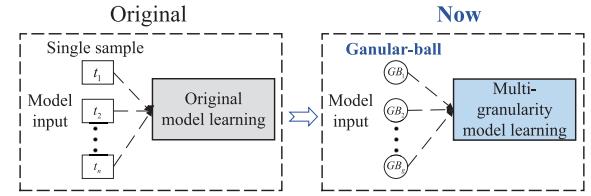


Fig. 2. Comparison between the original model learning and multi-granularity model learning.

all the classical sets on the universe  $U$  denoted as  $P(U)$ , all the fuzzy sets are denoted as  $F(U)$ . Since classical sets can be viewed as special cases where the membership degree takes the value 0 or 1,  $P(U) \subseteq F(U)$ . The fuzzy set is often denoted as  $R = \{(t, R(t) | t \in U)\}$  or  $R = R(t_1)/t_1 + R(t_2)/t_2 + \dots + R(t_n)/t_n = \sum_i^n R(t_i)/t_i$ .

As with classical sets, fuzzy sets have operations such as inclusion, union, and intersection. For any  $R_1, R_2 \in F(U)$  and  $t \in U$ , the relevant operations are defined as follows.

- 1) Inclusion:  $R_1(t) \leq R_2(t) \Rightarrow R_1 \subseteq R_2$ .
- 2) Intersection:  $(R_1 \cap R_2)(t) = R_1(t) \wedge R_2(t)$ .
- 3) Union:  $(R_1 \cup R_2)(t) = R_1(t) \vee R_2(t)$ .

The fuzzy relation  $R$  on  $U$  is defined as  $R : U \times U \rightarrow [0, 1]$ . The set of all the fuzzy relations on  $U$  is denoted as  $F(U \times U)$ . For any  $(t_i, t_j) \in U \times U$ ,  $R(t_i, t_j)$  indicates the degree to which  $t_i$  has a relation  $R$  with  $t_j$ . A fuzzy relation  $R$  on  $U$  is represented by a fuzzy relation matrix,  $M_R = [r_{ij}]_{n \times n}$ , where  $r_{ij} = R(t_i, t_j)$ , and each row represents a fuzzy set.

Depending on the different conditions satisfied, a fuzzy relation can be classified as a fuzzy equivalence relation and a fuzzy similarity relation. For any  $t_i, t_j, t_k \in U$ , if a fuzzy relation  $R$  satisfies  $R(t_i, t_i) = 1$ ;  $R(t_i, t_j) = R(t_j, t_i)$ ;  $R(t_i, t_k) \geq \min\{R(t_i, t_j), R(t_j, t_k)\}$ . Then  $R$  is called a fuzzy equivalence relation on  $U$ . If only the first two conditions are satisfied, then  $R$  is called a fuzzy similarity relation on  $U$ .

#### B. Granular-Ball Computing

Benefiting from human thinking's multi-granularity and coarse-grained cognitive abilities, we can quickly mine the useful information from our surroundings [46]. The process of GBC is consistent with the cognitive mechanism of human thinking, which can mine the knowledge in the data from multi-granularity levels [16]. As shown in Fig. 2, in the model constructed based on GBC, multi-granularity GBs are used to replace the single granularity and the finest grained samples of the original model. Original model learning has also been transformed into multi-granularity model learning.

*Definition 1:* A GB is defined as  $GB_j = \{t_i^j | i = 1, 2, \dots, g\}$ . Every GB has two important features, its center  $c$  and radius  $r$ , and they are defined as

$$c = \frac{1}{g} \sum_{i=1}^g t_i^j, r = \frac{1}{g} \sum_{i=1}^g \Delta(t_i^j, c) \quad (1)$$

where  $t_i^j$  denotes a sample in  $GB_j$ ;  $g$  denotes the number of samples in GB; and  $\Delta(\cdot)$  denotes the distance function.

The above definition gives the calculation method of the center and radius of GB under multi-dimensional data, where

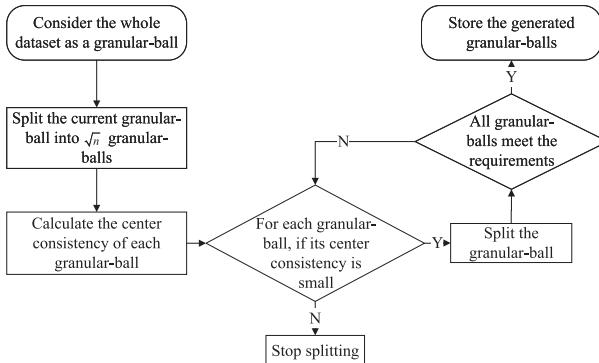


Fig. 3. Process of GB generation.

a variety of radii are obtained by taking different  $\Delta(\cdot)$ . In this study, Euclidean distance is used as the distance function.

GB generation is a critical step in GBC. Fig. 3 demonstrates the process of GB generation used in this study; at first, the whole dataset is considered as a GB. Subsequently, the GB is split into  $\sqrt{n}$  GBs using  $k$ -means clustering, and the center consistency of each GB is calculated as the evaluation criterion for GB splitting. We evaluate the center consistency of each GB. If it is low, we continue to split the GB; otherwise, we stop the splitting. If the quality of all the GBs meets the requirements, the splitting process terminates. Compared with previous generation methods, this GB generation method provides a more efficient solution for the application of GBs in unsupervised tasks, enabling the creation of higher quality GBs and thereby enhancing their performance across various scenarios.

#### IV. METHODOLOGY

In this section, we elaborate on our proposed method starting from fuzzy relations.

**Definition 2:** In  $IS = \langle U, C \rangle$ , for any  $B \subseteq C$  is a subset of condition attributes, the fuzzy relation  $R_B$  between  $t_i$  and  $t_j$  induced by  $B$  is calculated as

$$R_B(t_i, t_j) = 1 - \frac{1}{|B|} \sqrt{\sum_{a \in B} |a(t_i) - a(t_j)|^2} \quad (2)$$

where  $a(t_i)$  and  $a(t_j)$  are the attribute values  $a$  of the samples  $t_i$  and  $t_j$ , respectively, which are in the range  $[0, 1]$  because of the normalization. Obviously, the inequality  $0 \leq R_B(t_i, t_j) \leq 1$  holds. The fuzzy relation  $R_B$  is reflexive and symmetric, i.e., it is a fuzzy similarity relation.

The above definition gives an adaptive method of calculating fuzzy similarity relation, which is defined by the values of the attributes between the samples and the domain of values obtained to ensure that they are between  $[0, 1]$ .

**Definition 3:** In  $IS = \langle U, C \rangle$ , the GBs generated on the whole condition attributes  $C$  are denoted as  $GB_C = \{GB_1^C, GB_2^C, \dots, GB_f^C\}$ . For any  $GB_i^C, GB_j^C \in GB_C$ , the fuzzy similarity relation between  $GB_i^C$  and  $GB_j^C$  induced by  $C$  is defined as

$$R_C(GB_i^C, GB_j^C) = R_C(c_i, c_j) \quad (3)$$

where  $c_i$  and  $c_j$  denote the corresponding centers of the GBs.

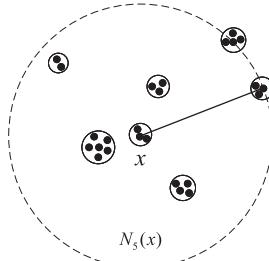


Fig. 4. Illustrative diagram of  $k$ -similarity GB neighborhood. The GB is represented by a solid circle with some black dots inside to represent samples. The closer the GB is to  $x$ , the greater the similarity between them.

In the above definition, the centers of the GBs are used to calculate the fuzzy similarity relations between the GBs, and this substitution avoids computing fuzzy relations between multiple samples and improves the efficiency of the method. For ease of representation, and without confusion,  $GB_C$  is simplified to  $\mathcal{GB}$ .

Next, we define the  $k$ -similarity of GB through the fuzzy similarity relation defined above, which is the basis for constructing critical measures for outlier detection and GBDO.

