Pattern Recognition 165 (2025) 111588

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Granular-ball computing-based Random Walk for anomaly detection

Sihan Wang [a], Zhong Yuan [a,*], Shitong Cheng [a], Hongmei Chen [b], Dezhong Peng [a,c,d]

[a] College of Computer Science, Sichuan University, Chengdu 610065, China
[b] School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China
[c] Sichuan National Innovation New Vision UHD Video Technology Co., Ltd, Chengdu, 610041, China
[d] Tianfu Jincheng Laboratory, Chengdu 610093, China

ARTICLE INFO

Dataset link: https://github.com/optimusprimeyy/GBRAD

Keywords:
Anomaly detection
Outlier detection
Granular-ball computing
Multi-granularity
Random walk

ABSTRACT

Anomaly detection is a key task in data mining, which has been successfully employed in many practical scenarios. However, most existing methods usually analyze the anomalous characteristics of samples at a single and finest granularity, which leads to high computational cost and low efficiency. As one of the significant mathematical models in the theory of granular computing, granular-ball computing can portray the distributional characteristics of data from a multi-granularity perspective. For this reason, this paper proposes an unsupervised anomaly detection method based on granular-ball computing. Firstly, the samples are covered by generating adaptive granular-balls, and the multi-granularity information represented by granular-balls with different sizes can reflect the data distribution characteristics of the corresponding region. Secondly, the granular-balls are used to fit the samples for constructing a state transfer matrix in Random walk. Then, the steady-state distribution is generated using iterative computation and is normalized as the degree of anomaly for each granular-ball. Finally, the anomaly score for each sample is computed by relating the anomaly degree of each granular-ball to the samples it covers. Comparative experiments show that the proposed anomaly detection method performs well on multiple datasets, demonstrating its feasibility and superiority in practical applications. The code is publicly available online at https://github.com/optimusprimeyy/GBRAD.

1. Introduction

Anomaly detection is a trending branch of research in the field of data mining and has been widely used in various fields such as intrusion detection [1], fraud detection [2], and anomaly segmentation [3]. Anomalies (also called outliers) often refer to samples that deviate significantly from the similar behavior of the majority of the data. By detecting anomalies, we can gain insights into potentially critical issues or abnormal behaviors that may require further investigation.

Granular computing is a new computational paradigm for complex problem-solving. Because of its capability of abstracting the global view and approximate solution shown by human beings when dealing with problems at multiple levels and from multiple perspectives, granular computing is gradually becoming an important theory for uncertainty problem-solving. The main models of granular computing theory include fuzzy sets [4], rough sets [5], and their extension models. Granular computing has been widely used in feature selection [6], anomaly detection [7], and other important tasks in the field of data mining.

Granular-Ball Computing (GBC) theory, on the other hand, is a novel research branch in granular computing theory. It mainly generates several GBs to represent the distribution information of data. Each GB can contain a certain number of data samples, which are divided into different regions in the sample space. Since each GB has a different radius and center location, GBC can capture information on different levels of data distribution, thus providing a multi-granularity view. In a multi-granularity view, different levels and details of samples can be considered and analyzed simultaneously. Therefore, introducing multi-granularity analysis can help researchers obtain more comprehensive and detailed information in data to detect anomalies more accurately. It is now widely used in machine learning tasks such as classification [8,9] and clustering [10,11], which effectively improves the performance of models in these tasks. However, there are few studies to date on the application of GBC to anomaly detection tasks.

Random Walk (RW) is a classical model of random processes with a wide range of applications in different fields, including physics, ecology, finance, and computer science. This versatility allows it to be applied to a variety of different types of problems, including image segmentation [12], node clustering [13], and classification [14]. It performs relatively well in anomaly detection [15,16].

* Corresponding author.
E-mail address: yuanzhong@scu.edu.cn (Z. Yuan).

https://doi.org/10.1016/j.patcog.2025.111588
Received 31 August 2024; Received in revised form 19 January 2025; Accepted 7 March 2025
Available online 16 March 2025

However, most existing anomaly detection methods usually analyze the anomalous characteristics of samples at a single and finest granularity, which may lead to problems such as high computational cost and low efficiency. On the other hand, GBC can represent the local distribution characteristics of samples from a multi-granularity perspective, which has the advantages of high efficiency and robustness. Therefore, a GBC-based method using Random walk for Anomaly Detection (GBRAD) is proposed in this paper. The contributions of this paper are listed as follows.

(1) GBC is introduced to characterize the distribution of data in different regions from a multi-granularity perspective, which helps to detect anomalies more accurately;
(2) By using the adaptive GBs to cover the samples and constructing the state transfer matrix, the anomaly degree of samples in GBs can be effectively mined from a multi-granularity perspective, and the computational cost is greatly reduced compared with the sample-based state transfer matrix;
(3) Through experimental comparison with existing mainstream methods, the analysis concludes that the anomaly detection model in this paper achieves better results on multiple datasets, demonstrating its feasibility and superiority in practical applications.

The remainder of this article is organized as follows. Section 2 reviews the existing anomaly detection methods, GBC, and RW theory. Section 3 provides a detailed description of our proposed algorithm GBRAD. Section 4 compares and analyzes GBRAD experimentally with various existing algorithms. We close with concluding remarks in Section 5.

2. Related work

This section provides an overview of current anomaly detection methods, GBC theory, and RW techniques.

2.1. Anomaly detection methods

Currently, anomaly detection methods have been widely researched and are mainly categorized into statistical-based, distance-based, density-based, clustering-based, and deep learning-based methods.

Statistical-based methods focus on identifying samples that do not obey the data distribution as outliers. They are mainly divided into parametric and non-parametric methods. Models constructed by parametric methods generally have a priori knowledge of data distribution, mainly Gaussian Mixture Models (GMM) and Regression models. In contrast, models constructed by non-parametric methods do not have any assumption of prior knowledge for datasets. For example, Latecki et al. [17] proposed a Kernel Density Estimation (KDE) outlier detection model, which is a non-parametric statistical model using kernel functions. Markus et al. [18] proposed an unsupervised histogram-based outlier detection algorithm. The algorithm assumes feature independence and uses histograms to model densities of single-variable features as well as to compute the outlier score for each sample.

Distance-based methods calculate the distance between samples. Then they identify samples with a large distance between them and their nearest neighbors as anomalies. k-Nearest Neighbors (kNN) [19], as one of the basic algorithms in distance-based methods, has been proposed with many related anomaly detection methods. However, some researchers have proposed Reverse k-Nearest Neighbors (RkNN) anomaly detection methods. RkNN methods usually assume that anomalies occupy only a relatively small proportion in the nearest neighbors of normal samples, while the nearest neighbors of anomalies may contain a relatively large number of anomalies. Therefore, it focuses on the ‘‘reverse’’ neighborhood, i.e., which samples identify the current sample as the nearest neighbor. For example, Jin et al. [20] proposed a local outlier detection method, which considered both the neighborhood and the reverse neighborhood of the target in estimating the density distribution.

Density-based methods typically assume that normal samples are generally distributed in areas of higher density and identify samples located in regions of lower data density as anomalies. Huang et al. [21] proposed a non-parametric outlier detection method, which combines the concept of natural neighborhood and density-based methods to achieve better detection results. Abdul et al. [22] proposed a density-based outlier detection algorithm, which uses Weighted Kernel Density Estimation (WKDE) to estimate the density at the sample location. Li et al. [23] proposed a robust anomaly detection method based on the changing rate of directed density ratio, which can effectively adapt to different local densities and distribution flow shapes.

Clustering-based methods typically involve clustering samples with similar attributes into a cluster. They assume that most samples should fall within well-defined clusters, while anomalies may fall outside the boundaries of the clusters or form isolated clusters. According to current research, clustering algorithms can be categorized into three classes based on the detection ways: (1) clustering along with finding anomalies [24], (2) removing anomalies for clustering [25], and (3) identifying anomalies through clustering [26].

Deep Learning-based (DL) methods are categorized into three conceptual paradigms: (1) DL for feature extraction, (2) learning feature representation for normality, and (3) end-to-end anomaly scoring learning according to Ref. [27]. The first category refers to deep learning used only in feature extraction to help anomaly detection tasks extract useful information more efficiently and computationally faster. Among the most frequently employed DL models are VGG and ResNet, etc; The second category is designed for learning an expressive representation of normality. It can be further divided into two subcategories based on how the representation is learned, i.e. whether or not existing shallow anomaly metrics (e.g., distance-based and clustering-based methods) are used to guide learning. The subcategory that does not use shallow metric methods refers to learning the feature representation of samples by optimizing a specific feature learning objective function, which includes Auto-Encoder (AE) and Generative Adversarial Networks (GANs). Zhang et al. [28] proposed a Deep Adversarial Anomaly Detection method (DAAD), which incorporates a comprehensive framework that combines the advantages of GAN adversarial training strategies and AE feature learning; The third category aims to learn anomaly scores in an end-to-end manner through neural networks. The neural networks [29] can learn anomaly scores directly, which requires the design of novel loss functions for anomaly scoring instead of relying on existing anomaly metrics.

