Pattern Recognition 176 (2026) 113212

Contents lists available at ScienceDirect

Pattern Recognition

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

Fast and robust outlier detection: A granular-ball center isolation and region consistency approach

Rongxiang Wang[a,b], Jihong Wan[a,c,b,*], Xiaoping Li[a,b], Shuaishuai Tan[a,b]

[a] School of Computer Science and Technology, Guangdong University of Technology, 510006, Guangzhou, China
[b] Intelligent Services Scheduling and Security Laboratory, Guangdong University of Technology, 510006, Guangzhou, China
[c] Department of Computing, Hong Kong Polytechnic University, Hong Kong SAR, 999077, China

ARTICLE INFO

Keywords:
Outlier detection
Granular-ball computing
k-nearest neighbor
Isolation

ABSTRACT

Outlier detection is an essential task in data mining, focused on identifying abnormal objects that deviate from normal distribution. The *k*-nearest neighbors-based detection method is one of the widely used techniques. However, as data scale increases, the process of finding *k*-nearest neighbors for each object becomes extremely time-consuming. Additionally, if neighbors of objects contain noise, it may interfere with computation of its relationships with neighbors, which affects detection performance. To address these issues, this paper proposes a fast and robust outlier detection method based on granular-ball (GB) center isolation and region consistency, called FROD. Specifically, generation of GBs is the first step. The dataset is covered by generating GBs with different granularities. Then, by calculating the GB center isolation ($GBCI$), it evaluates the isolation degree of different GB centers relative to other GB centers. From a global perspective, $GBCI$ indirectly reflects the position and isolation of each GB center within the overall data distribution. Furthermore, by calculating the GB center region consistency ($GBCRC$) of an object, it measures closeness between object and GB center neighborhood. From a local perspective, $GBCRC$ reflects the correlation between the object and the data distribution within the GB center neighborhood to which it belongs. Finally, by combining $GBCI$ and $GBCRC$, outlier factor of each object is obtained, and a corresponding detection algorithm is designed. Experimental results show that FROD performs excellently in terms of detection efficiency and accuracy, and demonstrates robustness in noisy environments.

1. Introduction

Outlier detection, as a unique and important research task in data mining, aims to identify those outliers in the data that are significantly different from the majority of the data [1]. In most data mining tasks, outliers are often regarded as noise and discarded [2]. However, in some applications, outliers from minority classes are often more valuable than normal data points. For instance, applications such as fraud detection in finance [3], intrusion detection in cybersecurity [4], fault diagnosis in manufacturing [5], and anomaly detection in medical data [6], etc.

Distance-based [7] and density-based [8] outlier detection methods are currently the most commonly used techniques for outlier detection. These methods often rely on the concept of *k*-nearest neighbors (*k*-NNs) [9]. Distance-based methods evaluate whether an object is an outlier by calculating the distance relationship between each object and its *k*-NNs. Density-based methods assess the region consistency within the neighborhood of an object and typically consider objects in low-density regions as outliers [10]. Although *k*-NNs-based outlier detection methods perform well in many applications [11], they still encounter the following challenges. (1) These methods typically require searching for the *k*-NNs of each object. As the size of the dataset increases, this process becomes very time-consuming. This leads to the inefficiency of nearest neighbor-based outlier detection methods. (2) When the *k*-NNs of a object contain noise, these noise can interfere with distance or density calculations. As a result, normal points being misclassified as outliers or true outliers being overlooked, which reduces the model's accuracy.

In response to the challenges faced by the aforementioned *k*-NNs-based outlier detection methods, this paper introduces granular-ball (GB) computing [12] and proposes a fast and robust outlier detection method based on GB center isolation ($GBCI$) and region consistency ($GBCRC$), called FROD. The method first generates GBs of different granularities to cover the entire dataset. Subsequently, it utilizes $GBCI$ and $GBCRC$ to characterize the distribution features of objects to effectively identify outliers. The contributions of this paper are as follows.

(i) This method innovatively defines a global metric, $GBCI$, and a local metric, $GBCRC$, to synergistically characterize data distribution for outlier detection. Unlike traditional global metrics, $GBCI$

* Corresponding author.
E-mail addresses: rxwang0113@foxmail.com (R. Wang), jhwan@gdut.edu.cn (J. Wan), xpli@gdut.edu.cn (X. Li), ss_tan@163.com (S. Tan).

https://doi.org/10.1016/j.patcog.2026.113212
Received 22 April 2025; Received in revised form 8 January 2026; Accepted 29 January 2026
Available online 3 February 2026

quantifies the global isolation of each GB by calculating the average distance between its center and the centers of other GBs, providing a preliminary assessment of the global outlier degree for objects inside the GB. On the other hand, $GBCRC$ constructs the core region of a GB using the k-NNs of its center and evaluates the local structural consistency of an object through the distances to the objects in this core region, instead of using the object's own k-NNs for computation. By integrating global and local information, this method achieves accurate identification of outliers.

(ii) When calculating $GBCRC$, the k-NNs of an object's GB center are used instead of the object's own k-NNs to evaluate regional consistency. Since the number of GBs is far smaller than dataset objects, this approach significantly reduces k-NN search time and improves detection efficiency. Additionally, the GB center is computed via weighted averaging to mitigate minor noise impacts, ensuring its k-NNs are nearly noise-free. Unlike traditional local metrics that rely on an object's own k-NNs and are prone to noise interference, this design avoids noise affecting $GBCRC$ calculation and enhances outlier detection accuracy and stability.

(iii) By comparing FROD with several k-NNs-based and non-k-NNs detection methods across 20 datasets, FROD demonstrates not only excellent detection accuracy but also high detection efficiency. Additionally, FROD exhibits a certain level of robustness to noise.

The organization of this paper is as follows. The relevant work is introduced in Section 2. In Section 3, the method proposed in this paper is detailed. Section 4 presents the comparative experiments and analyzes the results. Finally, a summary and an outlook are given in Section 5.