**Definition 4:** The  $k$ -similarity of GB  $x$  is denoted as  $s_k(x)$ , where  $k$  is a positive integer.  $s_k(x)$  is defined by the fuzzy similarity relation  $R_C(x, y)$  between  $x$  and  $y \in \mathcal{GB}$ . Specifically, the following two points are satisfied:

- 1) for at least  $k$  GBs  $\hat{y} \in \mathcal{GB} - \{x\}$  it holds that  $R_C(x, \hat{y}) \geq R_C(x, y)$ ;
- 2) for at most  $k-1$  GBs  $\hat{y} \in \mathcal{GB} - \{x\}$  it holds that  $R_C(x, \hat{y}) > R_C(x, y)$ .

The  $k$ -similarity of GB defined above can be considered as a neighborhood radius, which can be used to determine the  $k$ -similarity GB neighborhood of GB as follows.

**Definition 5:** The  $k$ -similarity GB neighborhood of GB  $x$  is defined as

$$N_k(x) = \{y \in \mathcal{GB} \mid R_C(x, y) \geq s_k(x)\}. \quad (4)$$

From the above definition, it follows that the  $k$ -similarity GB neighborhood of  $x$  is the set of GBs (including  $x$ ) whose fuzzy similarity degree with  $x$  is greater than or equal to  $s_k(x)$ . These GBs  $y$  are called the  $k$ -similarity GB neighbors of GB  $x$ . Neighborhoods are discussed here in terms of GBs rather than samples. Note that the positive integer  $k$  should not exceed the number of all the GBs generated on the dataset. For any  $x \in \mathcal{GB}$ , the cardinality of  $N_k(x)$  may be greater than  $k+1$ , since there may be more than one GB  $\hat{y} \in \mathcal{GB} - \{x\}$  such that  $R_C(x, \hat{y}) = R_C(x, y)$ .

As shown in Fig. 4, we give an illustrative diagram of  $k$ -similarity GB neighborhood with  $k = 5$ . We abstract the GBs into solid circles with some black sample points, and the distance represents the fuzzy similarity between the GBs. The greater the fuzzy similarity degrees between GBs, the closer the distance between the corresponding two balls in the diagram. For GB  $x$  in this diagram, its  $k$ -similarity GB neighborhood consists of seven GBs (including  $x$ ) inside and on the dashed line, where the number of  $N_5(x)$  is 7 instead of 6. Because the number of GBs exactly on the dotted line may be greater than or equal to 1.

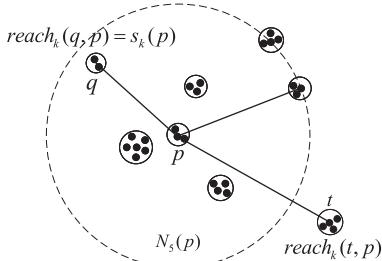


Fig. 5. Illustrative diagram of the reachability similarity.

Following the reachability distance defined in LOF [4], we define the reachability similarity between GBs by the  $k$ -similarity and the fuzzy similarity of GBs as follows.

**Definition 6:** The reachability similarity of GB  $x$  with respect to GB  $y$  is defined as

$$\text{reach}_k(x, y) = \min\{s_k(y), R_C(x, y)\}. \quad (5)$$

Fig. 5 demonstrates the reachability similarity of GB  $q$  with respect to GB  $p$ , where  $\text{reach}_k(q, p) = \min\{s_k(p), R_C(q, p)\} = s_k(p)$ . The closer the distance between the GBs in the diagram, the higher the similarity between them, and vice versa. Therefore, we can get  $s_k(p) < R_C(q, p)$ , so  $\text{reach}_k(q, p) = s_k(p)$ . Similar to LOF [4], the reason why we introduce reachability similarity is to reduce the statistical fluctuation of local fuzzy similarity. If the actual fuzzy similarity  $R_C(q, p)$  between GBs is used directly, when the similarity between  $q$  and  $p$  is too large or too small, the density defined later will fluctuate greatly and lead to unstable results.

With the above definition, the fuzzy similarity is used to decide the  $k$ -similarity and  $k$ -similarity GB neighborhood of GB, which is the key to characterizing the GB neighborhood information of GB. Referring to LOF [4], we define the local reachability similarity density of GB  $x$  based on  $N_k(x)$ .

**Definition 7:** The local reachability similarity density of GB  $x$  is defined as

$$\text{lrsd}_k(x) = \frac{\sum_{y \in N_k(x)} \text{reach}_k(x, y)}{\Lambda(x)} \quad (6)$$

where  $\Lambda(x) = \sum_{y \in N_k(x)} R_C(x, y)$  denotes the sum of the fuzzy similarity of  $x$  with all the GBs in  $N_k(x)$  induced by the whole condition attributes  $C$ .

We define a new local density measure above, which is calculated by the fuzzy similarity of  $x$  to  $y$  in  $N_k(x)$  and the sum of the reachability similarity of each GB  $y$  to  $x$ . For any  $y \in N_k(x)$ , if  $y$  with  $x$  has a large  $\text{reach}_k(x, y)$ , the result for  $\text{lrsd}_k(x)$  will also be large, indicating that all  $y$  and  $x$  are compact. Different from the existing density-based methods, we leverage the reachability similarity between GBs to measure the local density of GBs. This density measure corrects the problem of too large or too small density in traditional density based on Euclidean distance. Moreover, we use multi-granularity GBs as the calculation unit, which can help us analyze data from multiple granularity levels. Based on the local density we defined, we define the outlier factor as follows.

**Definition 8:** The local GB outlier factor of sample  $t$  is defined as

$$OF_k(t) = \frac{\sum_{y \in N_k(T(t))} \text{lrsd}_k(y)}{\Lambda(T(t))} \quad (7)$$

where  $T(\cdot)$  denotes a function that returns the GB to which sample  $t$  belongs. Through the processing of the function  $T(\cdot)$ , the relevant measures of each GB are mapped to the samples within that GB. It is easy to see that the lower  $T(t)$ 's local reachability similarity density is, and the higher the local reachability similarity densities of  $T(t)$ 's  $k$ -similarity GB neighbors are, the higher is the  $OF_k$  value of  $t$ . The outlier factor of a sample is measured by the GB neighborhood of the GB to which the sample belongs, where the neighborhood is the  $k$ -similarity GB neighborhood. The value of the outlier factor reflects the outlier degree of the sample. The larger the outlier factor, the more abnormal the sample. The relative density between the GBs to which the sample belongs and its surrounding GBs enables a better characterization of the local outlier of the sample, rather than just focusing on the global features. The above definition does not rely on the finest grained samples but instead calculates the GB outlier factor for samples based solely on the defined measures of pregenerated GBs. Consequently, compared with traditional methods based on individual samples, our method is more efficient.

**Definition 9:** Given an outlier determination threshold  $\xi$ . For any  $t \in U$ , if  $OF_k(t) > \xi$ , then  $t$  is said to be a local GB density-based outlier in  $U$ .

---

#### Algorithm 1 GBDO

---

```

Input: IS ==<  $U, C$  >,  $k$ 
Output:  $OF_k$ 
1 Initializing the set of outlier factors:  $OF_k \leftarrow \emptyset$ 
2 Generate GBs  $GB_C$  on  $C$  by the GB splitting strategy
   in [27];
3 for each GB  $x \in GB_C$  do
4   for each GB  $y \in GB_C$  do
5     Calculate the fuzzy similarity relationship
        $R_C(x, y)$  between GB  $x$  and  $y$  by Eq. (3);
6   end
7   Calculate the  $k$ -similarity  $s_k(x)$  and the  $k$ -similarity
      GB neighborhood  $N_k(x)$  of GB  $x$  by Definition 4
      and Eq. (4);
8 end
9 for each GB  $x \in GB_C$  do
10   for each GB  $y \in GB_C$  do
11     Calculate the reachability similarity
         $\text{reach}_k(x, y)$  of GB  $x$  with respect to  $y$ 
        by Eq. (5);
12   end
13   Calculate the local reachability similarity density
       $\text{lrsd}_k(x)$  of GB  $x$  by Eq. (6);
14 end
15 for each sample  $t \in U$  do
16   Calculate the local GB outlier factor  $OF_k(t)$  of
      sample  $t$  by Eq. (7);
17 end
18 return  $OF_k$ .