2.2. GBC

Due to the advantages of natural interpretability, multi-granularity, and robustness, GBC is now widely integrated into various classical computational models, such as GB rough set [30], GB classification [8,9], and GB clustering [10,11]. Xia et al. [30] proposed a GB Rough Set (GBRS) model by combining the robustness and adaptability of GBC. It can be transformed into GB Pawlak Rough Set (GBPRS) and GB Neighborhood Rough Set (GBNRS). GBRS can not only represent the knowledge in terms of equivalence classes but can also deal with both continuous and discrete data, which can make up for the existing defects of PRS and NRS. Xia et al. [8] combined GB with model KNN and Support Vector Machine (SVM) to propose more efficient classifiers GBKNN and GBSVM. Xie et al. [10] proposed a GB-based Spectral Clustering (GBSC) algorithm. Compared with the traditional spectral clustering algorithms, it uses GBs instead of samples in constructing the similarity matrix, thus the size of the similarity matrix is greatly reduced.

In conclusion, the combination of GBC with classical computational models offers significant advantages and positive benefits to the model

in solving the corresponding tasks. GBC can adaptively depict the local distribution characteristics of data, which is helpful for detecting anomalies. Therefore, GBC has the potential to be integrated with anomaly detection models, resulting in improved model performance.

2.3. Random walk

RW is now widely used in various tasks such as segmentation [12, 31], classification [14], and clustering [32], etc. Dong et al. [12] proposed a new sub-Markov RW algorithm with labeled prior, which outperforms traditional RW algorithms in seed image segmentation. Jiang et al. [31] designed a semantic segmentation network based on SegNet combined with RW (RWSNet). It uses RW on the image, which reduces edge blurring and achieves high-performance semantic segmentation of remote-sensing images. For the weak boundary and spatial fragmentation classification problem, Zhao et al. [14] proposed the Hierarchical RW Network (HRWN). It jointly optimizes the dual-tunnel CNNs and pixel affinity through the RW layers, which enhances the spatial consistency of the deep layers in the network. Wang et al. [33] proposed a seeded RW method to solve the multi-view semi-supervised classification problem. It takes the known labeled samples as random seeds and walks with a certain probability, which is effective in classifying different types of data. Hua et al. [32] proposed a clustering method called RW Gap (RWG) for extracting more information about the cluster structure.

Furthermore, RW has also been applied to anomaly detection tasks. Moonesinghe et al. [16] proposed a random graph-based anomaly detection method. It uses the similarity between samples to construct an adjacency matrix for graph representation and constructs a Markov model for RW so that each sample is assigned an anomaly score. Liu et al. [15] proposed a fuzzy granular anomaly detection method using RW, which makes up for the inadequacy of existing anomaly detection methods that are mainly applicable to deterministic data. Overall, it can be seen that RW is applicable in anomaly detection tasks.

Although there are various types of existing anomaly detection methods, most of them usually analyze the anomalous characteristics of samples at a single finest granularity, which is computationally expensive and inefficient. On the other hand, GBC can portray the local distribution characteristics of samples from a multi-granularity perspective and has the advantages of high efficiency and robustness. In view of this, this paper proposes a GBC-based method using Random walk for Anomaly Detection.

3. GBC-based random walk for anomaly detection

The method GBRAD in this paper is divided into three main phases: GB Generation, RW, and anomaly detection. The first phase is the foundation of the method GBRAD, which involves how to generate the GBs to cover the entire sample space. The second phase describes how to utilize these GBs to construct a novel state transfer matrix and iteratively computing the steady-state distribution. The third phase details how to detect anomalies by the GB-based steady-state distribution.

3.1. GB generation

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \in \mathbb{R}^{n\times b}$ be a data matrix containing n samples and b attributes, with each sample $\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(b)}) \in \mathbb{R}^b$, where $i = 1, \dots, n$. $x_i^{(j)}$ indicates the value of the sample $\mathbf{x}_i$ on the jth attribute, where $j = 1, \dots, b$.

The main idea of GBC is to generate GBs to fully cover or partially cover the sample space [8].

Definition 1. For each GB_k, it contains two items of information: $\mathbf{c}_k$ and r_k denote the center and radius of GB_k, respectively. They can be calculated as follows.

$$\mathbf{c}_k = \frac{1}{n_k}\sum_{l=1}^{n_k}\mathbf{x}_l = \left[\frac{1}{n_k}\sum_{l=1}^{n_k}x_l^{(1)}, \dots, \frac{1}{n_k}\sum_{l=1}^{n_k}x_l^{(b)}\right], \tag{1}$$

$$r_k = \frac{1}{n_k}\sum_{l=1}^{n_k}\|\mathbf{x}_l - \mathbf{c}_k\| = \frac{1}{n_k}\sum_{l=1}^{n_k}\sqrt{\sum_{j=1}^{b}(x_l^{(j)} - c_k^{(j)})^2}, \tag{2}$$

where n_k denotes the number of samples in GB_k, and $\|\cdot\|$ denotes 2-norm. It can be seen that the radius is obtained as the average of the distances from all the samples to the center in the GB. Therefore, the GB is partially covered with samples, which can be seen from Fig. 1.

Fig. 1 shows a brief description of the process of generating GBs. It can be divided into the following four main steps.

Step 1: Initialize the entire dataset as a GB.

Step 2: Perform the 2-Means clustering algorithm on that GB in order to split it into two sub-GBs.

Step 3: Set the splitting criteria. If any GB meets the splitting criteria, it proceeds with the split by the 2-Means. Repeat Step 3 until all GBs no longer meet the splitting criteria.

Step 4: Output the final set of GBs.

The existing criterion for splitting GB is usually that the purity of a GB is less than an artificially set threshold of purity [9]. Purity refers to the proportion of samples belonging to the label that has the most samples in a GB. It can be expressed as $\mathrm{Purity}(GB) = \max_i\left(\frac{|Label_i|}{|GB|}\right)$, where $|Label_i|$ denotes the number of samples belonging to the ith label in the GB, and $|GB|$ denotes the sample number in GB. Obviously, the variable Purity requires labeling information. This leads to the fact that the splitting criterion is not suitable for unsupervised tasks. To solve this problem, Xie et al. [10] proposed a novel criterion, i.e., GB splitting can be performed by satisfying the condition that the weighted sub-ball is of better quality than the parent ball. Based on the idea, we propose a sum distance splitting criterion.

Definition 2. Given a GB_k, let SD_k be the Sum Distance (SD) of GB_k, which can be calculated as

$$SD_k = \sum_{l=1}^{n_k}\|\mathbf{x}_l - \mathbf{c}_k\|. \tag{3}$$

Clearly, SD_k describes the sum of the distances from all samples to the center in GB_k, where $\|\cdot\|$ denotes the 2-norm operation. It is not only related to the distance between samples to the center but is also affected by the number of samples in the GB. When the SD of a GB is smaller relative to other GBs, it may reflect that the samples in the GB are more concentrated near the center or the GB contains fewer samples.

Definition 3. Let SD_k be the SD of the original GB_k. SD_{k_1} and SD_{k_2} denote SDs of the sub-GBs obtained by splitting GB_k using 2-Means, respectively. Then, the sum of the SDs of the sub-GBs is defined as

$$SD^k_{\text{childs}} = SD_{k_1} + SD_{k_2}, \tag{4}$$

It is evident that when a GB splits into two sub-GBs, the sum of the sample numbers in the sub-GBs remains equal to the sample number within the original GB. Thus, the most significant difference between the sum of SD of the two sub-GBs and SD of the original GB is the concentration of the samples in the GB towards the center. Then, we can take advantage of this trait to construct the following splitting criterion.