2. Related work

Outlier detection methods can be primarily categorized into statistical-based [13], k-nearest neighbors (k-NNs)-based [14], fuzzy rough set-based [15] and deep learning-based [16,17] approaches. In recent years, deep learning-based methods have demonstrated outstanding performance in complex high-dimensional data. This success is largely attributed to their powerful representation learning capability, which can automatically extract hierarchical and discriminative features from raw data [18–20]. However, their black-box nature often results in limited interpretability of outcomes, restricting their applicability in scenarios requiring explainability. In contrast, k-NNs-based methods remain a mainstream choice in many practical scenarios due to their strong intuitiveness and high interpretability. This study also relies on k-NNs and distance metrics to conduct experiments and explorations in outlier detection. Therefore, this section will focus on reviewing k-NNs-based outlier detection methods. Additionally, it will briefly introduce the research advances in the field of granular-ball computing.

2.1. Nearest neighbors-based outlier detection method

The k-nearest neighbors (k-NNs)-based outlier detection methods can generally be divided into distance-based and density-based outlier detection methods.

2.1.1. Distance-based method

Distance-based methods assume that outliers are far from their k-NNs. In this method, outlier detection is performed by calculating the distance from an object to its k-NNs [14]. Ramaswamy et al. [21] determined whether an object is an outlier by calculating the distances between the object and its k-NNs, and then ranking these distances. Furthermore, Zhang et al. [22] proposed a new local distance-based outlier factor (LDOF), which measures the outlier degree of an object in the dataset by analyzing the relative position of the object with respect to its neighbors. Xie et al. [23] presented a local gravity-based outlier detection method (LGOD), which determines outliers by calculating the total gravitational force between an object and its neighbors. Yang et al. [24] developed a mean shift-based outlier detection method (MOD). This method replaces the original object value with the mean of its k-NNs, which performs a mean shift on the data. Then, the outlier degree of the object is assessed by evaluating the change in its position before and after three iterations. Although distance-based methods are simple and intuitive, they are influenced by local density issues. As shown in Fig. 1, the distance between the normal point B in the low-density region and its neighboring points is greater than the distance between the outlier point A in the high-density region and its neighboring points. This can lead to the misclassification of point A as a normal point.

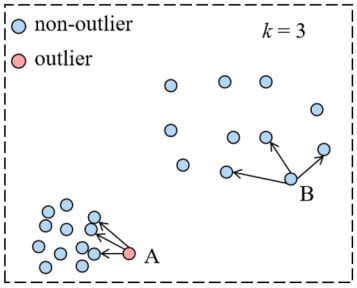


Fig. 1. Misclassification caused by density differences.

2.1.2. Density-based method

Density-based methods focus on the local density of objects, determining outliers by assessing their local density or the density difference with their neighbors [25]. Breunig et al. [26] assigned a local outlier factor (LOF) to each object by calculating the ratio of the neighborhood density to the object's own density. The higher the LOF value, the higher the outlier degree for the object. Cao et al. [27] presented a detection method based on the kernel neighborhood density change outlier factor (KNDCOF). The KNDCOF value reflects the density difference between an object and its kernel k-distance neighbors. The larger the KNDCOF value, the higher the outlier degree. Tang et al. [28] developed a relative density outlier score (RDOS) to measure the local outlier of an object. This method further considers reverse nearest neighbors and shared nearest neighbors, thus enabling better adaptation to complex data distributions. To avoid the uncertainty introduced by manually setting the k value, Zhang et al. [29] proposed an outlier detection method based on the relative skewness density ratio (SDROF). This method adaptively determines the number of neighbors through natural neighbor search. Additionally, Liu et al. [10] proposed a detection method based on potential energy and hubness score (PEHS). The core idea is to combine the concept of potential energy from physics with centrality from network analysis to identify outliers. Compared to distance-based methods, density-based methods are more effective in detecting outliers under most data distributions.

Table 1 summarizes the advantages and disadvantages of the aforementioned methods. As the data scale increases, both density-based and distance-based methods become computationally expensive in finding the k-NNs for each object, leading to low detection efficiency. Furthermore, when noise points are included in the k-NNs of a object, the calculation of distance or density may be disturbed, which affects the accuracy of outlier detection.

2.2. Granular-ball computing

Granular-ball computing (GBC) is an efficient, robust, and interpretable multi-granularity representation and computing method, which has gained widespread attention [30,31]. Xia et al. [12] used GBs to replace the original objects as inputs for support vector machines and k-NNs classifiers. This approach effectively enhanced the efficiency and robustness of the classifiers. Furthermore, they proposed a universal GB sampling method that significantly improves classification accuracy

Table 1
The advantages and disadvantages of methods based on k-nearest neighbors.

Categories	References	Methods	Advantages	Limitations
Distance-based	Ramaswamy et al. [21]	KNN	Easy to implement and straightforward	Sensitive to noisy Low detection efficiency Ineffective with imbalanced density data
	Zhang et al. [22]	LDOF		
	Xie et al. [23]	LGOD		
	Yang et al. [24]	MOD		
Density-based	Breunig et al. [26]	LOF	Insensitive to data distribution Effective for local outliers	Sensitive to noisy High computation cost Low detection efficiency
	Cao et al. [27]	KNDCOF		
	Tang et al. [28]	RDOS		
	Zhang et al. [29]	SDROF		
	Liu et al. [10]	PEHS		

in scenarios with noisy labels [32]. Chen et al. [30] introduced a GB-based selector that effectively performs attribute reduction through a data-adaptive sampling strategy, which enhances reduction efficiency. Peng et al. [33] proposed a robust variable parameter GB model aimed at achieving attribute reduction and classification in label noise environments from a coarse-grained perspective. Xie et al. [31] presented an improved Spectral Clustering algorithm based on GBs, which adaptively characterizes unlabeled data through a multi-granularity structure, significantly enhancing the efficiency and robustness of the algorithm. To address the high time complexity associated with analyzing high-dimensional data in prevalent learning, Cheng et al. [34] introduced GB into unsupervised prevalent learning and proposed the GB-USC and GB-USEC methods.