```

---

To determine the appropriate outlier threshold, we sort the scores in ascending order and set the threshold at the 95th percentile, classifying samples with scores above this value as outliers.

## V. ALGORITHM IMPLEMENTATION

The pseudocode of GBDO is shown in Algorithm 1. First, the GBs are generated on the input IS based on the GB splitting strategy in [27]; note that here they are generated over the entire condition attributes and input data have been normalized. Then, the fuzzy relation between two GBs is calculated, which is at the heart of the subsequent steps of GBDO. Afterward,  $N_k$  is generated based on the  $k$ -similarity of the GBs. Finally,  $lrsd_k$  and  $OF_k$  of the GBs are calculated within the  $N_k$  range. The output of the algorithm yields only the outlier factor, which needs to be followed up with a set threshold  $\xi$  to achieve outlier detection. In this algorithm, considering that the hyperparameter  $k$  is less than or equal to the number  $|GB_C|$  of generated GBs, the time consumption of the whole algorithm is mainly on calculating the fuzzy relation between GBs. The time complexity of GB generation is approximated as  $O(|U|^{(3/2)})$  [27]. The time complexity of the whole algorithm is  $O(|U|^{(3/2)} + |GB_C|^2)$ .

## VI. EXPERIMENTS

To validate the effectiveness of the proposed method, we conduct experiments in terms of the Receiver Operating Characteristic (ROC) curves and the Area Under the ROC Curve (AUC) results, geometric mean (g-mean) results, running times, hyperparameter sensitivity analysis, and statistical analysis. We first give the relevant setup of the experiments, after which the analysis is carried out based on the results.

### A. Experimental Setup

We conduct several experiments on several publicly available datasets.<sup>1,2</sup> The basic information of the experimental datasets is shown in Table I.

We compare our method with several outlier detection methods, including Granular-Ball computing-based Random walk for Anomaly Detection (GBRAD) [45], Multi-scale Granular Balls-based Outlier Detection (MGBOD) [5], Detecting Fuzzy Neighborhood Outlier (DFNO) [6], Score-Guided AutoEncoder (SG-AE) [47], Feature Encoding with Autoencoders for Weakly Supervised Anomaly Detection (FEAWAD) [48], Multi-Fuzzy Granules Anomaly Detection (MFGAD) [35], Weighted Fuzzy-Rough Density-based Anomaly (WFRDA) [9], Empirical Cumulative Distribution-based Outlier Detection (ECOD) [49], Directed Density Ratio Changing Rate-based Outlier Detection (DCROD) [11], Weighted Neighborhood Information Network-based Outlier Detection (WNINOD) [50], Rotation-based Outlier Detection (ROD) [51], COPula-based Outlier Detection (COPOD) [52], the VariancE structural score-based outlier detection (VarE) [53], reverse uNreaChability based outlier detection (NC) [54], Isolation Forest-based outlier detection (IForest) [55], and the LOF [4]. We use the optimal parameters provided in the

TABLE I  
EXPERIMENTAL DATASETS

Datasets	# Attributes	# Samples	% Outlier ratios
Breast	9	286	29.72%
Mushroom	22	4573	7.98%
Vote	16	296	9.80%
Zoo	16	101	16.83%
Cardio	21	1831	9.61%
Diab	8	526	4.94%
Ecoli	7	336	2.68%
Glass	9	214	4.21%
Iono	34	249	9.64%
Iris	4	111	9.91%
Vowels	12	1456	3.43%
Wave	21	3443	2.90%
Wbc	9	483	8.07%
Wdbc	32	396	9.85%
Wine	13	129	7.75%
Wpbc	33	198	23.74%
Yeast	8	1141	0.44%
Aba	9	4177	1.89%
Ann	38	798	5.26%
Arr	279	452	14.60%
Autos	25	205	12.20%
Bands	40	328	4.88%
Sick	29	3613	1.99%
Thyroid	28	9172	0.81%

papers of each method for experiments. For GBDO, there is an adjustable parameter  $k$ , which determines  $k$ -similarity, which in turn determines subsequent  $k$ -similarity GB neighborhood, reachability similarity, etc. The experimental results of GBDO on each dataset are the optimal results within a certain tuning range of  $k$ . We determine the tuning range of  $k$  based on how many GBs are generated on each dataset. Specifically, the tuning range is  $k \in [1, \min\{100, |GB_C|\}]$  with a step size of 1.

ROC curves and AUC are commonly used to evaluate the performance of different methods [9], [56]. ROC curves represent the tradeoff between the True-Positive Rate (TPR) and the False-Positive Rate (FPR). The closer the ROC curve of a method is to the upper left corner of the coordinate axis, the better the performance of the method. AUC is the area under the ROC curve. The closer the AUC of a method is to 1, the better the performance of the method. G-mean is the geometric mean, which is calculated by g-mean =  $(TNR \times TAR)^{1/2}$ , where TNR denotes the True Normal Rate and TAR denotes the True Abnormal Rate [57].

### B. Experimental Results

Fig. 6 shows the ROC curves of 17 methods on 24 datasets, where the yellow inverted triangle is the curve of GBDO. It can be seen from the figure that the curves of GBDO in the Breast, Mushroom, Cardio, Ecoli, Iris, Wpbc, Yeast, Ann, and Autos datasets are closer to the upper left corner, indicating that GBDO outperforms other methods on these datasets.

Next, the AUC results of different methods are given below to compare the performance of different methods intuitively. The AUC results of different methods are shown in Table II. Through this table, we can compare the performance of different methods more clearly and intuitively. GBDO can be regarded as an extension and improvement of LOF, so we first compare the experimental results of GBDO and LOF. From the

