It is hoped that this paper will be recommended by the conference to International Journal of Machine Learning and Cybernetics.

Granular-Ball Computing-Based Markov Random Walk for Anomaly Detection

Sihan Wang, Shitong Cheng, Zhong Yuan, Hongmei Chen, Dezhong Peng

***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 important 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 local distribution properties of the samples are inscribed. Secondly, the granular-balls are used to fit the samples for constructing a state transfer matrix in the Markov 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.**

***Index Terms*—Anomaly detection, Outlier detection, granular-ball computing, multi-granularity, Markov random walk.**

I. 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 fraud detection [1], intrusion detection [2], 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-ball Computing (GBC) theory is a novel research branch in granular computing theory. It mainly generates several GBs to represent the distribution information of data, which has the advantages of robustness, multi-granularity and interpretability. Multi-Granularity is the simultaneous consideration of several different levels of granularity in data processing or analysis. Granularity refers to the degree or level of detail at which data is divided or represented. By introducing multi-granularity analysis, it can help researchers to process and analyze data at different levels simultaneously, thus obtaining more comprehensive and detailed information in data to detect anomalies more accurately. It is now widely used in machine learning tasks such as classification [4], [5] and clustering [6], [7], and effectively improves the performance of models in these tasks. However, the application of GBC to anomaly detection tasks has not been the subject of any studies to date.

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 generality allows it to be applied to a wide variety of different types of problems, which has led to its widespread interest and use [8]–[10]. Markov Random Walk (MRW) is a special case of RW that has the Markov property. It performs relatively well in anomaly detection [11], [12].

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 Markov random walk for Anomaly Detection (GBMAD) is proposed in this paper. The framework of the GBMAD method is illustrated in Fig. 1.

The contributions of this paper are listed as follows.

1) This paper introduces GBC, which provides an innovative and efficient processing tool for the field of anomaly detection;
2) GBs are used to fit samples for constructing the state transfer matrix of MRW, which effectively mines local features of data from the perspective of multi-granularity and reduces computing cost;
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 II reviews the existing anomaly detection methods, GB computing and RW theory, respectively. Section III provides a detailed description of our proposed algorithm GBMAD. Section IV compares and analyzes GBMAD experimentally with various existing algorithms. We close with concluding remarks in Section V.

This work was supported by the National Natural Science Foundation of China (62306196, 62372315, and 62376230), Sichuan Science and Technology Program (2023YFQ0020), and the Fundamental Research Funds for the Central Universities (YJ202245). (. (Corresponding author: Zhong Yuan)

S. H. Wang, S. T. Tong, Z. Yuan, and P. D. Zhong are with the College of Computer Science, Sichuan University, Chengdu 610065, China (E-mail: wangsihan0713@foxmail.com, cst010423@163.com, yuanzhong@scu.edu.cn, pengdz@scu.edu.cn

H. M. Chen is with the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China (E-mail: hmchen@swjtu.edu.cn).

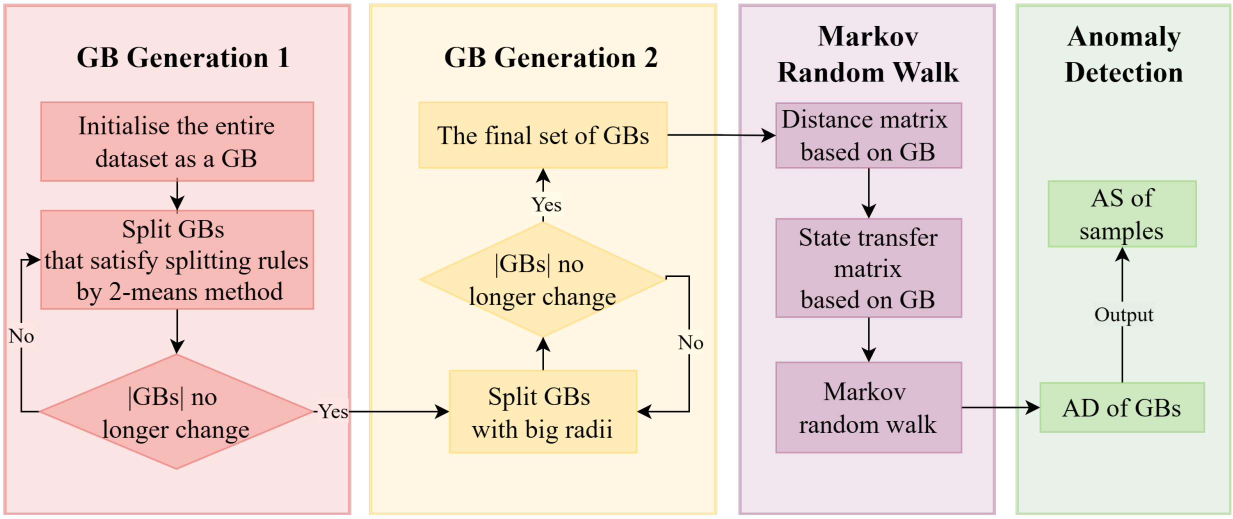


Fig. 1: The framework of GBMAD

II. Related Work

A. 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. It is 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) [13], [14] and Regression models [15], [16]. In contrast, models constructed by non-parametric methods do not have any assumption of prior knowledge for datasets. For examle, 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), as one of the basic algorithms in distance-based methods, has been proposed with many related anomaly detection methods [19], [20]. However, there are also researchers who have proposed Reverse K-Nearest Neighbors (RKNN) anomaly detection methods [21], [22]. RKNN methods usually assume that anomalies occupy only a relatively small proportion of 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 examples, Jin et al. [23] 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 areas of lower data density as anomalies. Huang et al. [24] 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 object location. Li et al. [25] 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. And they assume that the majority of 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 [26], removing anomalies for clustering [27], [28], and identifying anomalies through clustering [29]–[31].