From the above content, it can be seen that the GBC has two main advantages: 1) it can improve the execution efficiency of tasks, and 2) it can effectively reduce the impact of noisy data on task outcomes. These two advantages correspond precisely to the challenges faced by k-NNs-based outlier detection methods. However, research combining GBC with outlier detection is relatively scarce [35]. Therefore, how to utilize GBC to improve outlier detection efficiency and reduce the impact of noise on detection results remains an issue that requires further exploration.

3. The proposed method

To address the challenges faced by existing k-nearest neighbors (k-NNs) detection methods, this section proposes a fast and robust outlier detection method (FROD).

3.1. The FROD framework

The FROD framework is shown in Fig. 2, which can be divided into two stages: granular-balls (GBs) generation and outlier detection. Specifically, the original dataset is first divided into some GBs. Subsequently, the GB center isolation ($GBCI$) is calculated to quantify the relative isolation degree of each GB with respect to others, which indirectly reflects the positional relationships among different GBs within the overall dataset. Next, we define the k-NNs relationships of the GB centers and calculate the GB center region consistency ($GBCRC$) based on this relationship to measure the region consistency of each object. Finally, by combining $GBCI$ and $GBCRC$ to get GB center outlier factor ($GBCOF$) of each object and compare it with a threshold.

3.2. The FROD method

In this subsection, we now detail the specific method employed in FROD, beginning with the granular-ball (GB) generation process.

3.2.1. Granular-ball generation

Given a dataset $D = \{x_1, x_2, \ldots, x_n\}$, it is divided into g GBs represented as $GBs = \{GB_1, GB_2, \ldots, GB_g\}$, where each $GB_i \subseteq D$ and satisfies $D = \bigcup_{i=1}^{g} GB_i$.

Definition 1. Given a $GB_i = \{x_j | j = 1, 2, \cdots, N\}$, where x_j represents the objects contained in the GB, and N denotes the number of objects in GB_i, i.e., $|GB_i| = N$. The center c_i and radius r_i are two important features of GB_i [12], which are respectively defined as

$$c_i = \frac{1}{|GB_i|} \sum_{x_j \in GB_i} x_j, \tag{1}$$

$$r_i = \frac{1}{|GB_i|} \sum_{x_j \in GB_i} d(x_j, c_i), \tag{2}$$

where d denotes the distance function. In this study, the distance function is defined as the Euclidean distance, represented as $d(x_i, x_j) = \sqrt{(x_i - x_j)^2}$.

GB generation is a core step of our method, with the partitioning strategy affecting subsequent outlier detection performance. Unlike most existing studies that only use K-means for GB generation, this paper introduces two distinct partitioning methods: Gaussian Mixture Model (GMM) and Self-Organizing Map (SOM) to systematically explore original data-to-GB transformation effects under different strategies and their impacts on detection performance.

Additionally, the quality of the GBs directly affects the execution efficiency and performance of subsequent tasks. In supervised tasks, such as feature selection, the purity of GBs (i.e., the category consistency of objects within the GB) is an important metric for evaluating their quality [36]. However, this study focuses on the unsupervised outlier detection task, where purity is no longer applicable. According to a commonly used empirical rule in cluster analysis, the upper bound for the optimal number of clusters is $\sqrt{n}$ [37]. Additionally, Yu et al. [38] theoretically validated this empirical criterion by defining the uncertainty of clusters and cluster spaces, and through convex function properties. GBs essentially represent local clustering of the data. Therefore, we constrain the GB size to $|GB| <= \sqrt{n}$ to ensure the stability and effectiveness of local clustering while avoiding over-refinement or excessive coarseness [39].

Algorithm 1 presents the process of generating GBs, which supports three clustering methods K-means, GMM, and SOM. For each method, the parameters are set as follows. The number of clusters for K-means is set to K = 2, the Gaussian components for GMM are set to n_components = 2, and SOM uses a grid structure with dimensions x = 1 and y = 2 (where x and y denote the number of rows and columns in the grid, respectively). These settings ensure that each method divides the current GB into two sub-balls (sub1 and sub2) for iterative generation.

Fig. 3 illustrates the entire process of generating GBs. Taking the 2-means algorithm and let $|GB| \leq \sqrt{n}$ as an example. First, the whole dataset is treated as an initial GB. Then, the 2-means clustering algorithm is used to split GBs that do not meet the threshold, and this process continues iteratively. The partitioning stops when a GB reaches a size of $|GB| \leq \sqrt{n}$. The figure not only shows the generation process of the GBs but also clearly displays how these GBs cover all objects and the center of each GB. As can be seen from the figure, the final GBs generated are multi-granular, with a single GB potentially encompassing the common

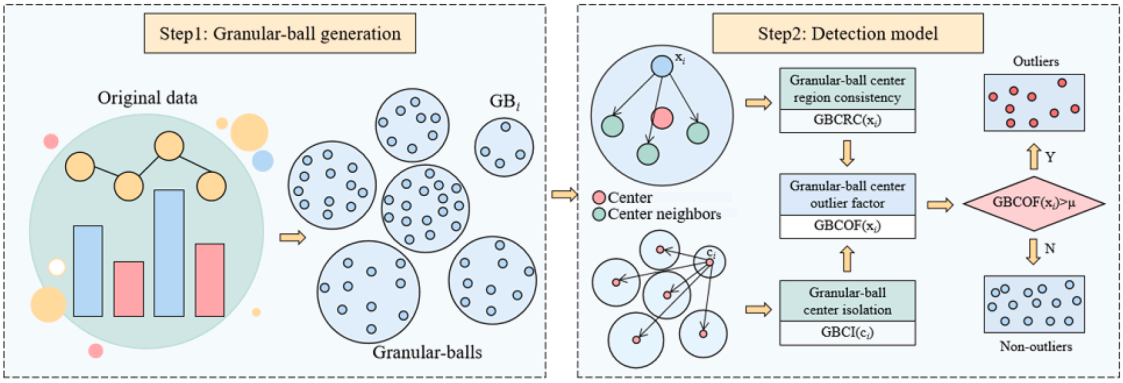


Fig. 2. The framework diagram of FROD.