Definition 4. The splitting criterion is

$$GB_k \Rightarrow \begin{cases} GB_{k_1}, GB_{k_2}, & \text{if } SD_k > SD^k_{\text{childs}}; \\ GB_k, & \text{otherwise.} \end{cases} \tag{5}$$

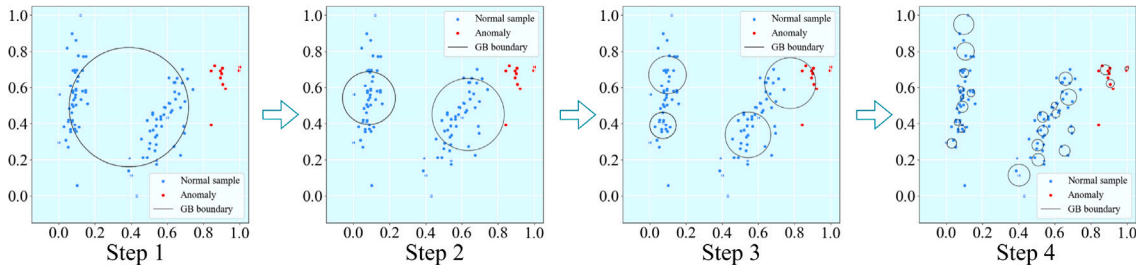


Fig. 1. A brief description of the process of generating GBs under the Iris dataset reduced to 2 dimensions. Red dots indicate anomalies and blue dots indicate normal samples. The black circle indicates the boundaries of GBs. Note that GB boundaries containing only one sample in the GB are not plotted. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

According to Definition 4, if SD^k_{childs} is smaller than SD_k, it suggests that the sum of the distances from samples to the center decreases and that there is a greater concentration of samples inside the sub-GBs. Therefore, GB_k needs to be split into GB_{k_1} and GB_{k_2}, otherwise it will not be split.

Considering the high time cost of splitting GBs using the 2-Means algorithm, we set an early termination criterion. That is, before using the 2-Means algorithm, we stipulate that GBs with fewer samples than a predefined parameter L can no longer be partitioned.

Moreover, considering the problem of failing to split some GBs with too large radii due to noise effects, Xie et al. [10] set an additional criterion: if $r_k > 2 \times \max(\text{mean}(r), \text{median}(r))$, i.e., then GB_k needs to be split, where $\text{mean}(r)$ and $\text{median}(r)$ refer to the mean and median of all GB radii, respectively. Until all GBs no longer meet the splitting criteria, the final set of GBs ($GBs = \{GB_1, GB_2, \ldots, GB_m\}$) is obtained, where m denotes the number of GBs.

3.2. Markov state transfer matrix based on GB

RW is a stochastic process. In the process, the system transfers from one state to another in discrete time steps. The process can be called a Markov chain if it has the property that its transfer to the next state depends only on the current state and is independent of past states.

This property can be expressed in terms of conditional probabilities. Assume a Markov chain with time steps ($t = 0, 1, \ldots, T$) and state space ($\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_l]$). The conditional transfer probability can be expressed as:

$$p(\mathbf{s}^{(t+1)} = \mathbf{s}_j | \mathbf{s}^{(t)} = \mathbf{s}_i, \mathbf{s}^{(t-1)}, \ldots, \mathbf{s}^{(0)}) = p(i, j), \tag{6}$$

where $p(\mathbf{s}^{(t+1)} = \mathbf{s}_j | \mathbf{s}^{(t)} = \mathbf{s}_i)$ refers to the probability that the next state $\mathbf{s}^{(t+1)}$ is $\mathbf{s}_j$ conditional on state $\mathbf{s}^{(t)}$ being $\mathbf{s}_i$ at the current time step t, regardless of the states of $t-1$ time steps. And $p(i, j)$ denotes the probability of transferring from state $\mathbf{s}_i$ to state $\mathbf{s}_j$. The state transfer probability matrix can be represented as $\mathbf{P} = [p_{ij}]_{l \times l}$.

Proposition 1. *The state transfer probability matrix* $\mathbf{P}$ *holds*

(1) *For any* i, j, $p_{ij} \geq 0$, *means that the probability is non-negative.*
(2) *For any* i, $\sum_{\mathbf{s}_j \in \mathbf{S}} p_{ij} = 1$, *indicates that the sum of probabilities from state* $\mathbf{s}_i$ *to other states is* 1.

Liu et al. [15] used the fuzzy granular distance between samples to define the state transfer probability matrix for the RW process. The larger the relative distance between sample $\mathbf{x}_i$ and $\mathbf{x}_j$ is, the larger the corresponding p_{ij}. Based on the property, the probability of an anomaly being visited tends to be higher.

However, calculating the distance between all samples is more complicated when dealing with larger datasets. For this reason, we can reconstruct the state transfer matrix using GBs generated in the previous section instead of the samples, which can greatly reduce the dimensionality of the matrix and improve the time cost.

Definition 5. For any $GB_i, GB_j \in GBs$, the distance between GB_i and GB_j is defined as

$$dis(GB_i, GB_j) = \|\mathbf{c}_i - \mathbf{c}_j\| + r_i + r_j, \tag{7}$$

where $\|\cdot\|$ denotes 2-norm.

Eq. (7) can be expressed as a distance matrix $\mathbf{Dis} = \left[dis(GB_i, GB_j)\right]_{m \times m}$. It can be found that **Dis** is a symmetric matrix. In order to make the difference between GBs more significant, we perform $\min - \max$ normalization on **Dis**.

We further use **Dis** to construct a novel GB-based state transfer matrix **P**. In order for **P** to satisfy Proposition 1, the value of each row of **Dis** is divided by the sum of all values in that row.

Definition 6. The novel state transfer matrix based on GB can be calculated as

$$\mathbf{P} = \begin{bmatrix} \frac{\mathbf{Dis}(1,1)}{\sum_{k=1}^{m} \mathbf{Dis}(1,k)} & \cdots & \frac{\mathbf{Dis}(1,m)}{\sum_{k=1}^{m} \mathbf{Dis}(1,k)} \\ \vdots & \ddots & \vdots \\ \frac{\mathbf{Dis}(m,1)}{\sum_{k=1}^{m} \mathbf{Dis}(m,k)} & \cdots & \frac{\mathbf{Dis}(m,m)}{\sum_{k=1}^{m} \mathbf{Dis}(m,k)} \end{bmatrix}. \tag{8}$$

There are usually large distances between anomalies and most samples. It is clear from the analysis that the denominator is fixed in each value of **P**. When the distance between GB_i and GB_j is larger, then p_{ij} is larger. Therefore, the probability of anomalies in GBs being visited tends to be higher.

3.3. RW based on GB for anomaly detection

By iteratively calculating the probability distribution of states using the state transfer matrix, the final state probability distribution converges to a steady-state distribution regardless of the value of initial probabilities. But it requires that the matrix **P** also needs to be both reductive and non-periodic. For any $GB_k \in GBs$, we can arbitrarily assign GB_k an initial probability value ($s_k^{(0)} = \frac{1}{m}, 1 \leq k \leq m$), where $s_k^{(0)}$ indicates the kth element of $\mathbf{s}$.

Definition 7. According to the state transfer matrix **P**, the process of RW for the state probability distribution is defined as

$$\mathbf{s}^{(t+1)} = d + (1-d)\mathbf{s}^{(t)}\mathbf{P}, \tag{9}$$

where $\mathbf{s}^{(t+1)} = [s_1^{(t+1)}, s_2^{(t+1)}, \ldots, s_m^{(t+1)}]$ denotes the probability distribution vector at the next time step $t+1$. Similarly, $\mathbf{s}^{(t)}$ denotes the probability distribution vector at the current time step t. d is a damping factor indicating that the probability that the walker stays in the current state is d, and the probability of visiting a neighboring GB is $1-d$.

The iteration will stop when the probability distribution vector tends to a steady-state distribution and denotes by $\mathbf{s}^*$ the probability distribution vector obtained at the end of the last iteration. Each value in $\mathbf{s}^*$ can denote the probability that a random walker stays at the corresponding GB, i.e., the Anomaly Degree (AD) of the GB. The larger the AD for a GB, the greater the probability that the samples it contains are anomalous.

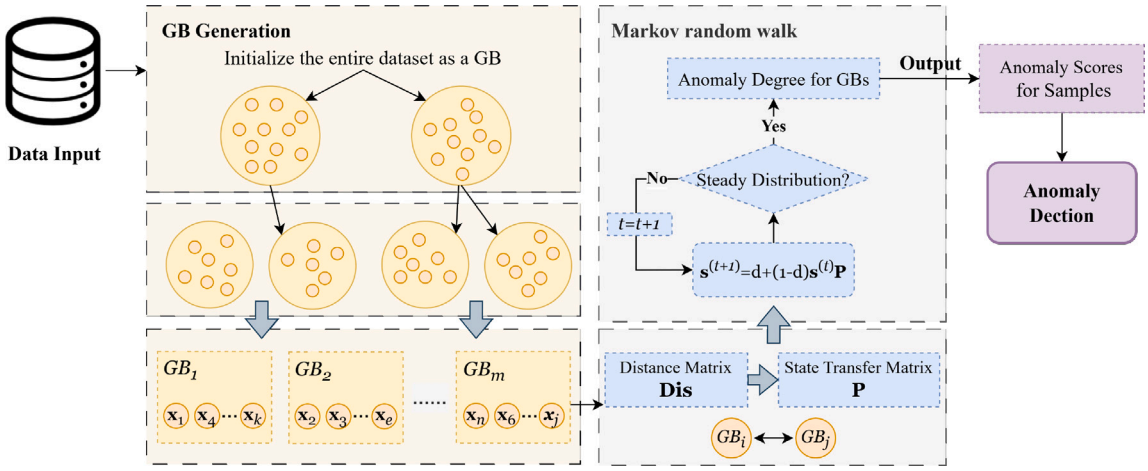


Fig. 2. The framework of GBRAD.