Deep learning-based (DL) methods construct deep neural networks to learn complex feature representations of data. It is also assumed that normal results are consistent with this feature distribution. Therefore, anomalies are detected by comparing the differences between the input data and the model predictions. DL methods are generally categorised into two main types: Generative Adversarial Networks (GAN) and Auto Encoder(AE). The GAN method consists of two parts, the generator and the discriminator. It achieves modeling of the input data distribution through adversarial training between the generator and the discriminator. And it performs anomaly detection by comparing the distributional differences between the generated synthetic data and real data. Li et al. [32] proposed an unsupervised multivariate anomaly detection method based on GANs, which fully exploits both the discrimination and reconstruction to get anomaly scores for samples. Whereas, AE methods generally use the values of residual vectors to measure anomalies by observing the reconstruction error in each feature space. An et al. [33] proposed an anomaly detection method based on the reconstruction probability of a variational autoencoder. It takes the reconstruction probability as the anomaly score of the sample, which is principled and objective.

B. Granular-ball computing

Due to the advantages of natural interpretability, multi-granularity, and robustness [34], GB computing is now widely integrated into various classical computational models, such as GB rough set [35], GB classification [4], [5], and GB clustering [6], [7], [36]. Xia et al. [35] proposed a GB Rough Set (GBRS) model by combining the robustness and adaptability of GB computing. It can be transformed into GB Pawlak Rough Set (GBPRS) and GB Neighborhood Rough Set (GBNRS). GBRS not only can represent the knowledge in terms of equivalence classes, but also can deal with both continuous and discrete data, which can make up for the existing defects of PRS and NRS. Xia et al. [4] combined GB with model KNN and Support Vector Machine (SVM) to propose more efficient classifiers GBKNN and GBSVM. Xie et al. [6] 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. And GBC can portray the local features of the samples from a multi-granularity perspective, which is helpful for the task of detecting anomalies. Therefore, GB computing has the potential to be integrated with anomaly detection models, resulting in improved model performance.

C. Random walk

RW is now widely used in various tasks such as segmentation [8], [37], classification [9], clustering [38], etc. Dong et al. [8] proposed a new sub MRW algorithm with labeled prior, which outperforms traditional RW algorithms in seed image segmentation. Jiang et al. [37] 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. [9] 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. [39] 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 certain probability, which is effective in classifying different types of data. Hua et al. [38] 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. [12] proposed a random graph-based anomaly detection method. It uses the similarity between objects to construct an adjacency matrix for graph representation, and constructs a Markov model for RW so that each object is assigned an anomaly score. Liu et al. [11] proposed a fuzzy granular anomaly detection method using RW, which makes up for the inadequacy of existing anomaly detection techniques 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 Markov random walk for Anomaly Detection.

III. GBC-based Markov random walk for anomaly detection

A. Granular-ball generation

Let a three-tuple $IS =< X, C, f >$ denote a data table without decision information, where $X = \{x_1, x_2, \cdots, x_n\}$ denotes the set of objects and $C = \{c_1, c_2, \cdots, c_b\}$ denotes the set of conditional attributes. For any $x \in X$ and $c \in C$, $f_c(x)$ denotes the value of the object x under the attribute c.

The main idea of GB computing is to generate GBs to cover or partially cover the sample space [4].

Definition 1: For each GB_k, it contains two items of information: c_k and r_k denote the center and radius of GB_k, respectively. They can be calculated as follows.

$$c_k = \frac{1}{n_k}\sum_{l=1}^{n_k} x_l, \tag{1}$$

$$r_k = \frac{1}{n_k}\sum_{l=1}^{n_k} ||x_l - c_k||, \tag{2}$$

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

A brief description of the process of generating GBs is shown in Fig 2. It can be divided into the following four main steps.

Step 1: Initialise the entire dataset as a GB.

Step 2: Perform 2-means clustering algorithm on that GB in order to split it into two sub-GBs that can cover all the samples.

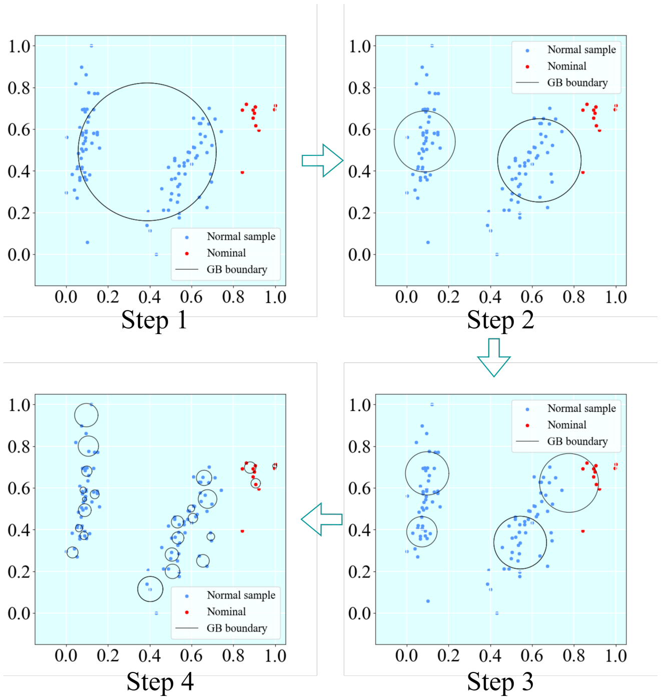


Fig. 2: A brief description of the process of generating GBs under the Iris dataset reduced to 2 dimensions. Red dots indicate anomalies and blue dots indicate normal samples.

Step 3: Set the splitting criteria. If a GB meets the splitting criteria, it proceeds with the split by 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 [5]. Purity refers to the proportion of objects belonging to the label which has the highest number of samples in a GB. It can be expressed as $\text{Purity}(GB) = \max_{i\in 1,2,\ldots,k}\left(\frac{|Label_i|}{|GB|}\right)$, where $|Label_i|$ denotes the number of samples belonging to the i-th label in the GB, and $|GB|$ denotes the sample number in the GB. Obviously, the variable Purity requires labelling information. Therefore, the splitting criterion is not suitable in unsupervised tasks. Xie et al. [6] proposed a novel criterion for splitting GB, i.e., satisfying the condition that the weighted sub-ball is of better quality than the parent ball will perform the GB split. 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

$$SD_k = \sum_{l=1}^{n_k} ||x_l - c_k||. \tag{3}$$

Clearly, SD describes the sum of the distances from all samples to the centre in the GB. It is not only related to the distance between samples to the centre, 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 centre or the GB contains fewer samples.