<sup>1</sup><https://github.com/BELLoney/Outlier-detection>

<sup>2</sup><https://odds.cs.stonybrook.edu/>

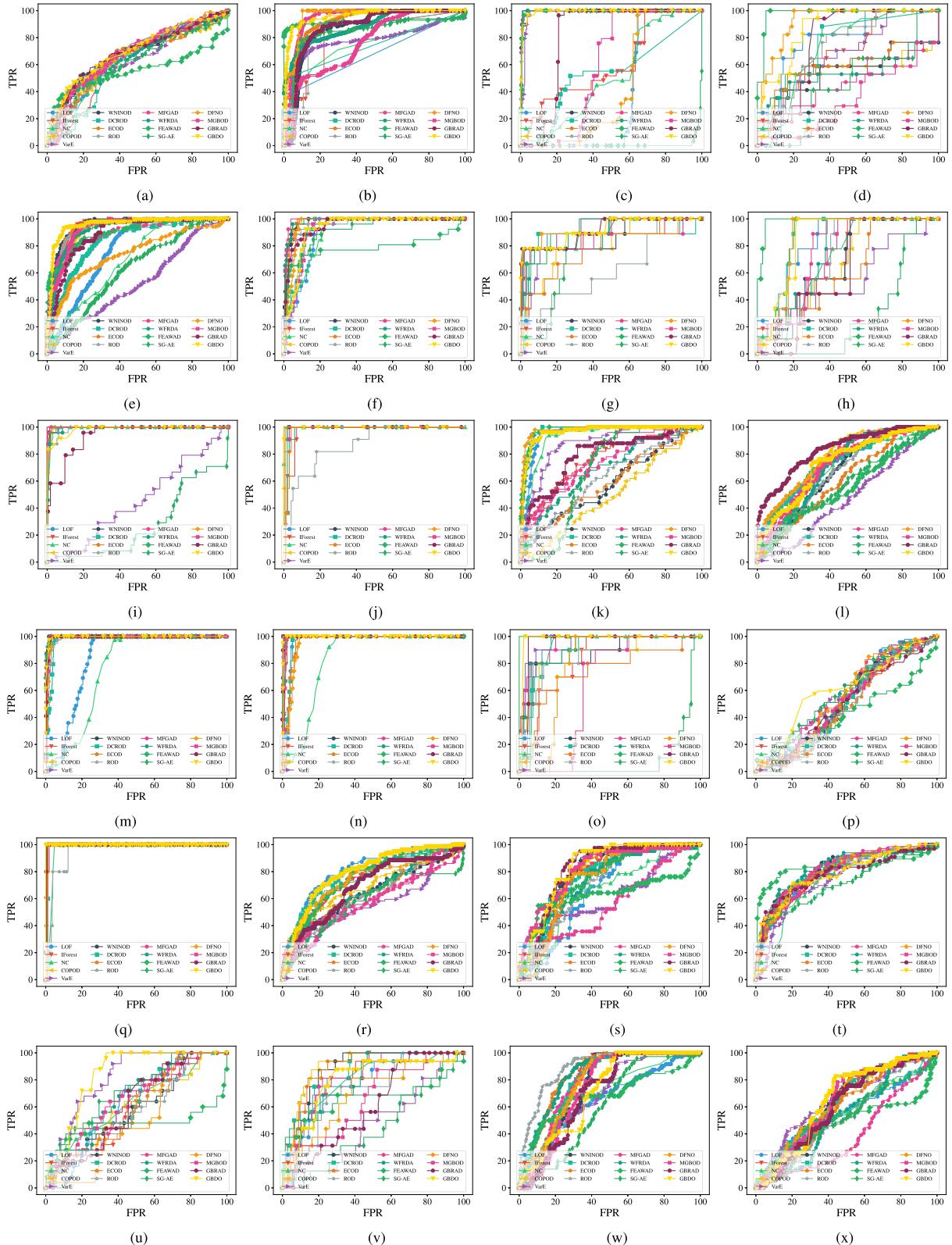


Fig. 6. Experimental comparison results on ROC. (a) Breast. (b) Mushroom. (c) Vote. (d) Zoo. (e) Cardio. (f) Diab. (g) Ecoli. (h) Glass. (i) Iono. (j) Iris. (k) Vowels. (l) Wave. (m) Wbc. (n) Wdbc. (o) Wine. (p) Wpbc. (q) Yeast. (r) Aba. (s) Ann. (t) Arr. (u) Autos. (v) Bands. (w) Sick. (x) Thyroid.

table, we can see that the average AUC results of GBDO on all the datasets are 11% higher than that of LOF, among which the difference in performance on the Vote dataset is the most obvious. This shows that GBDO performs better than LOF and can mine more potential outliers than LOF cannot detect. Next,

we continue to compare GBDO with other methods. GBDO achieves the best results and second-best results on 14 out of 24 datasets and has the best average AUC results on all the datasets. Although GBDO does not achieve the best results on all the datasets, it is optimal in overall performance.

TABLE II  
EXPERIMENTAL RESULTS ON AUC, WHERE THE BEST RESULT FOR EACH DATASET IS HIGHLIGHTED IN BOLD,  
AND THE SECOND-BEST RESULT IS UNDERLINED

Datasets	AUC																
	LOF	IForest	NC	COPOD	VarE	WNINOD	DCROD	ECOD	ROD	MFGAD	WFRDA	FEAWAD	SG-AE	DFNO	MGBOD	GRAD	GBDO
Breast	0.654	0.635	0.642	0.632	0.637	0.671	0.604	0.656	<u>0.692</u>	0.685	0.654	0.556	0.623	0.677	0.664	0.685	<b>0.700</b>
Mushroom	0.682	0.812	0.741	0.902	0.761	0.867	0.728	0.900	0.821	0.942	0.886	0.920	0.835	<u>0.955</u>	0.757	0.875	<b>0.959</b>
Vote	0.378	0.603	0.513	<u>0.995</u>	0.981	<u>0.995</u>	0.537	<u>0.995</u>	<b>0.997</b>	0.995	0.992	0.009	0.429	0.594	0.808	0.991	
Zoo	0.769	0.690	0.723	0.550	0.683	0.539	0.752	0.582	0.735	0.466	0.526	<b>0.979</b>	0.454	<u>0.921</u>	0.342	0.777	0.875
Cardio	0.773	0.941	0.703	0.922	0.526	<u>0.949</u>	0.834	0.935	0.932	0.897	0.922	0.898	0.628	0.741	0.929	0.889	<b>0.956</b>
Diab	0.905	0.976	0.902	0.986	0.965	0.981	0.935	0.979	0.954	<b>0.989</b>	0.984	0.784	0.920	0.962	0.924	0.956	0.929
Ecoli	0.880	0.867	0.881	0.809	0.899	0.874	0.875	0.781	0.613	0.849	0.875	0.903	0.799	0.890	0.894	<u>0.916</u>	<b>0.919</b>
Glass	0.832	0.682	0.704	0.645	0.538	0.674	<u>0.863</u>	0.621	0.738	0.746	0.746	<b>0.985</b>	0.284	0.856	0.724	0.618	0.853
Iono	0.991	<u>0.999</u>	0.993	0.995	0.456	0.996	<b>1.000</b>	0.994	0.987	<u>0.999</u>	0.993	<b>1.000</b>	0.275	<b>1.000</b>	<b>1.000</b>	0.935	0.985
Iris	0.998	0.971	0.996	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.983	0.977	0.866	<u>0.997</u>	<b>1.000</b>	<b>1.000</b>	0.990	<b>1.000</b>	<b>1.000</b>	0.990	<b>1.000</b>
Vowels	0.951	0.761	0.927	0.496	0.878	0.581	<u>0.978</u>	0.593	0.626	0.767	0.686	0.973	0.674	<b>0.980</b>	0.731	0.793	0.962
Wave	0.741	0.708	0.696	0.734	0.473	0.671	0.743	0.608	0.670	0.736	0.704	0.557	0.574	<u>0.757</u>	0.719	<b>0.835</b>	0.734
Wbc	0.837	0.996	0.756	0.995	<b>0.997</b>	<b>0.997</b>	0.976	0.995	0.985	0.994	<b>0.997</b>	0.994	0.989	0.991	0.995	0.996	
Wdbc	0.964	0.987	0.834	0.996	0.997	0.996	0.968	0.959	0.995	0.996	<b>0.999</b>	0.992	0.998	0.954	0.988	0.998	0.997
Wine	0.933	0.824	0.913	0.867	0.941	0.873	0.885	0.733	0.850	0.945	0.915	<b>1.000</b>	0.089	0.780	0.631	0.952	<u>0.983</u>
Wpbc	0.526	0.504	0.510	0.523	0.552	0.506	0.513	0.481	0.487	0.564	0.528	0.437	<u>0.570</u>	0.552	0.521	0.509	<b>0.586</b>
Yeast	0.992	0.997	0.970	0.997	<b>1.000</b>	<u>0.998</u>	0.990	0.995	0.949	0.992	<u>0.998</u>	<b>1.000</b>	0.994	0.989	0.991	0.997	<b>1.000</b>
Aba	<b>0.811</b>	0.753	0.729	0.696	0.548	0.641	0.780	0.653	0.629	0.565	0.661	0.744	0.576	0.726	0.603	0.687	<b>0.794</b>
Ann	0.665	0.789	0.653	0.797	0.619	0.791	0.755	0.787	0.712	0.554	0.739	0.625	0.761	0.742	0.818	<u>0.829</u>	<b>0.834</b>
Arr	0.800	0.788	0.734	0.805	0.739	0.815	0.796	0.807	0.801	0.790	<u>0.826</u>	<b>0.832</b>	0.708	0.800	0.822	0.780	0.783
Autos	0.628	0.596	0.625	0.600	<u>0.819</u>	0.588	0.619	0.583	0.562	0.605	0.672	0.467	0.714	0.505	0.668	0.610	<b>0.839</b>
Bands	0.789	0.847	0.793	<b>0.898</b>	0.458	0.859	0.864	<u>0.883</u>	0.719	0.816	0.779	0.666	0.432	0.693	0.649	0.606	0.794
Sick	0.653	0.784	0.678	0.780	0.691	0.754	0.788	0.843	<b>0.895</b>	0.786	<u>0.837</u>	0.683	0.601	0.787	0.737	0.704	0.712
Thyroid	0.549	0.614	0.564	0.611	<b>0.650</b>	0.640	0.646	0.581	0.533	0.388	0.531	0.496	0.510	0.645	0.638	0.605	<u>0.647</u>
Average	0.779	0.797	0.758	0.801	0.742	0.802	0.809	0.788	0.781	0.794	0.811	<u>0.812</u>	0.626	0.805	0.764	0.807	<b>0.868</b>