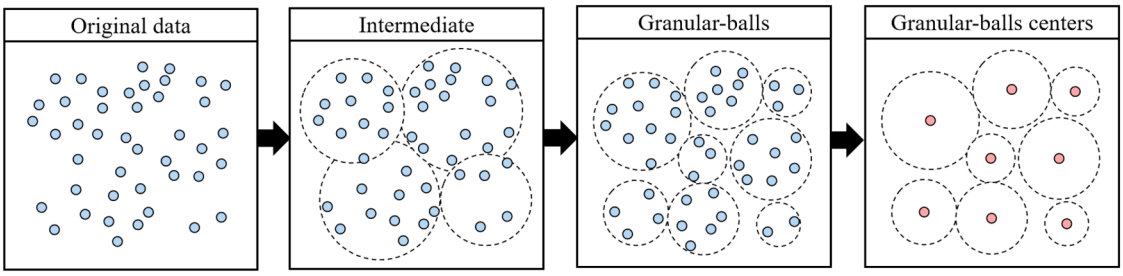


Fig. 3. The process of GBs generation.

Algorithm 1: GBs generation with K-means, GMM, and SOM.

Require: A dataset $D = \{x_1, x_2, \cdots, x_n\}$, clustering method C.
Ensure: GBs.

```
Initializing: n = |D|, GB = D, GBs = ∅;
Add GB to an empty queue Q;
while Q ≠ ∅ do
    Obtain the top GB from Q and remove it from Q;
    if |GB| > √n then
        if C is K-means then
            Divide GB into two sub-balls sub1 and sub2 using
            K-means (K = 2);
        else if C is GMM then
            Divide GB into two sub-balls sub1 and sub2 using
            GMM (n_components = 2);
        else if C is SOM then
            Divide GB into two sub-balls sub1 and sub2 using
            SOM (x = 1, y = 2);
        end if
        Add Sub1 and Sub2 to the tail of Q;
    end if
    if |GB| ≤ √n then
        GBs = GBs ∪ GB;
    end if
end while
return GBs.
```

features of multiple objects. The advantage of this structure is that it can effectively counter the interference of noise on the detection model. Additionally, the number of GBs $|GBs|$ is far smaller than the number of objects in the original dataset, allowing for the construction of an efficient detection model by using GBs as input instead of the original objects.

3.2.2. Detection model

After the generation of granular-balls (GBs), an efficient and robust outlier detection model is proposed in this subsection.

Definition 2. Given a set of $GBs = \{GB_1, GB_2, \ldots, GB_g\}$, with $\bigcup GBs = D$, the corresponding set of GB centers is $C = \{c_1, c_2, \cdots, c_g\}$. The GB center isolation for any $c_i \in C$ $(GBCI(c_i))$ is defined as

$$GBCI(c_i) = \frac{\sum_{c_j \in C - c_i} d(c_i, c_j)}{|C| - 1}. \tag{3}$$

$GBCI$ reflects the distribution of different GBs within the overall dataset. As shown in Fig. 4, GBs with darker centers are typically located in more remote areas, while those with lighter centers indicate closer proximity to other GB centers. A higher $GBCI$ value indicates that the GB center is more isolated relative to other GB centers. This suggests that the data distribution trend within its GB diverges significantly from the trends in other GBs, implying the potential presence of outliers within this GB. Unlike traditional global metrics that directly calculate distances between data points, $GBCI$ operates at the GB level by first evaluating the isolation degree of each GB relative to others, and then translating this assessment into a preliminary outlier degree evaluation for the objects contained within it.

Definition 3. Given a set of GB center $C = \{c_1, c_2, \cdots, c_g\}$, for any $c_i \in C$, its k-NNs $(ckNN(c_i))$ are defined as

$$ckNN(c_i) = \{x_1, x_2, \cdots, x_k\} \subseteq D, \tag{4}$$

where $\{x_1, x_2, \cdots, x_k\}$ are the k objects that are closest to c_i.

Definition 4. For any $x_i \in D$, if $x_i \in C_i$ and the corresponding GB center of C_i is c_i, the GB center region consistency $(GBCRC)$ for x_i $(GBCRC(x_i))$ is defined as

$$GBCRC(x_i) = \frac{1}{1 + \sum_{x_j \in ckNN(c_i)} d(x_i, x_j)}. \tag{5}$$

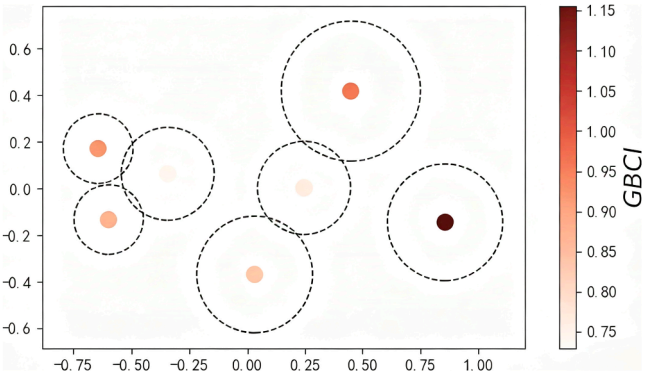


Fig. 4. Visualization of $GBCI$ values for GB centers.

The $GBCRC$ quantifies the consistency of a data object x_i with the core region of its corresponding GB, represented by $ckNN(c_i)$. A higher $GBCRC$ value indicates better consistency with the GB core region and a lower likelihood of x_i being an outlier. Conversely, a lower $GBCRC$ implies a higher probability of x_i being an outlier.

Traditional methods compute local metric based on the k-NNs of individual objects themselves, while $GBCRC$ uses the GB center as the reference point. It measures the consistency between a object and the core region of its GB by calculating the distance from the object to the k-NNs of the center point. This design brings two key advantages, which are as follows.