Definition 3: Let SD_k be SD of the original GB. SD_{k_1} and SD_{k_2} denote SDs of the sub-GBs from SD_k by 2-means. 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 if a GB splits into two sub-GBs, the sum of the sample numbers in the sub-GBs will remain equal to the sample numbers within the original GB. Thus, the most significant difference between the sum of SD of the sub-GBs and SD of the original GB is the concentration of the samples in the GB towards the centre. Then, we can take advantage of this trait to construct the following splitting criterion.

Definition 4: The splitting criterion is

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

According to Definition 4, if SD^k_{childs} is smaller than SD_k, it suggests that the sum distances from samples to the centre decreases and that there is a greater concentration of samples inside the sub-GBs. Therefore, GB_k needs to be splited 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 GBs with fewer samples than the predefined constant L can no longer be split (set to 8 in our experiment). And we give the experimental basis for choosing 8 as the minimum sample number of GB in Section V.

Moreover, considering the problem of failing to segment some GBs with too large radii due to noise effects, Xie et al. [6] set an additional criterion: if $r_k > 2 \times \max(mean(r), median(r))$, i.e., then GB_k needs to be splitted, where $mean(r)$ and $median(r)$ refer to the mean and median of all GB radii, respectively. Until all GBs do not satisfy the splitting criterions, the final set of GBs ($GBs = \{GB_1, GB_2, \cdots, GB_m\}$) is obtained, where m denotes the number of GBs.

B. Markov state transfer matrix based on GB

MKW 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 ($t = 0, 1, \cdots, T$) and state space ($S = \{S_1, S_2, \cdots, S_l\}$). The conditional transfer probability can be expressed as:

$$p(S^{(t+1)} = S_j | S^{(t)} = S_i, S^{(t-1)}, \cdots, S^{(0)}) = p(i,j), \tag{6}$$

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

Proposition 1: The state transfer probability matrix P holds

1) For any i, j, $P_{ij} \geq 0$, means that the probability is non-negative.
2) For any i, $\sum_{S_j \in S} P_{ij} = 1$, indicates that the sum of probabilities from state S_i to other states is 1.

Liu et al. [11] used the fuzzy granular distance between samples to define the state transfer probability matrix for MRW process. The larger relative distance between sample x_i and x_j is, the larger the corresponding P_{ij}. According to this trait, anomalies tend to have a higher probability of being visited. The state transfer probability matrix is as follows.

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1} & P_{n2} & \cdots & P_{nn} \end{bmatrix} \tag{7}$$

Based on the GBs generated in the previous subsection, we reconstruct the state transfer matrix using GBs instead of samples.

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

$$dis(GB_i, GB_j) = ||c_i - c_j|| + r_i + r_j. \tag{8}$$

Equation 8 can be expressed as a distance matrix $Dis = [dis(GB_i, GB_j)]_{m \times m}$. 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.

To satisfy Proposition 1, we row-normalize each row of Dis by dividing the values of each row by the sum of the values of that row.

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

$$P = \begin{bmatrix} \frac{Dis(1,1)}{\sum_{k=1}^{m} Dis(1,k)} & \cdots & \frac{Dis(1,m)}{\sum_{k=1}^{m} Dis(1,k)} \\ \vdots & \ddots & \vdots \\ \frac{Dis(m,1)}{\sum_{k=1}^{m} Dis(m,k)} & \cdots & \frac{Dis(m,m)}{\sum_{k=1}^{m} Dis(m,k)} \end{bmatrix}. \tag{9}$$

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, anomalies in GBs tend to have a higher probability of being visited, which is reasonable.

C. Markov random walk 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 the GB an initial probability value ($S^{(0)}(k) = \frac{1}{m}, 1 \leq k \leq m$), respectively.

Definition 7: According to the state transfer matrix P, the process of MRW for the state probability distribution is defined as

$$S^{(t+1)} = d + (1-d)S^{(t)}P, \tag{10}$$

where $S^{(t+1)} = [S^{(t+1)}(1), S^{(t+1)}(2), \cdots, S^{(t+1)}(m)]$ denotes the probability distribution vector at the $(t+1)$-th iteration. Similarly, $S^{(t)}$ denotes the probability distribution vector at the t-th iteration. d is a damping factor indicating that the walker visits one of the neighboring GBs with probability $(1-d)$ or requests another random state with probability d.

When the probability distribution S tends to a steady-state distribution, each value in it can represent the probability that a random walker stays at the corresponding GB, i.e., the Anomaly Degree (AD) of each GB. The larger the AD for each GB, the greater the probability that the samples it contains are anomalous.

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

$$AD(GB_k) = S(k), \tag{11}$$

where $S(k)$ denotes the value of the k-th value in the steady-state distribution vector 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 $x_i \in X$, if x_i is covered by GB_k, then the anomaly score of x_i can be calculated by

$$AS(x_i) = AD(GB_k) \times W(GB_k), \tag{12}$$

where $W(GB_k): U \to [0,1]$ is a weight coefficient computed by $1 - \sqrt[3]{\frac{|n_k|}{|X|}}$., 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 of the other samples, which makes them more likely to be anomalies in the dataset. Therefore, it is reasonable that we will assign a greater weight to that GB.

Finally, the following rules are set to determine whether the sample is anomalous or not.

Definition 10: Given a threshold ϵ, for any $x_i \in X$, the decision rules of anomaly detection are presented as

$$Rules: \begin{cases} \text{IF } AS(x_i) > \epsilon, & \text{THEN } x_i \text{ is anomalous;} \\ \text{IF } AS(x_i) \leq \epsilon, & \text{THEN } x_i \text{ is normal;} \end{cases} \tag{13}$$

D. Anomaly detection algorithm

This subsection provides a pseudo-code for the algorithm GBMAD and analyzes the time complexity of the algorithm.