TABLE III  
EXPERIMENTAL RESULTS ON G-MEAN, WHERE THE BEST RESULT FOR EACH DATASET IS HIGHLIGHTED IN BOLD,  
AND THE SECOND-BEST RESULT IS UNDERLINED

Datasets	G-mean																
	LOF	IForest	NC	COPOD	VarE	WNINOD	DCROD	ECOD	ROD	MFGAD	WFRDA	FEAWAD	SG-AE	DFNO	MGBOD	GRAD	GBDO
Breast	0.658	0.636	0.659	0.643	0.616	0.663	0.636	0.659	0.647	0.644	0.620	0.604	0.602	0.646	0.654	<u>0.666</u>	<b>0.678</b>
Mushroom	0.774	0.825	0.706	<u>0.834</u>	0.778	0.825	0.705	0.855	0.852	0.910	0.816	<u>0.920</u>	0.840	<b>0.947</b>	0.672	0.833	0.913
Vote	0.593	0.566	0.545	<b>0.994</b>	0.973	<b>0.994</b>	0.613	<b>0.994</b>	<u>0.985</u>	<b>0.994</b>	0.977	0.064	0.586	0.703	0.874	0.970	
Zoo	0.813	0.668	0.728	0.664	0.654	0.621	0.774	0.664	0.714	0.529	0.622	<b>0.976</b>	0.555	<u>0.893</u>	0.463	0.806	0.845
Cardio	0.732	<b>0.895</b>	0.640	0.859	0.508	0.884	0.778	<u>0.895</u>	0.878	0.846	0.861	0.826	0.610	0.696	0.874	<b>0.921</b>	
Diab	0.901	0.961	0.879	0.963	0.920	0.954	0.911	0.956	0.914	<b>0.984</b>	0.958	0.812	0.856	0.951	0.910	0.889	0.907
Ecoli	0.879	0.874	0.864	0.791	0.844	0.878	<b>0.899</b>	0.728	0.607	0.859	0.848	0.861	0.822	0.886	0.872	0.881	
Glass	0.835	0.709	0.684	0.663	0.549	0.698	0.900	0.613	0.759	0.808	0.787	<b>0.978</b>	0.411	0.897	0.746	0.698	0.886
Iono	0.975	<u>0.993</u>	0.984	0.984	0.486	0.972	<b>1.000</b>	0.975	0.948	<u>0.993</u>	0.970	<b>1.000</b>	0.401	<b>1.000</b>	<b>1.000</b>	0.876	0.943
Iris	0.990	0.964	0.990	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.975	0.980	0.819	0.985	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>0.995</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
Vowels	0.927	0.706	0.891	0.502	0.830	0.567	<b>0.950</b>	0.568	0.605	0.710	0.651	<u>0.943</u>	0.696	<b>0.950</b>	0.680	0.768	0.935
Wave	0.688	0.676	0.664	0.715	0.503	0.632	0.702	0.584	0.633	0.695	0.669	0.532	0.557	0.692	0.679	<b>0.752</b>	<u>0.718</u>
Wbc	0.856	0.986	0.777	0.986	0.989	<b>0.991</b>	0.974	0.986	0.957	0.988	0.989	<u>0.990</u>	0.985	0.983	0.985	0.989	0.983
Wdbc	0.972	0.987	0.837	0.994	0.997	0.997	0.959	0.959	0.986	0.993	0.997	0.989	0.994	0.951	0.989	<u>0.997</u>	<b>0.999</b>
Wine	0.917	0.794	0.898	0.844	0.904	0.864	0.836	0.744	0.832	0.903	0.868	<b>1.000</b>	0.213	0.830	0.722	0.974	0.987
Wpbc	0.566	0.531	0.547	0.548	0.568	0.537	0.552	0.509	0.529	<u>0.579</u>	0.571	0.480	0.564	0.575	0.535	0.554	<b>0.596</b>
Yeast	0.996	0.998	0.980	<u>0.999</u>	<b>1.000</b>	0.999	0.993	0.995	0.936	0.993	<u>0.999</u>	<b>1.000</b>	0.993	0.993	0.996	0.996	<b>1.000</b>
Aba	<b>0.759</b>	0.696	0.715	0.652	0.621	0.734	0.631	0.602	0.564	0.620	0.699	0.592	0.700	0.599	0.659	<u>0.756</u>	
Ann	0.654	0.758	0.693	0.755	0.613	0.774	0.756	0.759	0.675	0.532	0.719	0.700	0.698	0.751	0.765	<u>0.799</u>	<b>0.817</b>
Arr	0.743	0.740	0.701	0.740	0.700	0.747	0.745	0.740	0.737	0.735	0.765	<b>0.827</b>	0.657	0.742	0.744	0.731	<u>0.771</u>
Autos	0.659	0.600	0.618	0.602	<u>0.764</u>	0.560	0.615	0.566	0.541	0.613	0.651	0.604	0.672	0.538	0.645	0.626	<b>0.816</b>
Bands	0.758	0.771	0.765	<b>0.867</b>	0.528	0.839	0.809	<u>0.839</u>	0.721	0.807	0.740	0.691	0.488	0.709	0.646	0.585	0.787
Sick	0.628	0.771	0.651	0.767	0.652	0.763	0.783	0.789	<b>0.823</b>	0.749	0.788	0.686	0.603	0.757	0.738	0.694	0.738
Thyroid	0.560	0.624	0.554	0.611	0.627	0.634	0.628	0.596	0.528	0.445	0.579	0.578	0.537	0.626	<u>0.656</u>	0.617	<b>0.680</b>
Average	0.785	0.780	0.749	0.791	0.732	0.792	0.801	0.774	0.760	0.786	0.795	<u>0.820</u>	0.642	0.804	0.761	0.795	<b>0.855</b>

In addition to the AUC results, we also show the g-mean results of 17 methods on 24 datasets. As shown in Table III, first, GBDO consistently achieves better results than LOF, 9% higher than LOF in average results. Then, GBDO achieves the best results and second-best results on 13 datasets out of 24 datasets, and also the best results in average results, which further demonstrates the effectiveness of GBDO.

#### C. Statistical Analysis

&lt;p

TABLE IV

EXPERIMENTAL RESULTS ON RUNNING TIME, WHERE THE BEST RESULT FOR EACH DATASET IS HIGHLIGHTED IN BOLD,  
AND THE SECOND-BEST RESULT IS UNDERLINED