Definition 8. For any $GB_k \in GBs$, the AD for GB_k is computed by

$$AD(GB_k) = s^*(k), \tag{10}$$

where $s^*(k)$ denotes the kth value in the steady-state distribution vector $\mathbf{s}^*$, respectively.

Now that we have the AD of each GB. We need to calculate the anomaly score for each sample in the dataset so that we can determine whether the sample is anomalous or not by its anomaly score.

Definition 9. For any $\mathbf{x}_i \in \mathbf{X}$, if $\mathbf{x}_i$ is covered by GB_k, then the anomaly score of $\mathbf{x}_i$ can be calculated by

$$AS(\mathbf{x}_i) = AD(GB_k) \times W(GB_k), \tag{11}$$

where $W(GB_k) : U \rightarrow [0,1]$ is a weight coefficient computed by $1 - \sqrt[3]{\frac{|n_k|}{|\mathbf{X}|}}$, and n_k denotes the sample number in GB_k. When there are fewer samples covered by a GB, it can reflect laterally that the samples in it are more deviated from the distribution of most other samples, which makes them more likely to be anomalies in the dataset. Therefore, it is reasonable that we assign a greater weight to that GB.

Finally, the following rules are set to determine whether the sample is anomalous or normal.

Definition 10. Given a threshold ϵ, for any $\mathbf{x}_i \in \mathbf{X}$, the decision rules of anomaly detection are presented as

$$Rules: \begin{cases} \text{IF } AS(\mathbf{x}_i) > \epsilon, & \text{THEN } \mathbf{x}_i \text{ is anomalous;} \\ \text{IF } AS(\mathbf{x}_i) \leq \epsilon, & \text{THEN } \mathbf{x}_i \text{ is normal.} \end{cases} \tag{12}$$

Based on the above description of the method, the framework of the method GBRAD can be illustrated in Fig. 2.

3.4. Anomaly detection algorithm

The pseudo-code of the algorithm GBRAD is presented in this subsection, and the time complexity is analyzed.

In Algorithm 1, the calculation process can be divided into three stages as follows:

Stage 1: First, the whole data matrix is initially regarded as a GB in Step 1. Then, we enter the first **repeat** loop (Steps 2–11), which is to split the GBs according to the splitting criterion until the number of GBs no longer changes. Since the 2-Means method is used for each split, the time complexity of this loop is $O(bn)$.

Stage 2: The algorithm enters the second **repeat** loop (Steps 12–19), which only re-splits those GBs with larger radii, also with a time complexity of $O(bn)$.

Stage 3: In Step 21, the algorithm constructs the GB-based distance matrix **Dis** and normalizes it to obtain the state transfer matrix **P**. The construction time complexity of the matrices is $O(m^2 b)$. Further, the algorithm enters the third **repeat** loop (Steps 23-25) for RW until a steady-state distribution is obtained. In Steps 26-31, the AD is calculated for each GB, and the anomaly score for each sample is calculated. The computational complexity is $O(n)$.

Thus, in the worst case, the time complexity of Algorithm 1 is $O(m^2 b)$.

4. Experiments and analyses

In this section, experiments are systematically conducted on 16 public datasets, and the performance of the proposed algorithm is evaluated based on the experimental results. Moreover, the sensitivity of the parameters and the statistical test are analyzed.

4.1. Datasets

We collected a total of 16 datasets on the public web [1,2]. The datasets on the pages are typically used to evaluate the performance of anomaly detection methods [34,35], and are therefore suitable for use in this experiment.

In these datasets, the sample number extends beyond 101 to 6870, and the count of attributes reaches within the range of 8 to 279. Table 1 shows specific information about the experimental dataset.

4.2. Comparison algorithms

In total, we employed the following 11 comparative methods to conduct experiments to explore our proposed method's performance.

(1) A reverse unreachability-based method (NC) (2016) [36]: This method can identify the outliers and boundary points by exploiting the property that the reverse unreachability score is higher for samples with lower density.

[1] https://github.com/BELLoney/Outlier-detection
[2] http://odds.cs.stonybrook.edu

Algorithm 1: GBRAD Algorithm

Input: $\mathbf{X}, d, L$.
Output: Anomaly Scores (AS).

$GBs \leftarrow \mathbf{X}$;
repeat
  for *each* $GB_k \in GBs$ **do**
    if $|GB_k| \geq L$ **then**
      Calculate SD_k, SD^k_{childs};
      if $SD_j > SD^k_{\text{childs}}$ **then**
        Split GB_k into GB_{k_1}, GB_{k_2}; Add GB_{k_1}, GB_{k_2} to GBs, and remove GB_k from GBs;
      end
    end
  end
end
until *The number of GB in GBs remains the same.*;
repeat
  Calculate mean(r), median(r);
  for *each* $GB_k \in GBs$ **do**
    if $r_k \geq 2 \times \max(\text{mean}(r), \text{median}(r))$ **then**
      Split GB_k into GB_{k_1}, GB_{k_2}; Add GB_{k_1}, GB_{k_2} to GBs, and remove GB_k from GBs;
    end
  end
until *The number of GB in GBs remains the same.*;
Obtain the final set of GBs ($GBs = \{GB_1, GB_2, \cdots, GB_m\}$);
Construct **Dis** and **P** by Eqs. (7) and (8);
Initialize $t = 0, \mathbf{s}^{(t)} = [\frac{1}{m}, \frac{1}{m}, \cdots, \frac{1}{m}]$;
repeat
  $\mathbf{s}^{(t+1)} = d + (1-d)\mathbf{s}^{(t)}\mathbf{P}$; $t \leftarrow t+1$;
until ($\|\mathbf{s}^{(t+1)} - \mathbf{s}^{(t)}\| \leq 10^{-3}$);
for *each* $GB_k \in GBs$ **do**
  Calculate $AD(GB_k)$ by Eq. (10);
end
for *each* $\mathbf{x}_i \in \mathbf{X}$ **do**
  Calculate $AS(\mathbf{x}_i)$ by Eq. (11);
end
return AS.

Table 1
The specific information of datasets.

ID	Datasets	Abbreviations	#Attributes	#Samples	#Anomalies
1	Annealing_variant1	Annealing	38	798	42
2	Arrhythmia_variant1	Arrhythmia	279	452	66
3	Breast_cancer_variant1	Breast	9	286	85
4	Lymphography	Lymphography	18	148	6
5	Musk	Musk	166	3062	97
6	Pendigits	Pendigits	16	6870	156
7	Pima_TRUE_55_variant1	Pima	9	555	55
8	Satimage	Satimage	36	5803	71
9	Sonar_M_10_variant1	Sonar	60	107	10
10	Waveform_0_100_variant1	Waveform	21	3443	100
11	Wbc_malignant_39_variant1	Wbc	9	483	39
12	Wdbc_M_39_variant1	Wdbc	31	396	39
13	Wine	Wine	13	129	10
14	Wpbc_variant1	Wpbc	33	198	47
15	Yeast_ERL_5_variant1	Yeast	8	1141	5
16	Zoo_variant1	Zoo	16	101	17

(2) A Self-Representation-based Outlier detection method (SRO) (2017) [37]: It obtains an asymmetric affinity matrix using the property that samples can be represented as sparse linear combinations of each other. It exploits the matrix to construct a weighted directed graph in which RW is performed to detect outliers.
(3) A joint learning framework for detecting outliers in MIXed-type data (MIX) (2019) [38]: It can be used to detect anomalies in mixed-type data by iteratively optimizing the numerical and nominal space scoring stages separately and finally capturing final anomalies in a joint manner.
(4) Approximate structural score-based anomaly detection method (ApproE) (2020) [39]: This method calculates the approximate structural scores by measuring the angular variance weighted by the data representation.
(5) COPula-based Outlier Detector (COPOD) (2020) [40]: The method predicts the tail probability of each sample to determine its degree of extremity using empirical copula.
(6) Variance structural score-based anomaly detection method (VarE) (2020) [39]: This method calculates the variance structural scores of samples which are derived from measuring the angular variance weighted by data representation.
(7) Directed density ratio Changing Rate-based Outlier Detection method (DCROD) (2022) [23]: The method calculates the change degree of the directed density ratio of the sample by increasing the size of nearest neighbors, which is summed to obtain the final outlier score.
(8) Empirical-Cumulative-distribution-based Outlier Detection method (ECOD) (2022) [41]: This method calculates the outlier value for each sample by aggregating the tail probabilities estimated for each dimension.
(9) Incomplete Local and Global Neighborhood Information-based method (ILGNI) (2023) [42]: This method considers the smooth distribution of Markov models on the network and computes outliers.
(10) Weighted Fuzzy-Rough Density-based Anomaly detection method (WFRDA) (2023) [34]: It combines the density and fuzziness of each sample to construct its anomaly score.
(11) Fuzzy Granular Anomaly Score-based anomaly detection method (FGAS) (2023) [15]: The method employs the fuzzy granularity distances between samples as a state transfer matrix, and obtains a stationary distribution to represent the anomaly scores.