- **Reduce computational complexity:** By computing the $ckNN(c_i)$ of the GB center c_i, and utilizing $ckNN(c_i)$ to further calculate the $GBCRC$ of all objects in GB_i in batches. This approach avoids calculating the k-NNs for each object individually. Since the number of GB centers is much smaller than the number of objects in the dataset, this effectively reduces the overall time complexity of the algorithm.
- **Enhance model robustness:** The GB center c_i aggregates the information of multiple objects within the region, representing its overall distribution trend. Therefore, $ckNN(c_i)$ are primarily composed of objects consistent with this trend. A few noise points are typically balanced by the majority of normal data during the center calculation and thus rarely appear in $ckNN(c_i)$. This effectively reduces the interference of noise in the $GBCRC$ computation. Furthermore, from the perspective of the nature of outliers, a true outlier lies in the marginal region of the GB distribution. Even when it is microscopically close to a few normal points inside the ball, its overall distance to the core region of the GB remains significantly large. In contrast, traditional k-NNs methods often assign higher density values to such marginal outliers that are close to a small number of normal points within the ball, as their local density calculation is influenced by these neighboring points, which can easily lead to missed detections. In summary, this design effectively enhances the robustness of the detection model.

Definition 5. For any $x_i \in D$, if $x_i \in C_i$ and the corresponding GB center of C_i is c_i, the GB center outlier factor for x_i is $(GBCOF(x_i))$ defined as

$$GBCOF(x_i) = \frac{GBCI(c_i)}{GBCRC(x_i)}. \tag{6}$$

The $GBCOF$ combines the two factors of $GBCI$ and $GBCRC$ to measure the degree of abnormality of an object comprehensively. Specifically, the larger the outlier factor of an object, the greater its outlier degree, as it indicates that the GB center it belongs to is further away from other GB centers, while its region consistency is lower. Conversely, a smaller outlier factor indicates a smaller outlier degree.

Definition 6. Let a given threshold be μ. For any object x_i in the dataset D, if its outlier factor $GBCOF(x_i)$ satisfies $GBCOF(x_i) > \mu$, then x_i is classified as an outlier.

3.3. The FROD algorithm

Algorithm 2 the proposed FROD algorithm and analyzes its time complexity. Step 1 generates granular-balls (GBs) using Algorithm 1, and its time complexity depends on the clustering method. If K-means or SOM is used, the complexity is $O(mn)$; if GMM is employed, the complexity increases to $O(m^3 n)$. Steps 3-6 traverse each GB to compute its center set. Since $D = \bigcup_{i=1}^{g} GB_i$, the time complexity of Steps 3-6 is also $O(mn)$. Steps 7-9 iterate over each GB center to calculate its isolation and find its k-NNs. The number of iterations in this process is $|GBs|$. Additionally, a KD-tree is utilized to accelerate the nearest neighbor search in Step 9, and let $|GBs| = g$. Therefore, the time complexity for Steps 7-9 is $O(mg^2 + mg \log n)$. Steps 11-17 first compute the granular-ball center region consistency and then calculate the outlier factor for comparison against the threshold. The primary time cost of this process arises from calculating the granular-ball center region consistency, with a time complexity of $O(kmn)$. Consequently, the overall time complexity of the FROD algorithm varies depending on the granular-balls generation method employed. Its optimal time complexity is $O(m(n + g^2 + g \log n + kn))$ and the worst-case time complexity is $O(m(m^2 n + g^2 + g \log n + kn))$.

Algorithm 2: FROD.

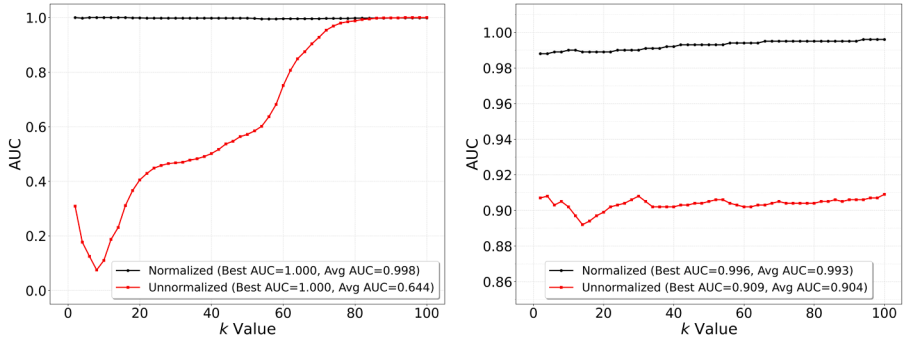
Require: A dataset $D = \{x_1, x_2, \cdots, x_n\}$; the size of neighbors k; the threshold μ.
Ensure: Outler set (OS).
1: Generate granular-balls GBs by Algorithm 1;
2: Initializing: center set $C = \emptyset$;
3: **for** every $GB_i \in GBs$ **do**
4: Calculate the center c_i of GB_i by Eq. (1);
5: $C \leftarrow C \cup c_i$;
6: **end for**
7: **for** every $c_i \in C$ **do**
8: Calculate the $GBCI(c_i)$ by Eq. (3);
9: $ckNN(c_i) \leftarrow$ find the k-NNs of c_i;
10: **end for**
11: **for** every $x_i \in D$ **do**
12: Calculate the $GBCRC(x_i)$ by Eq. (5);
13: Calculate the $GBCOF(x_i)$ by Eq. (6);
14: **if** $GBCOF(x_i) > \mu$ **then**
15: $OS \leftarrow OS \cup x_i$;
16: **end if**
17: **end for**
18: **return** OS.

4. Experiments

This section systematically evaluates the effectiveness of FROD (i.e., detection accuracy and detection efficiency) through comparative experiments with nine detection methods on twenty datasets. For the accuracy assessment, this experiment utilizes methods such as ROC curves, AUC values, boxplot, and statistical tests. The evaluation of detection efficiency is conducted by comparing the detection times of different algorithms on the same dataset. Additionally, noise experiments are performed to validate the robustness of FROD under various noise environments, and an analysis of the parameter sensitivity of FROD is conducted.

4.1. Experiment preparation

This subsection will outline the necessary preparations and settings for the experiments.



(a) Iris (b) Wdbc

Fig. 5. AUC results with varying values of k for normalized and unnormalized data.

Table 2
Experiment setup.