Datasets	Running time																
	LOF	IForest	NC	COPOD	VarE	WNINOD	DCROD	ECOD	ROD	MFGAD	WFRDA	FEAWAD	SG-AE	DFNO	MGBOD	GBRAD	GBDO
Breast	0.0042	0.0699	0.0141	<b>0.0032</b>	0.0330	0.0175	0.0220	0.0060	0.9935	0.0102	0.0106	22.3071	18.5129	0.0312	0.1270	0.2320	0.3940
Mushroom	0.0697	0.1969	0.9220	<b>0.0075</b>	21.0130	403.6079	1.5749	<u>0.0088</u>	13.9377	6.5090	21.9614	22.8913	94.5396	6.5641	7.8459	4.5899	4.9042
Vote	0.0033	0.0880	0.0860	<u>0.0009</u>	0.0140	0.0319	0.0490	<u>0.0015</u>	0.3718	0.0134	0.0194	23.9398	15.7244	0.0266	0.1590	0.2050	0.2250
Zoo	0.0013	0.0750	0.0090	<b>0.0005</b>	<u>0.0010</u>	0.0054	0.0070	<b>0.0005</b>	0.2631	0.0042	0.0023	21.8583	7.5427	0.0030	0.0450	0.0700	0.0930
Cardio	0.0134	0.1320	0.7380	<b>0.0038</b>	3.5310	13.2363	0.3280	0.0047	30.3341	8.1752	2.5837	22.7535	40.4815	0.9979	1.2790	1.3059	1.7800
Diab	0.0034	0.0920	0.1100	<b>0.0009</b>	0.1450	0.1089	0.0690	0.0016	0.3896	0.1487	0.0417	26.0332	30.4635	0.0810	0.2260	0.2880	0.4420
Ecoli	0.0017	0.0440	0.0460	<b>0.0006</b>	0.0580	0.0192	0.0290	0.0008	0.1560	0.0282	0.0117	21.9713	18.4232	0.0333	0.1330	0.1668	0.2894
Glass	0.0012	0.0402	0.0120	<b>0.0006</b>	0.0130	0.0095	0.0260	0.0007	0.3387	0.0148	0.0065	25.5066	12.2178	0.0130	0.0980	0.1668	0.1210
Iono	0.0020	0.0415	0.0350	<b>0.0012</b>	0.0180	0.0412	0.0190	0.0014	27.9228	0.5158	0.0270	28.3248	13.2960	0.0168	0.1280	0.1998	0.1240
Iris	0.0007	0.0394	0.0049	<b>0.0004</b>	0.0050	0.0024	0.0130	0.0009	0.0146	0.0029	0.0008	26.7523	7.1536	0.0040	0.0430	0.0457	0.0965
Vowels	0.0164	0.0500	0.1420	<b>0.0023</b>	1.3130	3.8254	0.2550	<u>0.0032</u>	4.7404	2.1712	0.6412	28.2050	47.6157	0.6350	0.8406	0.8594	1.4406
Wave	0.0259	0.0660	0.6550	<b>0.0099</b>	45.1260	153.5738	0.7310	0.0118	113.6711	39.8359	8.3084	34.4140	46.8567	3.6387	5.5414	2.8121	3.4206
Wbc	0.0026	0.0425	0.0410	<b>0.0007</b>	0.1180	0.0888	0.1720	0.0016	0.2633	0.1240	0.0362	29.7813	25.3664	0.0700	0.1350	0.4782	0.2695
Wdbc	0.0020	0.0449	0.0640	<b>0.0017</b>	0.0840	0.1335	0.0620	0.0019	27.5495	0.9526	0.0696	33.7311	20.9644	0.0470	0.3130	0.2170	0.4190
Wine	0.0008	0.0419	0.0070	<b>0.0005</b>	0.0040	0.0058	0.0180	0.0007	0.5871	0.0223	0.0028	35.8032	9.0646	0.0040	0.0800	0.0685	0.1049
Wpbc	0.0014	0.0411	0.0090	<b>0.0010</b>	0.0060	0.0270	0.0240	0.0013	0.5871	0.2200	0.0169	41.7231	11.8373	0.0110	0.1910	0.1012	0.1920
Yeast	0.0108	0.0473	0.0810	<b>0.0012</b>	0.7650	1.6424	0.1410	0.0013	0.5805	0.5068	0.2814	37.2338	46.7444	0.3847	0.4105	0.6892	1.5012
Abs	0.0397	0.3460	0.3460	<b>0.0041</b>	76.1620	108.6773	0.5480	<u>0.0050</u>	8.5298	7.3366	11.3265	38.8927	75.1148	5.4983	3.6704	4.2838	4.1666
Ann	0.0035	0.1350	0.1350	<b>0.0018</b>	0.3310	1.9494	0.1270	0.0027	12.5723	3.9809	0.3745	42.2949	34.8078	0.1893	0.5837	0.5769	0.6956
Arr	0.0050	0.1860	0.1860	<u>0.0093</u>	0.3190	1.9181	0.2390	0.0117	19782.1753	83.6246	0.7677	87.9736	20.3054	0.0711	1.9800	0.3983	0.4610
Autos	0.0018	0.0406	0.0060	<b>0.0008</b>	0.0410	0.0211	0.0290	0.0015	41.6273	0.1400	0.0946	45.2689	13.3094	0.0120	0.2110	0.1332	0.1943
Bands	0.0017	0.0428	0.0140	<b>0.0013</b>	0.1420	0.0909	0.0240	0.0014	30.3742	0.5505	0.0533	48.4226	21.1358	0.0330	0.3960	0.1658	0.2870
Sick	0.0283	0.3190	0.3190	<b>0.0069</b>	42.2210	249.3891	0.5400	0.0088	43.9078	62.9006	13.8094	47.0279	45.8542	4.0728	4.4116	4.0123	2.5501
Thyroid	0.0580	0.3080	0.3080	<b>0.0241</b>	497.6691	4059.2471	3.6030	0.0247	130.0694	359.5128	201.6389	47.9551	57.7544	27.0691	28.3159	16.2323	8.1929
Average	0.0125	0.2209	0.2943	<b>0.0036</b>	28.7138	208.2362	0.3604	0.0044	844.6649	24.0542	10.9203	35.0444	30.6286	2.0628	2.3818	1.5958	1.3485

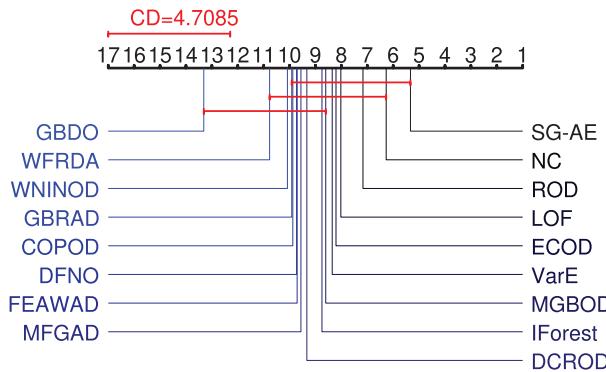


Fig. 7. Nemenyi test figure of 17 methods and 24 datasets on AUC.

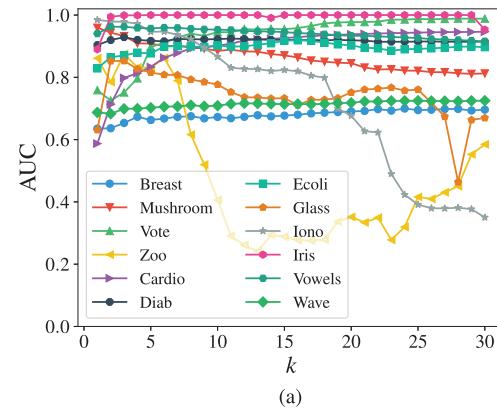
The Nemenyi post-hoc test offers a comprehensive way to assess the relative performance of various evaluated methods called Nemenyi test figure. Specifically, when adopting a significance threshold of  $\alpha = 0.1$ , the Nemenyi test delineates a critical distance (CD), denoted as  $CD_{0.1} = 4.7085$ . The average ordinal values of each method are plotted along a singular axis, supplemented by the visualization of a CD as a horizontal line segment.

The Nemenyi test figure is shown in Fig. 7; the presence of a red horizontal line segment labeled CD, spanning across the ordinal values of certain methods, indicates that there is no significant difference between these methods. From the figure, we can see that GBDO only overlaps with WFRDA, WNINOD, GBRAD, COPOD, DFNO, FEAWAD, MFGAD, DCROD, IForest, and MGBOD in horizontal line segments, indicating that there is no obvious evidence of the statistical difference between them.

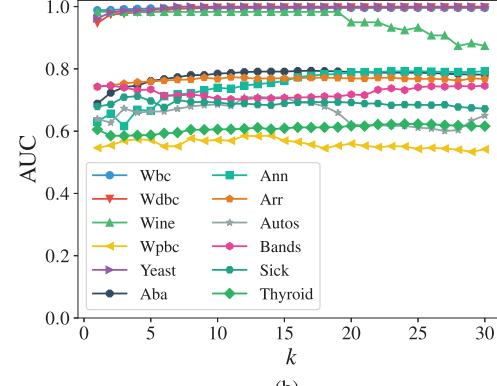
#### D. Hyperparameter Analysis

GBDO is characterized by its simplicity and efficiency, and it contains only one hyperparameter, denoted as  $k$ .