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

Stage 1: First, set the anomaly score set AS to $\emptyset$ and the whole dataset 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 granules 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 12-21), which only re-split some of the GBs with larger radii, also with a time complexity of $O(bn)$.

Step 3: The obtained GB_{new} is used as the final set GBs in Step 22. In Step 23, 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 25-28) for RW until a steady-state distribution is obtained. At Steps 29-34, the AD is calculated for each GB and the anomaly score is calculated for each sample. The computational complexity is $O(n)$.

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

Algorithm 1: GBMAD Algorithm

```
    Input: $IS =< X, C, f >, d, L$.
    Output: Anomaly Score set($AS$).
 1  $AS \leftarrow \emptyset, GB_{\text{new}} \leftarrow X$;
 2  repeat
 3      $GB_{\text{last}} \leftarrow GB_{\text{new}}$;
 4      for each $GB_k \in GB_{new}$ do
 5          if $|GB_k| \leq L$ then
 6              Continue;
 7          end
 8          Calculate $SD_k$, $SD^k_{childs}$;
 9          if $SD_j \leq SD^k_{childs}$ then
10              Split $SD_k$ into $SD_{k_1}, SD_{k_2}$;
11              Add $SD_{k_1}, SD_{k_2}$, delete $GB_k$ to $GB_{\text{new}}$;
12          end
13      end
14  until $(|GB_{new}| = |GB_{last}|)$;
15  repeat
16      $GB_{\text{last}} \leftarrow GB_{\text{new}}$;
17      Calculate $mean(r), median(r)$;
18      for each $GB_k \in GB_{new}$ do
19          if $r_k \geq 2 \times \max(mean(r), median(r))$ then
20              Split $SD_k$ into $SD_{k_1}, SD_{k_2}$;
21              Add $SD_{k_1}, SD_{k_2}$, delete $GB_k$ to $GB_{\text{new}}$;
22          end
23      end
24  until $(|GB_{new}| = |GB_{last}|)$;
25  $GBs \leftarrow GB_{\text{new}}$;
26  Construct $Dis$ and P by Eq. (8)-(9);
27  Intialize $S^{(0)} = [\frac{1}{m}, \frac{1}{m}, \cdots, \frac{1}{m}]$;
28  repeat
29      $S^{(t+1)} = d + (1-d)S^{(t)}P$;
30      $t \leftarrow t + 1$;
31  until $(\|S^{(t+1)} - S^{(t)}\| \leq 10^{-3})$;
32  for each $GB_k \in GBs$ do
33      Calculate $AD(GB_k)$ by Eq. (11)
34  end
35  for each $x_i \in X$ do
36      Calculate $AS(x_i)$ by Eq. (12)
37  end
38  return $AS$.
```

IV. Experiments and Analyses

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

A. Datasets

We collected a total of 16 datasets on the public web pages[1],[2]. The datasets on the pages are usually used to evaluate the performance of anomaly detection methods [40]–[42], and are therefore suitable for use in this experiment.

In these datasets, the sample number extend beyond 101 to 5803 and the attribute number reach within the range of 5 to 100. Table I shows specific information about the experimental dataset.

B. Comparison algorithms

In total, we employed the following 11 comparative methods to conduct experiments for evaluating the performance of our proposed method.

1) INFLO (2006) [23]: An local outlier detection method based on symmetric neighborhood relations, which considers both the neighborhood and the reverse neighborhood of the target in estimating the density distribution.
2) OutRanka (2008) [43] :A random graph-based data outlier detection algorithm, which considers building a data graph representation based on object similarity.
3) LDOF (2009) [44] : A local distance-based outlier detection method, which determines the degree for an object has deviated from its neighbors by the relative position of an object to its neighbors.
4) LoOP (2009) [45]: A local density-based outlier detection method, which provides each sample with an outlier probability as its anomaly score.
5) NOF (2016) [24]: A natural neighborhood-based algorithm without any parameters, which proposes the concept of natural outlier factor to measure the outliers.
6) NC (2016) [46]: A representation-based method that can be used to identify both outliers and boundary points.
7) COPOD (2020) [47]: An outlier detection method based on empirical copula, which predicts the tail probability of each sample to determine its degree of extremity.
8) VarE (2020) [48]: An anomaly detection method on the basis of structure scores, which are derived from measuring the angular variance weighted by data representation.
9) DCROD (2022) [25]: An anomaly detection method based on the rate of change of directional density ratio.
10) ECOD (2022) [49]: An empirical-cumulative-distribution-based outlier detection method, which calculates the outlier value for each sample by aggregating the tail probabilities estimated for each dimension.
11) ILGNI (2023) [50]: A network method for incomplete local and global neighborhood information, which considers the smooth distribution of MRW models on the network and computes outliers.

C. Evaluation indicators

According to Refs [41] and [40], the evaluation metrics ROC (Receiver Operating Characteristic) curve and AUC

[1] https://github.com/BElloney/Outlier-detection

[2] http://odds.cs.stonybrook.edu

TABLE I: The specific information of datasets

ID	Datasets	Abbreviation	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	Ecoli	Ecoli	7	336	9
5	Lymphography	Lymphography	18	148	6
6	Musk	Musk	166	3062	97
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