Software and hardware	Parameters
CPU	AMD Ryzen 7 7735H with Radeon Graphics 3.20 GHz
Memory	16GB
Hard Disk	1024GB
Operating System	64-bit Windows 11
Develop Environment	PyCharm 2023
Compilation Environment	Python 3.12
Virtualization Tools	Python 3.12

4.1.1. Experiment setting

The experimental setup is shown in Table 2, which mainly includes the software and hardware, as well as the parameters corresponding to each configuration.

4.1.2. Evaluation metric

To ensure the fairness and robustness of detection performance evaluation, we aim to avoid subjective biases introduced by the selection of specific decision thresholds. Therefore, this study adopts the receiver operating characteristic (ROC) curve and its area under the curve (AUC) as core evaluation metrics. In the outlier detection scenario, the ROC curve is constructed by generating a dynamic threshold sequence based on the outlier factors output by the algorithm. With the true positive rate (TPR) on the vertical axis and the false positive rate (FPR) on the horizontal axis, the curve serves as a complete performance characterization tool. The curve's morphology and AUC value objectively quantify algorithm performance across thresholds, providing a standardized evaluation framework for multi-model comparison independent of single-threshold settings.

The shape of the ROC curve intuitively reflects the model's trade-offs at different thresholds. The closer the curve is to the upper-left corner (i.e., high TPR and low FPR), the better the model's performance. The diagonal line (i.e., $TPR = FPR$) indicates that the model performs no better than random guessing, with no discriminative power. AUC measures a model's classification ability at different thresholds by calculating the area under the ROC curve. The AUC value ranges from 0 to 1, with values closer to 1 indicating stronger outlier detection capability.

4.1.3. Dataset

In this experiment, we selected twenty publicly available datasets for comparative experiments to evaluate the effectiveness of our method. These datasets are sourced from public websites[1], [2]. As shown in Table 3, the number of objects in these datasets ranges from 101 to 49,097, the number of attributes ranges from 4 to 279, and the number of outliers varies from 5 to 3,511, with outlier proportions ranging from 0.4% to 29.7%. All categorical attributes have been converted to integer values, and all attribute values have undergone min-max normalization, mapping them to the range of $[0, 1]$.

4.1.4. Comparison methods

We compared the FROD algorithm with nine other algorithms on the aforementioned twenty datasets. Table 4 presents the key descriptive information for each algorithm, including algorithm name (and corresponding year), parameters and their value ranges, as well as time complexity. Among these, ECOD [13] is parameter-free. The parameter σ for HGBAD [35] and WFRDA [40] algorithms has value ranges of $[0.1, 1]$ and $[0.1, 2]$ respectively, with a step of 0.1. In IForest [41], t denotes the number of trees, while ψ indicates the size of the subsample, corresponding parameters are set as $t = 100,\ \psi = 256$. DIF [42] is an extension of IForest, where r represents the number of deep ensemble members, with parameters set as $r = 50,\ t = 100,\ \psi = 256$. For k-nearest neighbors (k-NNs) algorithms such as DCROD [43], MOD [24], COF [44], LOF [26], and FROD, the range of the k parameter is $[2, 100]$ with a step of 2.

4.2. Impact of data preprocessing on detection performance

This subsection aims to investigate the impact of preprocessing steps on detection performance. It should be noted that in this subsection, we temporarily employ the K-means (K = 2) algorithm for granular-ball generation.

As shown in Fig. 5, the normalization process has a significant impact on the detection performance for both the Iris and Wdbc numerical datasets. For the Iris dataset, after normalization, FROD achieves a stable AUC close to 1.0 even at relatively small k values (average AUC = 0.998). In contrast, the unnormalized data exhibit significantly degraded performance over the same k range (average AUC = 0.644), with AUC values reaching higher levels only when k approaches 100. For the WDBC dataset, the AUC values of the unnormalized data consistently fluctuate at a lower level (approximately 0.90-0.91) across all k values and remain stably lower than those of the normalized data (approximately 0.99-0.996). These results validate the necessity of normalization in outlier detection. In unprocessed data, attributes with larger scales dominate the distance calculation, leading to metric distortion. Normalization, by scaling all attributes to the $[0, 1]$ interval, ensures equitable weighting of each attribute in the distance metric. It should be noted that for datasets containing categorical attributes, conversion to numerical types is essential for distance computation.

4.3. Ablation studies

This subsection conducts ablation studies to evaluate the time efficiency of different granular-ball (GB) generation methods and their

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

Table 3
Experiment datasets.

No.	Datasets	Abbr.	Objects	Attributes	Outilers	Percentage	Subject Area
1	Zoo_variant1	Zoo	101	16	17	16.8%	Biology
2	Iris_Irisvirginica_11_variant1	Iris	111	4	11	9.9%	Biology
3	Wine	Wine	129	13	10	7.8%	Business
4	Ionosphere_b_24_variant1	Iono	249	34	24	9.6%	Physics and Chemistry
5	Breast_cancer_variant1	Breast	286	9	85	29.7%	Health and Medicine
6	Ecoli	Ecoli	336	7	9	2.7%	Biology
7	Bands_band_27_variant1	Band27	339	39	27	8.0%	Physics and Chemistry
8	Bands_band_42_variant1	Band42	354	39	42	11.9%	Physics and Chemistry
9	Wdbc_M_39_variant1	Wdbc	396	31	39	9.8%	Health and Medicine
10	Arrhythmia_variant1	Arrhy	452	279	66	14.6%	Health and Medicine
11	Yeast_ERL_5_variant1	Yeast	1141	8	5	0.4%	Biology
12	Cardio	Cardio	1831	21	176	9.6%	Health and Medicine
13	Chess_nowin_227_variant1	Chess	1896	36	227	12%	Games
14	Waveform_0_100_variant1	Wave	3443	21	100	2.9%	Physics and Chemistry
15	Mushroom_p_365_variant1	Mush	4573	22	365	8.0%	Biology
16	Pageblocks_1_258_variant1	Page	5171	10	258	5.0%	Computer Science
17	Satimage2	Sati	5803	36	71	1.2%	Climate and Environment
18	Mnist	Mnist	7603	100	700	9.2%	Computer Vision
19	Adult_morethan50K_343_variant1	Adult	34357	14	343	1.0%	Social Science
20	Shuttle	Shut	49097	9	3511	7.2%	Physics and Chemistry

Table 4
The description of experimental algorithms.