To demonstrate the influence of the variations in  $k$  on the performance of GBDO, Fig. 8 is presented, showcasing the relation between different  $k$  values and the corresponding AUC results across 24 datasets. It should be noted that since the



(a)



(b)

Fig. 8. Variation curves of AUC on  $k$ . (a) AUC variation curves from Breast to Wave. (b) AUC variation curves from Wbc to Thyroid.

tuning range of  $k$  in GBDO is  $k \in [1, \min\{100, |GB_C|\}]$  in the experiment, the range of  $k$  values in different datasets is different. For the convenience of comparison, we set the range of  $k$  to  $[1, 30]$ .

As can be seen in Fig. 8, the performance of GBDO on most datasets improves as  $k$  increases and finally stabilizes. On some datasets with smaller sample sizes, an overly large  $k$  can lead to worse detection results, such as the Iono dataset. There is a different trend change in the Zoo dataset. This

is because all the attributes in the Zoo dataset are nominal attributes, and GBDO cannot effectively process nominal data, which leads to GBDO's inability to extract rich abnormal information in Zoo, which in turn causes a different trend in the AUC curve. In general, GBDO is insensitive to the hyperparameter  $k$ , and the recommended default parameter  $k$  is 5.

## VII. CONCLUSION

In this study, we focus on detecting local outliers and propose a novel density-based outlier detection method, named GBDO. GBDO considers the local GB density of samples and characterizes the outlier degree of a sample by calculating the local reachability similarity density between the  $k$ -similarity GB neighborhood of the GB to which the sample belongs. The introduction of GBC and FST is well-suited to help GBDO efficiently mine potential uncertainty information in the data. The experimental results demonstrate that GBDO's comprehensive performance under 24 datasets outperforms 16 methods. However, like most unsupervised outlier detection methods, selecting the hyperparameter  $k$  in GBDO is challenging. Therefore, in future work, we plan to eliminate the need for hyperparameters or adopt a learning-based method to achieve adaptive outlier detection.

## REFERENCES

- [1] A. Smiti, "A critical overview of outlier detection methods," *Comput. Sci. Rev.*, vol. 38, 2020, Art. no. 100306.
- [2] A. H. Rabie and A. I. Saleh, "Diseases diagnosis based on artificial intelligence and ensemble classification," *Artif. Intell. Med.*, vol. 148, Feb. 2024, Art. no. 102753.
- [3] P. Chatterjee, D. Das, and D. B. Rawat, "Digital twin for credit card fraud detection: Opportunities, challenges, and fraud detection advancements," *Future Gener. Comput. Syst.*, vol. 158, pp. 410–426, Sep. 2024.
- [4] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 93–104.
- [5] C. Gao, X. Tan, J. Zhou, W. Ding, and W. Pedrycz, "Fuzzy granule density-based outlier detection with multi-scale granular balls," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 3, pp. 1182–1197, Mar. 2025.
- [6] Z. Yuan, P. Hu, H. Chen, Y. Chen, and Q. Li, "DFNO: Detecting fuzzy neighborhood outliers," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 1, pp. 200–209, Jan. 2025.
- [7] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "LoOP: Local outlier probabilities," in *Proc. 18th ACM Conf. Inf. Knowl. Manage. (CIKM)*, Nov. 2009, pp. 1649–1652.
- [8] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "LOCI: Fast outlier detection using the local correlation integral," in *Proc. 19th Int. Conf. Data Eng.*, Mar. 2003, pp. 315–326.
- [9] Z. Yuan, B. Chen, J. Liu, H. Chen, D. Peng, and P. Li, "Anomaly detection based on weighted fuzzy-rough density," *Appl. Soft Comput.*, vol. 134, Feb. 2023, Art. no. 109995.
- [10] S. A. Naghavi Nozad, M. Amir Haeri, and G. Folino, "SDCOR: Scalable density-based clustering for local outlier detection in massive-scale datasets," *Knowl.-Based Syst.*, vol. 228, Sep. 2021, Art. no. 107256.
- [11] K. Li et al., "Robust outlier detection based on the changing rate of directed density ratio," *Expert Syst. Appl.*, vol. 207, Nov. 2022, Art. no. 117988.
- [12] P. Wang, C. Da, and C. Yao, "Multi-granularity prediction for scene text recognition," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 339–355.
- [13] J. Xie, X. Xiang, S. Xia, L. Jiang, G. Wang, and X. Gao, "MGNR: A multi-granularity neighbor relationship and its application in KNN classification and clustering methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 7956–7972, Dec. 2024.
- [14] Y. Li, D. Peng, H. Huang, Y. Liu, H. Zheng, and Z. Liu, "Multi-granularity confidence learning for unsupervised text-to-image person re-identification with incomplete modality," *Knowl.-Based Syst.*, vol. 315, Apr. 2025, Art. no. 113304.
- [15] Q. Wang, Q. Hu, Z. Gao, P. Li, and Q. Hu, "AMS-net: Modeling adaptive multi-granularity spatio-temporal cues for video action recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 12, pp. 18731–18745, Dec. 2024.
- [16] S. Xia, Y. Liu, X. Ding, G. Wang, H. Yu, and Y. Luo, "Granular ball computing classifiers for efficient, scalable and robust learning," *Inf. Sci.*, vol. 483, pp. 136–152, May 2019.
- [17] S. Xia, H. Zhang, W. Li, G. Wang, E. Giem, and Z. Chen, "GBNRS: A novel rough set algorithm for fast adaptive attribute reduction in classification," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 3, pp. 1231–1242, Mar. 2022.
- [18] S. Xia, X. Dai, G. Wang, X. Gao, and E. Giem, "An efficient and adaptive granular-ball generation method in classification problem," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 5319–5331, Apr. 2024.
- [19] S. Xia, S. Zheng, G. Wang, X. Gao, and B. Wang, "Granular ball sampling for noisy label classification or imbalanced classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 2144–2155, Apr. 2023.
- [20] Q. Xie et al., "GBG++: A fast and stable granular ball generation method for classification," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 8, no. 2, pp. 2022–2036, Apr. 2024.
- [21] S. Xia et al., "GBRS: A unified granular-ball learning model of pawlak rough set and neighborhood rough set," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 1, pp. 1719–1733, Jan. 2025.
- [22] H. Bai, F. Shen, W. Kong, and J. Feng, "Granular-ball clustering based neighbourhood outliers detection method," in *Proc. 6th Int. Conf. Electron. Technol. (ICET)*, May 2023, pp. 1306–1312.
- [23] X. Su, Z. Yuan, B. Chen, D. Peng, H. Chen, and Y. Chen, "Detecting anomalies with granular-ball fuzzy rough sets," *Inf. Sci.*, vol. 678, Sep. 2024, Art. no. 121016.
- [24] X. Su, S. Cheng, D. Peng, H. Chen, and Z. Yuan, "Fusing multi-granular-ball fuzzy information to detect outliers," *Appl. Soft Comput.*, vol. 175, May 2025, Art. no. 113045.
- [25] Q. Li, Z. Yuan, D. Peng, X. Song, H. Zheng, and X. Su, "Granular-ball fuzzy information-based outlier detector," *Int. J. Approx. Reasoning*, vol. 185, Oct. 2025, Art. no. 109473.
- [26] D. Cheng, Y. Li, S. Xia, G. Wang, J. Huang, and S. Zhang, "A fast Granular-ball-based density peaks clustering algorithm for large-scale data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 12, pp. 17202–17215, Dec. 2024.
- [27] S. Xia, B. Shi, Y. Wang, J. Xie, G. Wang, and X. Gao, "GBCT: Efficient and adaptive clustering via granular-ball computing for complex data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 12, pp. 1–14, Dec. 2025.
- [28] J. Liu, H. Jianye, Y. Ma, and S. Xia, "Unlock the cognitive generalization of deep reinforcement learning via granular ball representation," in *Proc. 41st Int. Conf. Mach. Learn.*, vol. 139, Jul. 2024, pp. 1234–1245.
- [29] S. Xia, X. Ma, Z. Liu, C. Liu, S. Zhao, and G. Wang, "Graph coarsening via supervised granular-ball for scalable graph neural network training," in *Proc. AAAI Conf. Artif. Intell.*, vol. 39, Apr. 2025, pp. 12872–12880.
- [30] H. Li and J. Wang, "From soft clustering to hard clustering: A collaborative annealing fuzzy c-means algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 32, no. 3, pp. 1181–1194, Mar. 2024.
- [31] F. Li, J. Wang, Y. Qian, G. Liu, and K. Wang, "Fuzzy ensemble clustering based on self-coassociation and prototype propagation," *IEEE Trans. Fuzzy Syst.*, vol. 31, no. 10, pp. 3610–3623, Oct. 2023.
- [32] S. Zhao et al., "An accelerator for rule induction in fuzzy rough theory," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 12, pp. 3635–3649, Dec. 2021.
- [33] T. Yin et al., "A robust multilabel feature selection approach based on graph structure considering fuzzy dependency and feature interaction," *IEEE Trans. Fuzzy Syst.*, vol. 31, no. 12, pp. 4516–4528, Dec. 2023.
- [34] Z. Yuan, H. Chen, P. Xie, P. Zhang, J. Liu, and T. Li, "Attribute reduction methods in fuzzy rough set theory: An overview, comparative experiments, and new directions," *Appl. Soft Comput.*, vol. 107, Aug. 2021, Art. no. 107353.
- [35] Z. Yuan, H. Chen, C. Luo, and D. Peng, "MFGAD: Multi-fuzzy granules anomaly detection," *Inf. Fusion*, vol. 95, pp. 17–25, Jul. 2023.
- [36] D. Dubois and H. Prade, "Rough fuzzy sets and fuzzy rough sets," *Int. J. General Syst.*, vol. 17, nos. 2–3, pp. 191–209, 1990.
- [37] Y. Gao, D. Chen, H. Wang, and R. Shi, "Optimization attribute reduction with fuzzy rough sets based on algorithm stability," *IEEE Trans. Fuzzy Syst.*, vol. 32, no. 4, pp. 2052–2062, Apr. 2024.
- [38] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Proc. 6th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining* Cham, Switzerland: Springer, May 2002, pp. 535–548.