In the experimental settings, for the parameter k involved in the algorithms NC and DCROD, we let it range from 2 to 60 with a step of 1. The parameter λ of the algorithms ApproE and VarE is varied from -3 to 3 in steps of 1. We run the MIX experiment 10 times and then take the maximum AUC value as the final result. We let the parameter α in the algorithm SRO range from 2 to 20, with the step of 1. The parameter δ in WFRDA is set to range from 0.1 to 2 within steps of 0.1. The parameter δ in FGAS is set to range from 0 to 1 within steps of 0.05. The algorithms COPOD, ECOD, and ILGNI do not involve parameter settings. Moreover, in our algorithm GBRAD, the parameter L in GB splitting is varied from 3 to 20 in steps of 1, and the parameter d in the process of RW is set to 0.1 by default according to Ref. [43]. We give the experimental basis for setting d as 0.1 in Section 4.5. The experimental result with the highest AUC value is taken as the final result.

4.3. Evaluation indicators

The evaluation metrics ROC (Receiver Operating Characteristic) curve and AUC (Area Under Curve) are generally used for outlier detection performance evaluation [34,35]. For each dataset, the outlier detection algorithm obtains an anomaly score for each sample, and samples are sorted in descending order based on their anomaly scores.

For any k, the samples at the top k of the ranking are identified as anomalies and form a set of anomalies $A(k)$. A_{true} and X_{normal} represent the set of all true anomalies and all normal samples in the dataset, respectively.

The ROC curve consists of the horizontal coordinate $FPR(k)$ (False Positive Rate) and the vertical coordinate $TPR(k)$ (True Positive Rate). Their respective equations are as follows.

$$FPR(k) = \frac{|A(k) - A_{\text{true}}|}{|X_{\text{normal}}|} \times 100\%; \tag{13}$$

Table 2
Comparative results of AUC experiments.

Datasets	NC	SRO	MIX	ApproE	COPOD	VarE	DCROD	ECOD	ILGNI	WFRDA	FGAS	GBRAD
Annealing	0.6526	0.6926	**0.8636**	0.7526	0.7967	0.6195	0.7552	0.7874	0.6864	0.7387	0.7879	0.8421
Arrhythmia	0.7337	0.7570	**0.8264**	0.6907	0.8046	0.7385	0.7961	0.8071	0.8164	0.8260	0.8238	0.8263
Breast	0.6420	0.6287	**0.7336**	0.6416	0.6322	0.6369	0.6035	0.6555	0.6587	0.6536	0.6806	0.6958
Lymphography	0.9824	0.9695	**0.9977**	0.4765	0.9941	0.5070	0.9542	0.9965	0.9906	0.9930	0.9894	0.9566
Musk	0.5719	0.6892	0.9986	0.0716	0.9463	0.3457	0.8633	0.9559	0.9930	**1.0000**	**1.0000**	**1.0000**
Pendigits	0.5367	0.5617	0.8857	0.7869	0.9048	0.6649	0.7118	0.9274	0.8625	0.9423	0.9458	**0.9603**
Pima	0.8882	0.6853	0.9447	0.9200	0.9423	0.9040	0.9278	0.9472	0.9705	**0.9745**	0.9007	0.9472
Satimage2	0.5896	0.5115	0.9953	0.6621	0.9745	0.5371	0.8649	0.9649	0.9941	0.9717	0.9867	**0.9972**
Sonar	**0.9985**	0.5619	0.9887	0.9412	0.9856	0.9546	0.9887	0.9660	0.9649	0.9907	0.9948	0.9964
Waveform	0.6957	0.7339	0.6941	0.5952	0.7343	0.4730	0.7433	0.6084	0.7052	0.7043	0.6987	**0.8982**
Wbc	0.7559	0.8695	0.9969	**0.9973**	0.9955	0.9973	0.9764	0.9955	0.9880	0.9968	0.9971	**0.9972**
Wdbc	0.8343	0.4890	0.9961	0.9998	0.9956	0.9973	0.9682	0.9591	0.9512	0.9988	0.9977	**1.0000**
Wine	0.9130	0.3706	0.9437	0.9387	0.8672	0.9412	0.8849	0.7328	0.7445	0.9151	0.8849	**1.0000**
Wpbc	0.5099	0.5037	0.5452	0.5573	0.5226	0.5516	0.5129	0.4810	0.4210	0.5277	0.5941	**0.6146**
Yeast	0.9697	0.9945	**1.0000**	**1.0000**	0.9974	**1.0000**	0.9898	0.9952	0.9810	0.9975	0.9986	**1.0000**
Zoo	0.7234	0.7465	0.7689	0.3235	0.5497	0.6828	0.7521	0.5819	0.5889	0.5259	0.5735	**0.9695**
Average	0.7498	0.6728	0.8862	0.7097	0.8527	0.7220	0.8308	0.8351	0.8323	0.8598	0.8659	**0.9188**

$$TPR(k) = \frac{|A(k) \cap A_{\text{true}}|}{|A_{\text{true}}|} \times 100\%, \tag{14}$$

where FPR denotes the proportion of all normal samples that are incorrectly predicted to be anomalies. And TPR denotes the proportion of all anomaly samples that are correctly predicted as anomalies.

However, if multiple ROC curves on a dataset are intertwined, it can become difficult to evaluate the performance of the corresponding algorithm. For this purpose, we use the AUC metric for a more intuitive evaluation. AUC refers to the area under the ROC curve, which ranges from 0 to 1. The closer the AUC is to 1, the better the detection performance of the corresponding algorithm is. When AUC is close to 0.5, it indicates that the detection performance of the algorithm is not much different from that of random guessing. And it is calculated as

$$\text{AUC} = \underset{\mathbf{x}_i \in A_{\text{true}}, \mathbf{x}_j \in X_{\text{normal}}}{\text{Mean}} \begin{cases} 1, & AS(\mathbf{x}_i) > AS(\mathbf{x}_j); \\ 0.5, & AS(\mathbf{x}_i) = AS(\mathbf{x}_j); \\ 0, & AS(\mathbf{x}_i) < AS(\mathbf{x}_j), \end{cases} \tag{15}$$

where $AS(\mathbf{x}_i)$ denotes the anomaly score of the sample $\mathbf{x}_i$.

4.4. Analysis of experimental results

Fig. 3 shows the ROC curves for each algorithm on each dataset. The dark blue curve with a solid diamond is the ROC of the proposed algorithm GBRAD. It can be noticed that in most of the datasets (e.g., Musk, Pendigits, Satimage, Waveform, Wbc, Wdbc, Wine, Yeast, and Zoo), the dark blue curve is clearly closest to the upper left corner compared to the other curves. In datasets Annealing, Arrhythmia, Breast, Pima, Sonar, and Wpbc, the GBRAD curve partially overlaps with those of the other algorithms. It is not possible to visually analyze which algorithm performs best, so AUC metrics are needed to assess comparative performance. The dark blue curves of GBRAD in datasets Lymphography and Pima, are not closest to the upper left corner, slightly worse than algorithms such as WFRDA and FGAS, etc.