TABLE II: Comparative results of AUC experiments

Datasets	INFLO	OutRanka	LDOF	LoOP	NOF	NC	COPOD	VarE	DCROD	ECOD	ILGNI	GBMAD
Annealing	0.7055	0.8059	0.6270	0.6230	0.7750	0.6526	0.7967	0.6195	0.7552	0.7874	0.6864	**0.8310**
Arrhythmia	0.7820	0.8098	0.7493	0.7708	0.8160	0.7337	0.8046	0.7385	0.7961	0.8071	0.8164	**0.8341**
Breast	0.6445	0.5313	0.6539	0.6159	0.6761	0.6420	0.6322	0.6369	0.6035	0.6555	0.6587	**0.6941**
Ecoli	0.8624	0.8790	0.8627	0.8858	0.9178	0.8814	0.8087	0.8987	0.8746	0.7808	0.6381	**0.9186**
Lymphography	0.9941	0.7934	0.9918	0.9930	0.9765	0.9824	0.9941	0.5070	0.9542	**0.9965**	0.9906	0.9554
Musk	0.5337	0.9557	0.6631	0.5573	0.9638	0.5719	0.9463	0.3457	0.8633	0.9559	0.9930	**1.0000**
Pima	0.8940	0.7448	0.8555	0.8604	0.9416	0.8882	0.9423	0.9040	0.9278	0.9472	0.9007	**0.9481**
Satimage	0.6603	0.9954	0.6487	0.6154	**0.9988**	0.5896	0.9745	0.5371	0.8649	0.9649	0.9941	0.9986
Sonar	0.9835	0.9021	0.9639	0.9814	0.9845	**0.9985**	0.9856	0.9546	0.9887	0.9660	0.9649	0.9979
Waveform	0.7034	0.6692	0.7112	0.7161	0.8265	0.6957	0.7343	0.4730	0.7433	0.6084	0.7052	**0.8898**
Wbc	0.8746	0.6410	0.7922	0.6447	0.9966	0.7559	0.9955	**0.9973**	0.9764	0.9955	0.9880	**0.9973**
Wdbc	0.9537	0.7646	0.7019	0.8393	0.9949	0.8343	0.9956	0.9973	0.9682	0.9591	0.9512	**1.0000**
Wine	0.8504	0.3034	0.8160	0.8555	0.9471	0.9130	0.8672	0.9412	0.8849	0.7328	0.7445	**1.0000**
Wpbc	0.4947	0.4829	0.5037	0.5051	0.5125	0.5099	0.5226	0.5516	0.5129	0.4810	0.4210	**0.5967**
Yeast	0.9921	0.9896	0.9873	0.9877	**1.0000**	0.9697	0.9974	**1.0000**	0.9898	0.9952	0.9810	0.9988
Zoo	0.6625	0.6043	0.7038	0.6982	0.7010	0.7234	0.5497	0.6828	0.7521	0.5819	0.5889	**0.7966**
Average	0.7870	0.7420	0.7645	0.7593	0.8768	0.7714	0.8467	0.7366	0.8410	0.8260	0.8139	**0.9036**

(Area Under Curve) are generally used for outlier detection performance evaluation. 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 $AS(k)$. And AS_{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.

$$FPR(k) = \frac{|AS(k) - AS_{\text{true}}|}{|X_{\text{normal}}|} \times 100\%; \tag{14}$$

$$TPR(k) = \frac{|AS(k) \cap AS_{\text{true}}|}{|AS_{\text{true}}|} \times 100\%, \tag{15}$$

where FPR denotes the proportion of all samples that are actually normal samples that are incorrectly predicted to be anomalies, and TPR denotes the proportion of all samples that are actually anomalies that are correctly predicted to be anomalies.

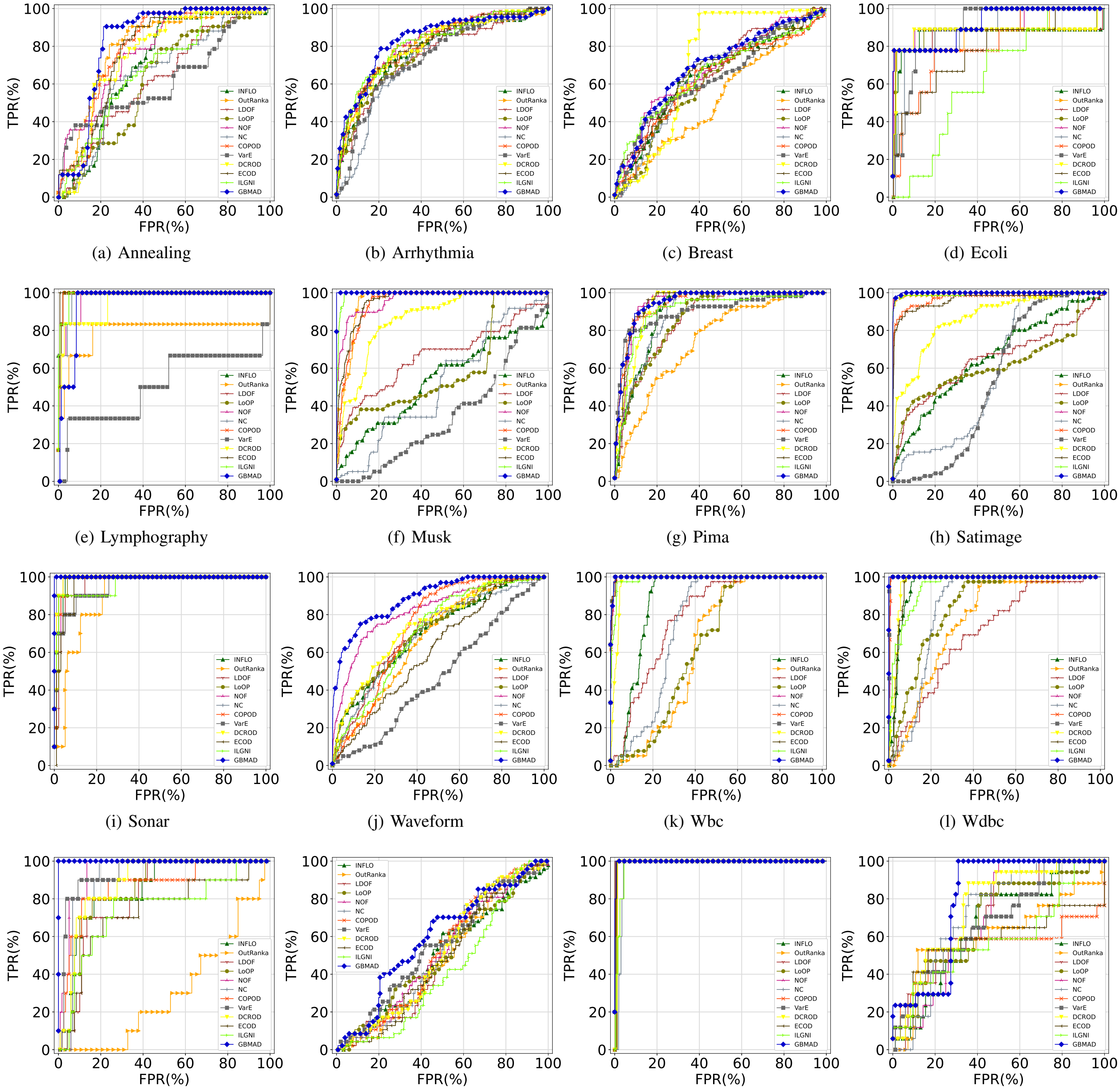
However, if multiple ROC curves on a dataset are intertwined, it can be difficult when evaluating the performance of the corresponding algorithm. For this purpose, we use the AUC metric for more intuitive evaluation. AUC refers to the area under the ROC curve, and its value ranges from 0 to 1. The closer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