Categories	Algorithms (Year)	Description	Parameter ranges	Time complexities
Others	HGBAD (2024) [35]	Hybrid granular-ball fuzzy information granules-based anomaly detection	$\sigma \in [0.1, 1]$, stepsize = 0.1	$O(m(n + \lvert GBs\rvert^2))$
	WFRDA (2023) [40]	Weighted fuzzy rough density-based anomaly detection	$\sigma \in [0.1, 2]$, stepsize = 0.1	$O(mn^2)$
	DIF (2023) [42]	Deep isolation forest for anomaly detection	$r = 50; t = 6; \psi = 256$	$O(mnrt)$
	ECOD (2022) [13]	Outlier detection using empirical cumulative distribution functions	None	$O(mn)$
	IForest (2012) [41]	Isolation forest-based anomaly detection	$t = 100, \psi = 256$	$O(t\psi^2 + nt\psi)$
k-NNs-based	DCROD (2022) [43]	Directed density ratio changing rate-based outlier detection	$k \in [2, 100]$, stepsize = 2	$O(mn \log n)$
	MOD (2021) [24]	Mean-shift outlier detection	$k \in [2, 100]$, stepsize = 2	$O(mn^2)$
	COF (2002) [44]	Connectivity-based outlier factor	$k \in [2, 100]$, stepsize = 2	$O(mn^2)$
	LOF (2000) [26]	Identifying density-based local outliers	$k \in [2, 100]$, stepsize = 2	$O(mn^2)$
	FROD (Ours)	Outlier detection based on granular-ball center isolation and region consistency	$k \in [2, 100]$, stepsize = 2	$O(m(n + g^2 + glogn + kn))$

impacts on subsequent outlier detection performance. The three clustering methods adopted in the experiments (K-means, SOM and GMM) all maintain consistent parameter settings with those described in Section 3.1 (Algorithm 1). Specifically, the number of clusters for K-means is set to $K = 2$, the number of Gaussian components for GMM is set to $n\_components = 2$, and SOM uses a grid structure of 1×2 (i.e. $x = 1$ row and $y = 2$ columns). The above settings ensure that each method can stably split the current GB into two sub-balls during the iterative process.

Fig. 6 presents the comparison of GB generation time of the three methods across all datasets. K-means achieves the optimal time efficiency with an average generation time of 0.424 seconds and the maximum time consumption on each dataset does not exceed 1.891 seconds. SOM ranks second in efficiency with an average time of 0.607 seconds although its time consumption is slightly lower than that of K-means on datasets such as Zoo, Iris, and Wine its generation time can reach 4.612 seconds on complex datasets like Mnist indicating relatively weak stability. GMM has the highest computational overhead with an average time of 5.051 seconds which is approximately 11.9 times that of K-means and even as high as 21.451 seconds on the Shut dataset.

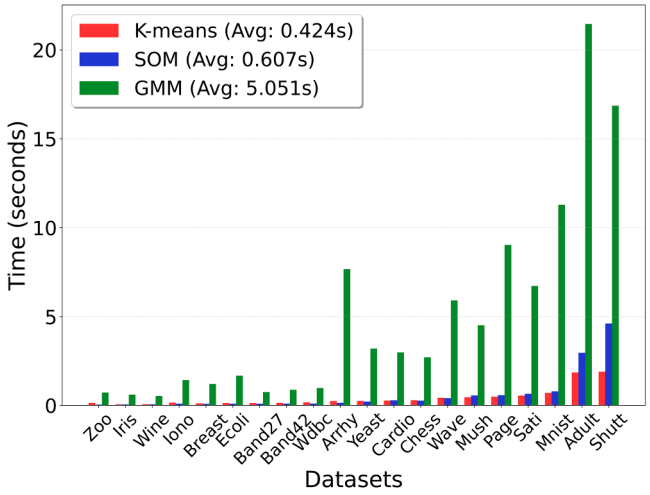
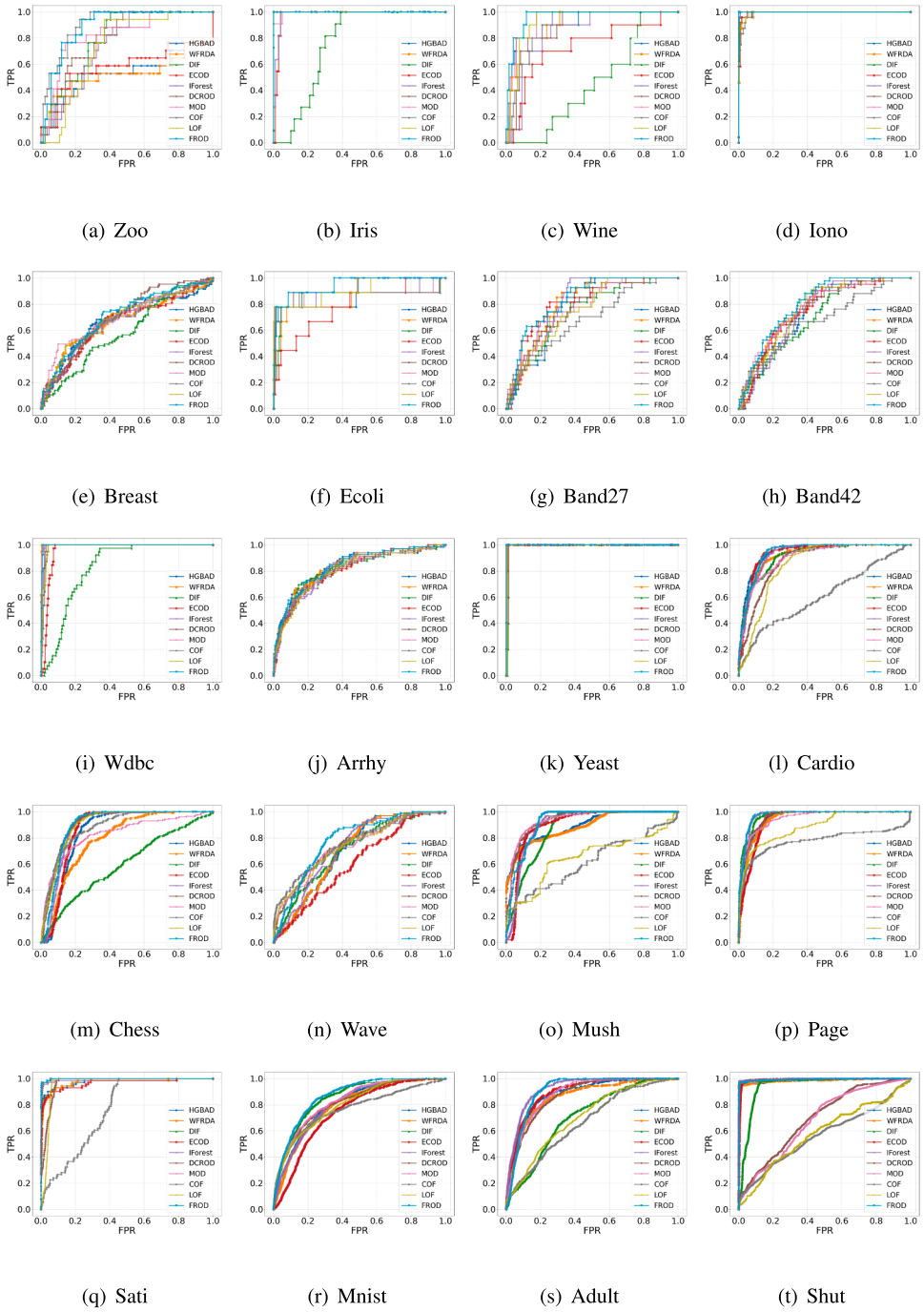