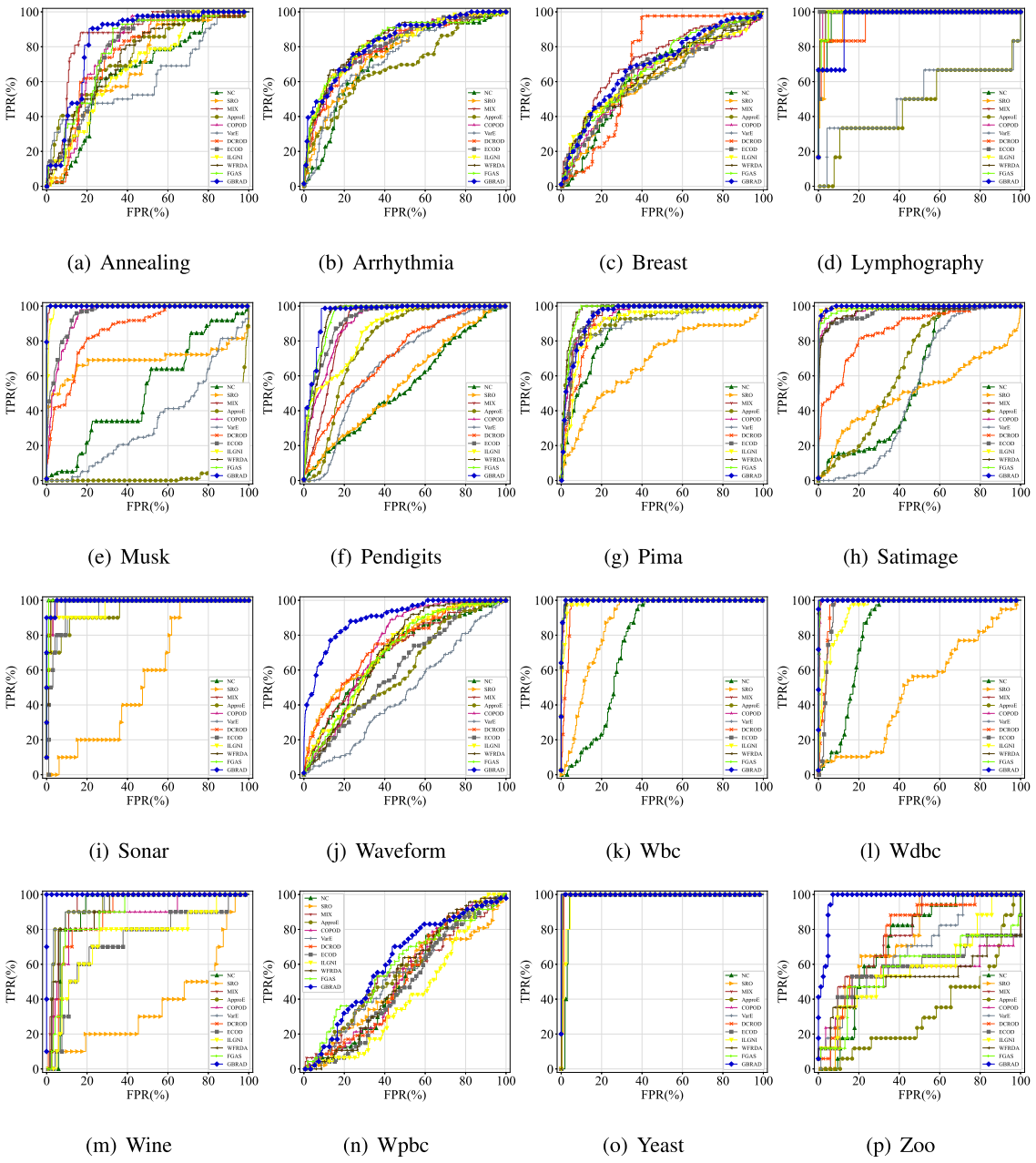
Table 2 shows the results of the experimental comparison of the AUCs, where the bold numbers indicate the corresponding algorithm that performed best on that row of the dataset, as well as the underlined number represent the next best algorithm in performance in that row. It can be found that the number of best AUC values achieved by algorithms NC, SRO, MIX, ApproE, COPOD, VarE, DCROD, ECOD, ILGNI, WFRDA, FGAS, and GBRAD on these 16 datasets are 1, 0, 5, 2, 0, 1, 0, 0, 0, 2, 1, and 10, respectively. In addition to this, GBRAD also has 4 next-best AUC values. On the other hand, the average value of AUCs can effectively evaluate the performance of various algorithms. The average AUC values of these 12 algorithms are 0.7498, 0.6728, 0.8862, 0.7097, 0.8527, 0.7220, 0.8308, 0.8351, 0.8323, 0.8598, 0.8659, and 0.9188, respectively.

Among them, GBRAD provides the best results, which is much higher than the other methods. Therefore, it can be concluded that the detection performance of GBRAD is better than the other 11 algorithms on most of the datasets. The superiority of the method GBRAD in terms of overall results can be attributed to its ability to leverage the multi-granularity property of GBC. The multi-granularity representation allows the algorithm GBRAD to comprehensively analyze the characteristics of the data distribution in different regions of the sample space. It helps to accurately capture anomalies, which tend to exhibit significantly different characteristics from normal samples at a particular level of granularity in the data distribution. In some datasets such as Lymphography and Pima, GBRAD did not reach the highest value in AUC. This may be due to the fact that the algorithm GBRAD directly assigns the anomaly degree of the GB to the anomaly scores of the samples it covers, whereas in reality there should be differences in the anomaly degree of samples belonging to the same GB. As a result, there exist some normal samples that are not abnormal enough to be labeled as anomalies, but are misclassified as anomalies because they belong to the same GB as some actual anomalies. And the dataset Lymphography is a nominal dataset, GBRAD uses the Euclidean distance to calculate the GB radius, which may not be very applicable to nominal data. Consequently, a few normal samples may be incorrectly assigned higher anomaly scores, which results in the failure of the AUC to reach the optimal value.

4.5. Parameter sensitivity analysis

We proceed to analyze the sensitivity of the algorithm GBRAD to the parameters L and d. Fig. 4 shows the AUC curves of GBRAD with respect to parameters L and d on six randomly selected experimental datasets. It can be clearly found that under the same L, when d changes from 0 to 1, the AUC of most datasets remains almost unchanged. This shows that the performance of the algorithm GBRAD is not sensitive to the parameter d. Therefore, it is reasonable to set d as 0.1 in our comparative experiment.

Figs. 5(a)–5(b) plot the curves of AUC with respect to the parameter L for the first 8 and the last 8 datasets, where d is set to 0.1. L ranges from 3 to 40, with a step size of 1. As can be seen from Fig. 5, the AUC curves of most datasets have relatively small fluctuations relative to L, such as the datasets Musk, Satimage, and Yeast. However, the AUC curves of the datasets Lymphography, Waveform, Wine, Wpbc, and Zoo have large fluctuations and no obvious trend. The reason may be that the distribution characteristics of these datasets are not obvious. When the distribution characteristics of the dataset are obvious, the randomness of the 2-means algorithm will be relatively small, and the results of GB splitting will be relatively stable and consistent.



(a) Annealing (b) Arrhythmia (c) Breast (d) Lymphography

(e) Musk (f) Pendigits (g) Pima (h) Satimage

(i) Sonar (j) Waveform (k) Wbc (l) Wdbc

(m) Wine (n) Wpbc (o) Yeast (p) Zoo

Fig. 3. Experimental comparison results on ROC.

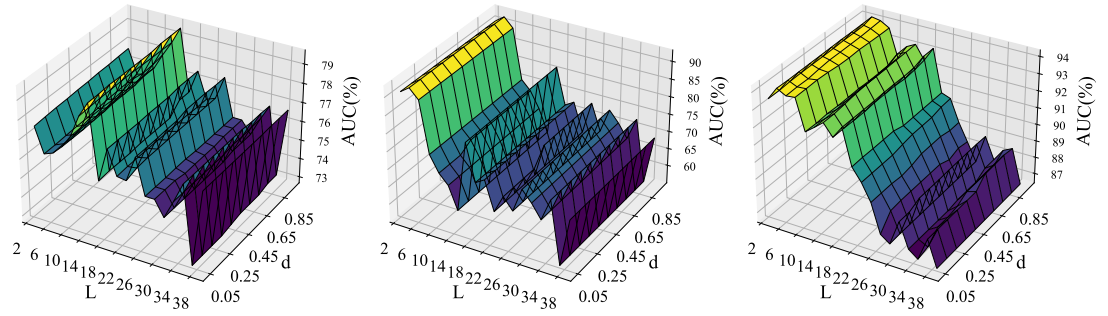
4.6. Statistical test

We continue with a statistical test to explore the statistical difference of each algorithm [34]. For each dataset, the AUC values of all the algorithms are to be sorted in ascending order and assigned ordinal numbers $1, 2, \ldots$. In cases of multiple algorithms having the same AUC value, their current ordinal numbers are averaged as their respective ordinal numbers. First, we exploit Friedman's test to estimate the existence of significant differences between all the algorithms. A total of 16 datasets are collected in this experiment as well as a total of 12 algorithms are compared. Therefore, it can be obtained that the variable τ_F in Friedman's test follows the F-distribution with 11 and 165 degrees of freedom. Setting the significance level as 0.05, it is calculated that $\tau_F = 8.9649$, which is much larger than the critical value of 1.8471. Thus, it is evident that the initial hypothesis stating that all algorithms perform equally cannot be accepted.