$$\text{AUC} = \underset{x_i \in AS_{\text{true}}, x_j \in X_{\text{normal}}}{\text{Mean}} \begin{cases} 1, & score(x_i) > score(x_j) \\ 0.5, & score(x_i) = score(x_j) \\ 0, & score(x_i) < score(x_j) \end{cases},$$

where $score(x_i)$ denotes the anomaly score of the sample x_i.

D. 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 ROC of the proposed algorithm GBMAD. It can be noticed that in most of the datasets (e.g., Musk, Satimage, Sonar, Waveform, Wbc, Wdbc, Wine, Wpbc, and Yeast), the dark blue curve is clearly closest to the upper left corner compared to the other curves. In Annealing, Breast, Ecoli, and Pima, the GBMAD curve partially overlaps with those of the other



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

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

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

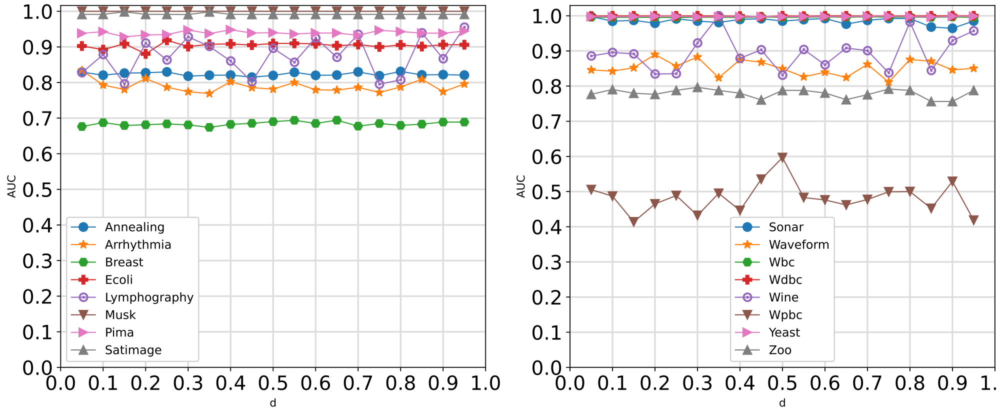
(m) Wine (n) Wpbc (o) Yeast (p) Zoo

Fig. 3: Experimental comparison results on ROC

algorithms. And it is not possible to visually analyze which algorithm performs best, so AUC metrics are needed to assess comparative performance. The dark blue curve of GBMAD in Lymphography, is not closest to the upper left corner, slightly worse than algorithms such as COPOD, ECOD and ILGNI, etc.

Table II shows the results of the experimental comparison of the AUCs, where the bold number in each row representing the algorithm that performed best on the corresponding dataset, as well as the underlined numbers represent the next best algorithm in performance in that row. It can be found that the number of best AUC values achieved by algorithms INFLO, OutRanka, LDOF, LoOP, NOF, NC, COPOD, VarE, DCROD, ECOD, ILGNI, and GBMAD on these 16 datasets are 0, 0, 0, 0, 2, 1, 0, 2, 0, 1, 0, and 12 respectively. In addition to this, GBMAD also has two 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 0.7870, 0.7420, 0.7645, 0.7593, 0.8768, 0.7714, 0.8467, 0.7366, 0.8410, 0.8260, 0.8139, and 0.9036, respectively.

Among them, GBMAD provides the best results, which is



(a) Datasets 1-8

(b) Datasets 9-16

Fig. 4: Variation curve of AUC with d

much higher than the other methods. Therefore, it can be concluded that the detection performance of GBMAD is better than the other 11 algorithms on most of the datasets.

E. Parameter sensitivity analysis

In this subsection, we analyze the sensitivity of the algorithm GBMAD to the parameter d. Figs. 4(a)-(b) plot the curves of AUC with respect to the parameter d for the first 8 and the last 8 datasets, respectively. d ranges from 0.05 to 0.95 in steps of 0.05. As can be seen from Fig. 4, most of the datasets have small fluctuations in the AUC curves with respect to d. It indicates that the detection model is less sensitive to the parameter d. However, the AUC curve in the dataset Wdbc fluctuates more and does not have a certain trend. The reason for this may be that the distribution of data in Wdbc is not significant, leading to large differences in the results of each 2-means.

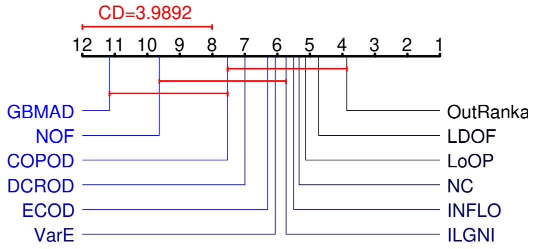


Fig. 5: Nemenyi's test figure on AUC

F. Statistical test

In this subsection, we perform an statistical test to explore the statistical difference of each algorithm [40], [51]. For each dataset, the AUC values of all the algorithms are to be sorted in ascending order and assigned ordinal numbers $1, 2, \cdots$. 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 $\tau_F = 7.0766$, which is much larger than the critical value of 1.6129. 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 $CD = 3.9892$ when the significance level is 0.1.The mean ordinal values obtained by each algorithm are plotted on the numerical axis of the Nemenyi test plot, respectively. If several of the algorithms are connected by a red horizontal line, 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 significantly different. Fig. 5 demonstrates that the algorithm GBMAD has no connection with DCROD, VarE, ECOD, NC, INFLO, ILGNI, LoOP, LDOF, and OutRanka, suggesting that GBMAD is significantly different from these algorithms. Whereas, GBMAD is connected with NOF and COPOD, indicating that it is not possible to prove that GBMAD is statistically different from these algorithms.