- [39] D. Ren, B. Wang, and W. Perrizo, "RDF: A density-based outlier detection method using vertical data representation," in *Proc. 4th IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2004, pp. 503–506.
- [40] W. Hu, J. Gao, B. Li, O. Wu, J. Du, and S. Maybank, "Anomaly detection using local kernel density estimation and context-based regression," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 2, pp. 218–233, Feb. 2020.
- [41] A. Wahid and C. S. R. Annavarapu, "NaNOD: A natural neighbour-based outlier detection algorithm," *Neural Comput. Appl.*, vol. 33, no. 6, pp. 2107–2123, Mar. 2021.
- [42] Q. Zhang et al., "Incremental learning based on granular ball rough sets for classification in dynamic mixed-type decision system," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 9319–9332, Sep. 2023.
- [43] W. Qian, F. Xu, J. Huang, and J. Qian, "A novel granular ball computing-based fuzzy rough set for feature selection in label distribution learning," *Knowl.-Based Syst.*, vol. 278, Oct. 2023, Art. no. 110898.
- [44] S. Cheng, X. Su, B. Chen, H. Chen, D. Peng, and Z. Yuan, "GBMOD: A granular-ball mean-shift outlier detector," *Pattern Recognit.*, vol. 159, Mar. 2025, Art. no. 111115.
- [45] S. Wang, Z. Yuan, S. Cheng, H. Chen, and D. Peng, "Granular-ball computing-based random walk for anomaly detection," *Pattern Recognit.*, vol. 165, Sep. 2025, Art. no. 111588.
- [46] L. Chen, "Topological structure in visual perception," *Science*, vol. 218, no. 4573, pp. 699–700, Nov. 1982.
- [47] Z. Huang, B. Zhang, G. Hu, L. Li, Y. Xu, and Y. Jin, "Enhancing unsupervised anomaly detection with score-guided network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 10, pp. 14754–14769, Oct. 2024.
- [48] Y. Zhou, X. Song, Y. Zhang, F. Liu, C. Zhu, and L. Liu, "Feature encoding with autoencoders for weakly supervised anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2454–2465, Jun. 2022.
- [49] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. H. Chen, "ECOD: Unsupervised outlier detection using empirical cumulative distribution functions," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12181–12193, Mar. 2022.
- [50] Y. Wang and Y. Li, "Outlier detection based on weighted neighbourhood information network for mixed-valued datasets," *Inf. Sci.*, vol. 564, pp. 396–415, Jul. 2021.
- [51] Y. Almardeny, N. Boujnah, and F. Cleary, "A novel outlier detection method for multivariate data," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 9, pp. 4052–4062, Sep. 2022.
- [52] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, "COPOD: Copula-based outlier detection," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2020, pp. 1118–1123.
- [53] X. Li, J. Lv, and Z. Yi, "Outlier detection using structural scores in a high-dimensional space," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 2302–2310, May 2020.
- [54] X. Li, J. Lv, and Z. Yi, "An efficient representation-based method for boundary point and outlier detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 51–62, Jan. 2018.
- [55] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Trans. Knowl. Discovery Data*, vol. 6, no. 1, pp. 1–39.
- [56] Z. Yuan, H. Chen, T. Li, B. Sang, and S. Wang, "Outlier detection based on fuzzy rough granules in mixed attribute data," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 8399–8412, Aug. 2022.
- [57] X. Wang, L. Duan, C. He, Y. Chen, and X. Wu, "An efficient adaptive multi-kernel learning with safe screening rule for outlier detection," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 8, pp. 3656–3669, Aug. 2024.



**Xinyu Su** is currently pursuing the Ph.D. degree in computer science and technology with the College of Computer Science, Sichuan University, Chengdu, China.

His research interests include machine learning, granular computing, and outlier detection.



**Xiwen Wang** is currently pursuing the Ph.D. degree in software engineering with the College of Computer Science, Sichuan University, Chengdu, China.

Her research interests include machine learning and outlier detection.



**Dezhong Peng** received the B.Sc. degree in applied mathematics and the M.Sc. and Ph.D. degrees in computer software and theory from the University of Electronic Science and Technology of China, Chengdu, China, in 1998, 2001, and 2006, respectively.

From 2001 to 2007, he was an Assistant Lecturer and a Lecturer with the University of Electronic Science and Technology of China. From 2007 to 2009, he was a Post-Doctoral Research Fellow with the School of Engineering, Deakin University, Geelong, VIC, Australia. He is currently a Professor with the College of Computer Science, Sichuan University, Chengdu. His research interests include neural networks and signal processing.



**Xiaomin Song** received the B.S. degree in computer and application from Kunming University of Science and Technology, Kunming, China, in 1994.

He is currently the Vice President of Sichuan National Innovation New Vision UHD Video Technology Company Ltd., Chengdu, China. His research interests include ultrahigh definition video processing, hardware acceleration, and engineering of core algorithms.



**Huiming Zheng** received the B.Sc. degree in information engineering from Chengdu University of Technology, Chengdu, China, in 2001.

He is currently the Deputy Director of the Industry Service Center of the Sichuan National Innovation New Vision UHD Video Technology Company Ltd. (National Innovation Center For UHD Video Technology), Chengdu. His research interests include ultrahigh definition video, multisensor fusion, artificial intelligence, visual perception, and data transmission.



**Zhong Yuan** received the M.Sc. degree in mathematics from Sichuan Normal University, Chengdu, China, in 2018, and the Ph.D. degree in computer science and technology from Southwest Jiaotong University, Chengdu, in 2022.

He is currently a Distinguished Associate Researcher with the College of Computer Science, Sichuan University, Chengdu. His research interests include uncertainty artificial intelligence, granular computing, and outlier detection.