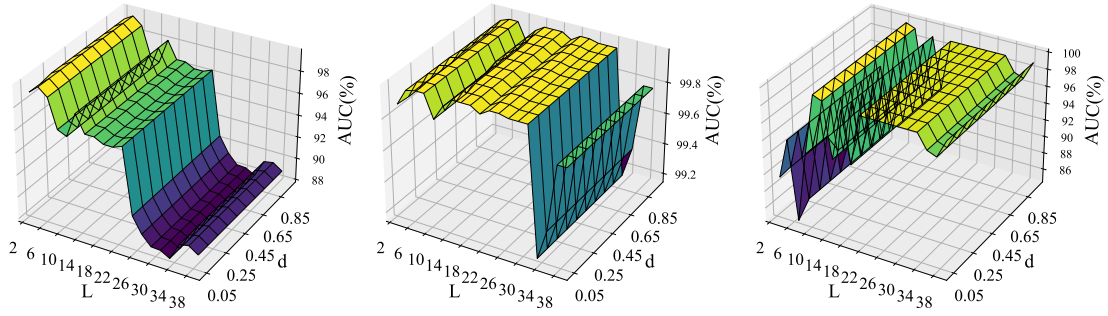
Subsequently, we use the Nemenyi post-hoc test to compare the performance difference between algorithms. The critical distance for the Nemenyi experiment is $CD = 4.1659$ when the significance level is 0.05. The mean ordinal values obtained by each algorithm are plotted on the numerical axis of the Nemenyi test plot, respectively. If a red horizontal line connects several of the algorithms, it means that the difference in mean ordinal values between them is less than the critical distance CD. Thus, it cannot be proven that they are statistically different. Fig. 6 demonstrates that the algorithm GBRAD has no connection with COPOD, ECOD, ILGNI, DCROD, ApproE, VarE, NC, and SRO, suggesting that GBRAD is significantly different from these algorithms. GBRAD is connected with MIX, FGAS, and WFRDA, indicating that it is unable to prove that GBRAD is statistically different from these algorithms.

5. Conclusion

This paper proposed an anomaly detection method based on GBC using the tool of RW. Through the multi-granularity of GBs, the technique effectively captured the distributional characteristics of data, providing

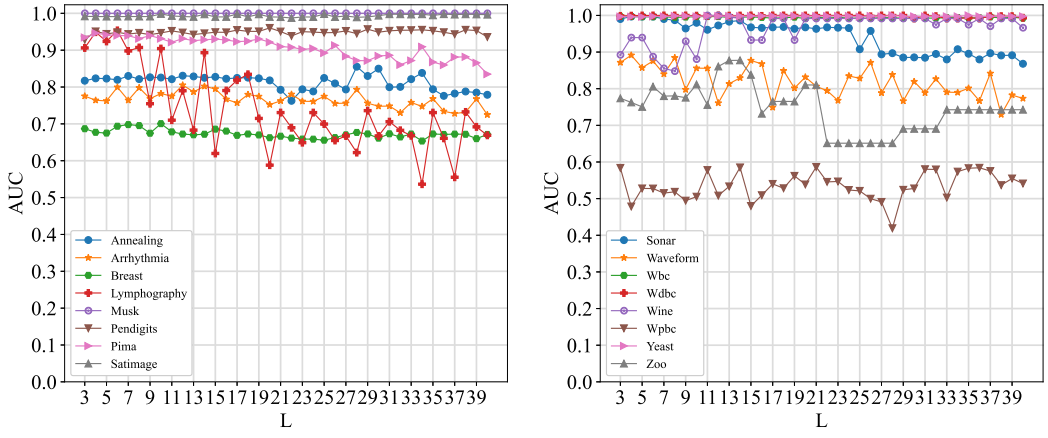


(a) Arrhythmia (b) Lymphography (c) Pima



(d) Sonar (e) Waveform (f) Wine

Fig. 4. AUC varies with the parameter λ and d.



(a) Datasets 1-8 (b) Datasets 9-16

Fig. 5. Variation curve of AUC with L.

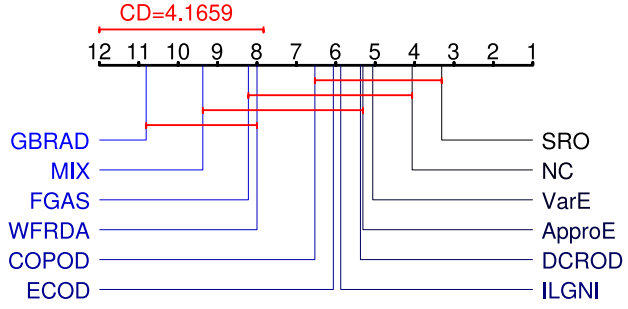


Fig. 6. Nemenyi's test figure on AUC.

a more comprehensive analysis of anomalies. The use of adaptive GBs and the construction of a novel GB-based state transfer matrix for RW improved computational efficiency without compromising detection accuracy. The experimental results demonstrated the superiority of the proposed method compared to existing approaches. The method showcased excellent performance across multiple datasets, proving its feasibility and practical applicability. Overall, this research contributes to anomaly detection by introducing a novel method that combines GBC theory. However, our method directly maps the degree of anomaly of GB to the anomaly scores of those samples it covers. It is possible that the actual anomaly degree of these samples will vary. This may improve the anomaly detection performance of the model if information about the differences between samples within a GB can continue to be mined.

In future work, we will explore applying our methods to open-set recognition. Open set recognition [44,45] is a challenging task aimed at identifying data points in categories that have not been seen during training. It is critical in areas such as anomaly detection and unknown detection. Since our method is unsupervised, it signifies that it does not rely on category labels for learning, which provides a unique advantage in dealing with open-set data. We believe that this extension will further enhance the utility of our method and make a substantial contribution to the development of automatic anomaly detection systems.

CRediT authorship contribution statement

Sihan Wang: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Zhong Yuan:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition. **Shitong Cheng:** Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Hongmei Chen:** Resources, Project administration, Funding acquisition. **Dezhong Peng:** Validation, Funding acquisition.

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.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62306196, 62372315, and 62376230), the Sichuan Science and Technology Program (2024YFHZ0089, 2024NSFTD0049, 2024YFHZ0144, and 2024NSFSC0443), Chengdu Science and Technology Project (2023-XT00-00004-GX), and the Fundamental Research Funds for the Central Universities (YJ202245).

Data availability

The URL link to the research data and relevant code for the article is as follows: https://github.com/optimusprimeyy/GBRAD.