G. Effect of setting the GB's minimum sample number L

As we mentioned on the criterion setting in Section III, GBs with fewer than 8 samples will no longer split. This is a consideration for the time-consuming nature of 2-means. In order to analyze the reasonableness of this setting, we took the GB's minimum sample number L from 3 to 20, with a step size of 1. The mean AUC values obtained for all experimental datasets at each setting are calculated, as well as the average time consumption. They are presented in Table III. Overall, the average AUC generally remains in the interval

TABLE III: Comparative results of Minimum Number setting

L	3	4	5	6	7	8	9	10	11
AUC	0.8964	0.8962	0.8968	0.9030	0.9014	0.9036	0.9041	0.8999	0.8991
Time (s)	507.21	311.35	199.57	199.84	155.87	165.63	111.52	82.25	98.31
L	12	13	14	15	16	17	18	19	20
AUC	0.9039	0.9060	0.9040	0.9003	0.8931	0.8992	0.8889	0.8977	0.8974
Time (s)	91.69	100.13	77.05	90.83	66.01	74.47	60.88	62.47	57.53

$[0.8890, 0.9040]$, indicating that the model is not sensitive to the choice of this parameter. When L is set small such as $3-5$, its AUC value decreases by about 0.01. And the average time consumption is higher, for example, when L is set to 3, the average time consumption is as high as 500s or more. When L is set to $6-14$, the AUC value can be roughly maintained at about 0.90, and the time consumption becomes less. When L is set too large, e.g. $16-20$, the time consumption is more reduced, but the AUC value also decreases slightly. It may be due to the fact that some of the larger GBs are not able to continue splitting and thus the detection performance is reduced. Therefore, L can be set among $6-14$, or adjusted according to the actual size of the dataset.

In our experiment, L is set to 8, at which time the average time consumption is 165s, which greatly reduces the time overhead and improves the AUC value by a certain amount, so it is reasonable.

V. Conclusion

This paper proposed an anomaly detection method based on GBC using the tool of Markov random walk. Through the multi-granularity of GBs, the method effectively captured the distributional characteristics of data, providing a more comprehensive analysis of anomalies. The use of adaptive GBs and the construction of a novel state transfer matrix based on GBs for MRW improve 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, showcasing its feasibility and practical applicability.

Overall, this research contributes to the field of anomaly detection by introducing a novel method that combines GBC theory. In future work, we will focus on further optimizing the computational efficiency of the granular-ball generation and exploring its applications in real-world scenarios.

References

[1] K. G. Al-Hashedi and P. Magalingam, "Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019," *Computer Science Review*, vol. 40, p. 100402, 2021.

[2] J. Xie, H. Wang, J. M. Garibaldi, and D. Wu, "Network intrusion detection based on dynamic intuitionistic fuzzy sets," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 9, pp. 3460–3472, 2022.

[3] C. Baur, S. Denner, B. Wiestler, N. Navab, and S. Albarqouni, "Autoencoders for unsupervised anomaly segmentation in brain mr images: a comparative study," *Medical Image Analysis*, vol. 69, p. 101952, 2021.

[4] S. Xia, Y. Liu, X. Ding, G. Wang, H. Yu, and Y. Luo, "Granular ball computing classifiers for efficient, scalable and robust learning," *Information Sciences*, vol. 483, pp. 136–152, 2019.

[5] S. Xia, X. Dai, G. Wang, X. Gao, and E. Giem, "An efficient and adaptive granular-ball generation method in classification problem," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[6] J. Xie, W. Kong, S. Xia, G. Wang, and X. Gao, "An efficient spectral clustering algorithm based on granular-ball," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[7] S. Xia, D. Peng, D. Meng, C. Zhang, G. Wang, E. Giem, W. Wei, and Z. Chen, "Ball k k-means: Fast adaptive clustering with no bounds," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 87–99, 2020.

[8] X. Dong, J. Shen, L. Shao, and L. Van Gool, "Sub-markov random walk for image segmentation," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 516–527, 2015.

[9] X. Zhao, R. Tao, W. Li, H.-C. Li, Q. Du, W. Liao, and W. Philips, "Joint classification of hyperspectral and lidar data using hierarchical random walk and deep cnn architecture," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 10, pp. 7355–7370, 2020.

[10] R. Liu, Z. Lin, and Z. Su, "Learning markov random walks for robust subspace clustering and estimation," *Neural Networks*, vol. 59, pp. 1–15, 2014.

[11] C. Liu, Z. Yuan, B. Chen, H. Chen, and D. Peng, "Fuzzy granular anomaly detection using markov random walk," *Information Sciences*, vol. 646, p. 119400, 2023.

[12] H. Moonesinghe and P.-N. Tan, "Outlier detection using random walks," in *2006 18th IEEE international conference on tools with artificial intelligence (ICTAI'06)*, pp. 532–539, IEEE, 2006.

[13] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International conference on learning representations*, 2018.

[14] J. Qu, Q. Du, Y. Li, L. Tian, and H. Xia, "Anomaly detection in hyperspectral imagery based on gaussian mixture model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 11, pp. 9504–9517, 2020.

[15] O. Salem, A. Guerassimov, A. Mehaoua, A. Marcus, and B. Furht, "Anomaly detection in medical wireless sensor networks using svm and linear regression models," *International Journal of E-Health and Medical Communications (IJEHMC)*, vol. 5, no. 1, pp. 20–45, 2014.

[16] 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 Transactions on Knowledge and Data Engineering*, vol. 32, no. 2, pp. 218–233, 2018.

[17] L. J. Latecki, A. Lazarevic, and D. Pokrajac, "Outlier detection with kernel density functions," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 61–75, Springer, 2007.