References

[1] J. Xie, H. Wang, J.M. Garibaldi, D. Wu, Network intrusion detection based on dynamic intuitionistic fuzzy sets, IEEE Trans. Fuzzy Syst. 30 (9) (2022) 3460–3472.
[2] K.G. Al-Hashedi, P. Magalingam, Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019, Comput. Sci. Rev. 40 (2021) 100402.
[3] C. Baur, S. Denner, B. Wiestler, N. Navab, S. Albarqouni, Autoencoders for unsupervised anomaly segmentation in brain MR images: a comparative study, Med. Image Anal. 69 (2021) 101952.
[4] Q. Guo, K. Liu, T. Xu, P. Wang, X. Yang, Fuzzy feature factorization machine: Bridging feature interaction, selection, and construction, Expert Syst. Appl. 255 (2024) 124600.
[5] D. Qian, K. Liu, J. Wang, S. Zhang, X. Yang, Attribute reduction based on directional semi-neighborhood rough set, Int. J. Mach. Learn. Cybern. (2024) 1–13.
[6] D. Qian, K. Liu, S. Zhang, X. Yang, Semi-supervised feature selection by minimum neighborhood redundancy and maximum neighborhood relevancy, Appl. Intell. (2024) 1–15.
[7] B. Chen, Y. Li, D. Peng, H. Chen, Z. Yuan, Fusing multi-scale fuzzy information to detect outliers, Inf. Fusion 103 (2024) 102133.
[8] S. Xia, Y. Liu, X. Ding, G. Wang, H. Yu, Y. Luo, Granular ball computing classifiers for efficient, scalable and robust learning, Inform. Sci. 483 (2019) 136–152.
[9] S. Xia, X. Dai, G. Wang, X. Gao, E. Giem, An efficient and adaptive granular-ball generation method in classification problem, IEEE Trans. Neural Netw. Learn. Syst. 35 (4) (2022) 5319–5331.
[10] J. Xie, W. Kong, S. Xia, G. Wang, X. Gao, An efficient spectral clustering algorithm based on granular-ball, IEEE Trans. Knowl. Data Eng. 35 (9) (2023) 9743–9753.
[11] S. Xia, D. Peng, D. Meng, C. Zhang, G. Wang, E. Giem, W. Wei, Z. Chen, Ball k k-means: Fast adaptive clustering with no bounds, IEEE Trans. Pattern Anal. Mach. Intell. 44 (1) (2020) 87–99.
[12] X. Dong, J. Shen, L. Shao, L. Van Gool, Sub-Markov random walk for image segmentation, IEEE Trans. Image Process. 25 (2) (2015) 516–527.
[13] C. Ji, T. Zhao, Q. Sun, X. Fu, J. Li, Higher-order memory guided temporal random walk for dynamic heterogeneous network embedding, Pattern Recognit. 143 (2023) 109766.
[14] X. Zhao, R. Tao, W. Li, H.-C. Li, Q. Du, W. Liao, W. Philips, Joint classification of hyperspectral and LiDAR data using hierarchical random walk and deep CNN architecture, IEEE Trans. Geosci. Remote Sens. 58 (10) (2020) 7355–7370.
[15] C. Liu, Z. Yuan, B. Chen, H. Chen, D. Peng, Fuzzy granular anomaly detection using Markov random walk, Inform. Sci. 646 (2023) 119400.
[16] H. Moonesinghe, P.-N. Tan, Outlier detection using random walks, in: 2006 18th IEEE International Conference on Tools with Artificial Intelligence, ICTAI'06, IEEE, 2006, pp. 532–539.
[17] L.J. Latecki, A. Lazarevic, D. Pokrajac, Outlier detection with kernel density functions, in: International Workshop on Machine Learning and Data Mining in Pattern Recognition, Springer, 2007, pp. 61–75.
[18] M. Goldstein, A. Dengel, Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm, KI- 2012: Poster Demo Track 1 (2012) 59–63.
[19] X. Wang, X.L. Wang, Y. Ma, D.M. Wilkes, A fast MST-inspired kNN-based outlier detection method, Inf. Syst. 48 (2015) 89–112.
[20] W. Jin, A.K. Tung, J. Han, W. Wang, Ranking outliers using symmetric neighborhood relationship, in: Advances in Knowledge Discovery and Data Mining: 10th Pacific-Asia Conference, PAKDD 2006, Singapore, April 9-12, 2006. Proceedings 10, Springer, 2006, pp. 577–593.
[21] J. Huang, Q. Zhu, L. Yang, J. Feng, A non-parameter outlier detection algorithm based on natural neighbor, Knowl.-Based Syst. 92 (2016) 71–77.
[22] A. Wahid, C.S.R. Annavarapu, NaNOD: A natural neighbour-based outlier detection algorithm, Neural Comput. Appl. 33 (2021) 2107–2123.
[23] K. Li, X. Gao, S. Fu, X. Diao, P. Ye, B. Xue, J. Yu, Z. Huang, Robust outlier detection based on the changing rate of directed density ratio, Expert Syst. Appl. 207 (2022) 117988.
[24] H. Liu, J. Li, Y. Wu, Y. Fu, Clustering with outlier removal, IEEE Trans. Knowl. Data Eng. 33 (6) (2019) 2369–2379.
[25] J. Yang, S. Rahardja, P. Fränti, Mean-shift outlier detection and filtering, Pattern Recognit. 115 (2021) 107874.
[26] S.A.N. Nozad, M.A. Haeri, G. Folino, SDCOR: Scalable density-based clustering for local outlier detection in massive-scale datasets, Knowl.-Based Syst. 228 (2021) 107256.
[27] G. Pang, C. Shen, L. Cao, A.V.D. Hengel, Deep learning for anomaly detection: A review, ACM Comput. Surv. 54 (2) (2021) 1–38.
[28] X. Zhang, J. Mu, X. Zhang, H. Liu, L. Zong, Y. Li, Deep anomaly detection with self-supervised learning and adversarial training, Pattern Recognit. 121 (2022) 108234.
[29] G. Pang, C. Yan, C. Shen, A.v.d. Hengel, X. Bai, Self-trained deep ordinal regression for end-to-end video anomaly detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12173–12182.
[30] S. Xia, C. Wang, G. Wang, X. Gao, W. Ding, J. Yu, Y. Zhai, Z. Chen, Gbrs: a unified granular-ball learning model of pawlak rough set and neighborhood rough set, IEEE Transactions on Neural Networks and Learning Systems 36 (1) (2025) 1719–1733.
[31] J. Jiang, C. Lyu, S. Liu, Y. He, X. Hao, RWSNet: a semantic segmentation network based on SegNet combined with random walk for remote sensing, Int. J. Remote Sens. 41 (2) (2020) 487–505.
[32] J. Hua, J. Yu, M.-S. Yang, Fast clustering for signed graphs based on random walk gap, Soc. Netw. 60 (2020) 113–128.
[33] S. Wang, Z. Wang, K.-L. Lim, G. Xiao, W. Guo, Seeded random walk for multi-view semi-supervised classification, Knowl.-Based Syst. 222 (2021) 107016.
[34] Z. Yuan, B. Chen, J. Liu, H. Chen, D. Peng, P. Li, Anomaly detection based on weighted fuzzy-rough density, Appl. Soft Comput. 134 (2023) 109995.
[35] Z. Yuan, P. Hu, H. Chen, Y. Chen, Q. Li, Dfno: detecting fuzzy neighborhood outliers, IEEE Transactions on Knowledge and Data Engineering 37 (1) (2025) 200–209.
[36] X. Li, J. Lv, Z. Yi, An efficient representation-based method for boundary point and outlier detection, IEEE Trans. Neural Netw. Learn. Syst. 29 (1) (2016) 51–62.
[37] C. You, D.P. Robinson, R. Vidal, Provable self-representation based outlier detection in a union of subspaces, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3395–3404.
[38] H. Xu, Y. Wang, Y. Wang, Z. Wu, Mix: A joint learning framework for detecting both clustered and scattered outliers in mixed-type data, in: 2019 IEEE International Conference on Data Mining, ICDM, IEEE, 2019, pp. 1408–1413.
[39] X. Li, J. Lv, Z. Yi, Outlier detection using structural scores in a high-dimensional space, IEEE Trans. Cybern. 50 (5) (2018) 2302–2310.

[40] Z. Li, Y. Zhao, N. Botta, C. Ionescu, X. Hu, COPOD: copula-based outlier detection, in: 2020 IEEE International Conference on Data Mining, ICDM, IEEE, 2020, pp. 1118–1123.
[41] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, G.H. Chen, Ecod: Unsupervised outlier detection using empirical cumulative distribution functions, IEEE Trans. Knowl. Data Eng. 35 (12) (2022) 12181–12193.
[42] R. Li, H. Chen, S. Liu, X. Li, Y. Li, B. Wang, Incomplete mixed data-driven outlier detection based on local–global neighborhood information, Inform. Sci. 633 (2023) 204–225.
[43] H. Liu, E. Li, X. Liu, K. Su, S. Zhang, Anomaly detection with kernel preserving embedding, ACM Trans. Knowl. Discov. Data (TKDD) 15 (5) (2021) 1–18.
[44] H. Huang, Y. Wang, Q. Hu, M.-M. Cheng, Class-specific semantic reconstruction for open set recognition, IEEE Trans. Pattern Anal. Mach. Intell. 45 (4) (2022) 4214–4228.
[45] Q. Hu, L. Ji, Y. Wang, S. Zhao, Z. Lin, Uncertainty-driven active developmental learning, Pattern Recognit. 151 (2024) 110384.

Sihan Wang is currently pursuing the M.Sc. degree with the College of Computer Science, Sichuan University, Chengdu, China. Her research interests include granular computing, granular-ball computing, and outlier detection.

Zhong Yuan received the MS degree in mathematics from Sichuan Normal University, Chengdu, China, in 2018. He received the Ph.D. degree from the Southwest Jiaotong University, Chengdu, China, in 2022. He is currently a Distinguished Associate Researcher at the College of Computer Science, Sichuan University. His research interests include uncertainty artificial intelligence, granular computing, and outlier detection.

Shitong Cheng is currently pursuing the M.Sc. degree with the College of Computer Science, Sichuan University, Chengdu, China. His research interests include granular computing, anomaly detection, and outlier detection.

Hongmei Chen received the M.Sc. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2000. She received the PhD degree from the Southwest Jiaotong University, Chengdu, China, in 2013. She is currently a professor at the School of Information Science Technology, Southwest Jiaotong University. Her research interests include the areas of data mining, pattern recognition, fuzzy sets, and rough sets.

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