[18] M. Goldstein and A. Dengel, "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm," *KI-2012: poster and demo track*, vol. 1, pp. 59–63, 2012.

[19] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 427–438, 2000.

[20] X. Wang, X. L. Wang, Y. Ma, and D. M. Wilkes, "A fast mst-inspired knn-based outlier detection method," *Information Systems*, vol. 48, pp. 89–112, 2015.

[21] M. Radovanović, A. Nanopoulos, and M. Ivanović, "Reverse nearest neighbors in unsupervised distance-based outlier detection," *IEEE transactions on knowledge and data engineering*, vol. 27, no. 5, pp. 1369–1382, 2014.

[22] A. Wahid and C. S. R. Annavarapu, "Nanod: A natural neighbour-based outlier detection algorithm," *Neural Computing and Applications*, vol. 33, pp. 2107–2123, 2021.

[23] W. Jin, A. K. Tung, J. Han, and 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*, pp. 577–593, Springer, 2006.

[24] J. Huang, Q. Zhu, L. Yang, and J. Feng, "A non-parameter outlier detection algorithm based on natural neighbor," *Knowledge-Based Systems*, vol. 92, pp. 71–77, 2016.

[25] K. Li, X. Gao, S. Fu, X. Diao, P. Ye, B. Xue, J. Yu, and Z. Huang, "Robust outlier detection based on the changing rate of directed density ratio," *Expert Systems with Applications*, vol. 207, p. 117988, 2022.

[26] H. Liu, J. Li, Y. Wu, and Y. Fu, "Clustering with outlier removal," *IEEE transactions on knowledge and data engineering*, vol. 33, no. 6, pp. 2369–2379, 2019.

[27] K. K. Sharma and A. Seal, "Outlier-robust multi-view clustering for uncertain data," *Knowledge-Based Systems*, vol. 211, p. 106567, 2021.

[28] F. Jiang, G. Liu, J. Du, and Y. Sui, "Initialization of k-modes clustering using outlier detection techniques," *Information Sciences*, vol. 332, pp. 167–183, 2016.

[29] S. A. N. Nozad, M. A. Haeri, and G. Folino, "Sdcor: Scalable density-based clustering for local outlier detection in massive-scale datasets," *Knowledge-based systems*, vol. 228, p. 107256, 2021.

[30] J. Li, H. Izakian, W. Pedrycz, and I. Jamal, "Clustering-based anomaly detection in multivariate time series data," *Applied Soft Computing*, vol. 100, p. 106919, 2021.

[31] G. Pu, L. Wang, J. Shen, and F. Dong, "A hybrid unsupervised clustering-based anomaly detection method," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 146–153, 2020.

[32] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks," in *International conference on artificial neural networks*, pp. 703–716, Springer, 2019.

[33] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.

[34] S. Xia, G. Wang, and X. Gao, "Granular ball computing: an efficient, robust, and interpretable adaptive multi-granularity representation and computation method," *arXiv preprint arXiv:2304.11171*, 2023.

[35] S. Xia, C. Wang, G. Wang, X. Gao, W. Ding, J. Yu, Y. Zhai, and 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*, 2023.

[36] 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 Transactions on Neural Networks and Learning Systems*, 2023.

[37] J. Jiang, C. Lyu, S. Liu, Y. He, and X. Hao, "Rwsnet: a semantic segmentation network based on segnet combined with random walk for remote sensing," *International Journal of Remote Sensing*, vol. 41, no. 2, pp. 487–505, 2020.

[38] J. Hua, J. Yu, and M.-S. Yang, "Fast clustering for signed graphs based on random walk gap," *Social Networks*, vol. 60, pp. 113–128, 2020.

[39] S. Wang, Z. Wang, K.-L. Lim, G. Xiao, and W. Guo, "Seeded random walk for multi-view semi-supervised classification," *Knowledge-Based Systems*, vol. 222, p. 107016, 2021.

[40] Z. Yuan, B. Chen, J. Liu, H. Chen, D. Peng, and P. Li, "Anomaly detection based on weighted fuzzy-rough density," *Applied Soft Computing*, vol. 134, p. 109995, 2023.

[41] Z. Yuan, H. Chen, C. Luo, and D. Peng, "Mfgad: Multi-fuzzy granules anomaly detection," *Information Fusion*, vol. 95, pp. 17–25, 2023.

[42] Z. Yuan, H. Chen, T. Li, X. Zhang, and B. Sang, "Multigranulation relative entropy-based mixed attribute outlier detection in neighborhood systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 8, pp. 5175–5187, 2021.

[43] H. Moonesinghe and P.-N. Tan, "Outrank: a graph-based outlier detection framework using random walk," *International Journal on Artificial Intelligence Tools*, vol. 17, no. 01, pp. 19–36, 2008.

[44] K. Zhang, M. Hutter, and H. Jin, "A new local distance-based outlier detection approach for scattered real-world data," in *Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009 Proceedings 13*, pp. 813–822, Springer, 2009.

[45] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Loop: local outlier probabilities," in *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 1649–1652, 2009.

[46] X. Li, J. Lv, and Z. Yi, "An efficient representation-based method for boundary point and outlier detection," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 1, pp. 51–62, 2016.

[47] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, "Copod: copula-based outlier detection," in *2020 IEEE international conference on data mining (ICDM)*, pp. 1118–1123, IEEE, 2020.

[48] X. Li, J. Lv, and Z. Yi, "Outlier detection using structural scores in a high-dimensional space," *IEEE transactions on cybernetics*, vol. 50, no. 5, pp. 2302–2310, 2018.

[49] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. Chen, "Ecod: Unsupervised outlier detection using empirical cumulative distribution functions," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[50] R. Li, H. Chen, S. Liu, X. Li, Y. Li, and B. Wang, "Incomplete mixed data-driven outlier detection based on local–global neighborhood information," *Information Sciences*, vol. 633, pp. 204–225, 2023.

[51] S. Wang, Z. Yuan, C. Luo, H. Chen, and D. Peng, "Exploiting fuzzy rough entropy to detect anomalies," *International Journal of Approximate Reasoning*, vol. 165, p. 109087, 2024.