Fig. 6. Comparison of GB generation time using different clustering methods.

Table 5 summarizes the outlier detection AUC results of three GB generation methods under three different size constraints across 20 datasets. Among these results, values with bold and underlined formatting indicate the optimal-performing method under the corresponding constraint. Overall, the average performance of the three methods is highly comparable (AUC ranging from 0.902 to 0.913), demonstrating that the proposed detection framework exhibits strong robustness to the selection of GBs generation methods. Further analysis reveals that under different size constraints, all GB generation methods exhibit favorable outlier detection performance, with their AUC values consistently maintaining a high level. Notably, under the constraint of $|GB| \leq \sqrt{n}$, the average AUC of the three methods reaches or approaches the optimal level. In particular, K-means achieves the best average performance under this constraint (AUC = 0.913), which is slightly superior to that of SOM and



(a) Zoo (b) Iris (c) Wine (d) Iono

(e) Breast (f) Ecoli (g) Band27 (h) Band42

(i) Wdbc (j) Arrhy (k) Yeast (l) Cardio

(m) Chess (n) Wave (o) Mush (p) Page

(q) Sati (r) Mnist (s) Adult (t) Shut

Fig. 7. Comparison results of ten outlier detection algorithms on the ROC curve.

GMM (both 0.907). This can be attributed to the compact spherical cluster structure formed by K-means, which is highly consistent with the geometric assumption of GBs. To sum up, K-means outperforms SOM and GMM in both time efficiency and detection effectiveness. Therefore, K-means (2-means) will be uniformly adopted as the GB generation method in subsequent comparative experiments, with $|GB| \leq \sqrt{n}$ set as the size constraint.

4.4. Experimental results and analysis

In this subsection, we conducted a comparative analysis of the ROC curves, AUC values, and boxplot of FROD against various detection algorithms across twenty datasets.

4.4.1. ROC curves analysis

Fig. 7 illustrates the ROC curves of ten algorithms across twenty datasets, with the cyan color representing the ROC curve of the FROD algorithm. According to the definition of the evaluation metrics, the closer the ROC curve is to the upper-left corner of the first quadrant, the better the algorithm's performance. From Fig. 7, it is evident that the ROC curve of the FROD algorithm is closest to the upper-left corner on datasets such as Wine, Ecoli, Band27, Band42, Wave, Sati, and Mnist, indicating its superior performance compared to other algorithms. Further analysis reveals that on datasets such as Iris, Iono, and Yeast, the ROC curve of the FROD algorithm nearly coincides with the axes, showcasing its outstanding performance on these datasets. Additionally, on certain datasets, such as Zoo, Breast, and Arrhy, the ROC curve of the

Table 5
The AUC comparison using different GB generation methods under different size constraints.

Datasets	$\|GB\| <= \sqrt{n}$			$\|GB\| <= \sqrt{n/2}$			$\|GB\| <= \sqrt{2n}$		
	K-means	SOM	GMM	K-means	SOM	GMM	K-means	SOM	GMM
Zoo	0.894	0.934	0.894	0.888	0.893	0.888	0.888	0.915	0.888
Iris	1.000	0.999	1.000	1.000	0.999	1.000	1.000	0.999	1.000
Wine	0.964	0.932	0.948	0.929	0.897	0.903	0.961	0.929	0.944
Iono	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Breast	0.704	0.703	0.681	0.705	0.686	0.691	0.694	0.692	0.688
Ecoli	0.946	0.888	0.893	0.906	0.890	0.901	0.922	0.898	0.891
Band27	0.833	0.816	0.833	0.832	0.809	0.832	0.826	0.811	0.826
Band42	0.803	0.805	0.803	0.815	0.797	0.815	0.807	0.807	0.807
Wdbc	0.996	0.996	0.997	0.996	0.996	0.995	0.996	0.996	0.997
Arrhy	0.834	0.811	0.834	0.830	0.816	0.830	0.831	0.802	0.831
Yeast	0.999	1.000	0.999	0.999	1.000	0.999	0.999	1.000	0.999
Cardio	0.940	0.941	0.940	0.942	0.942	0.939	0.939	0.940	0.936
Chess	0.898	0.893	0.906	0.904	0.895	0.905	0.894	0.881	0.897
Wave	0.778	0.763	0.774	0.784	0.772	0.773	0.777	0.754	0.769
Mush	0.923	0.917	0.925	0.939	0.923	0.945	0.916	0.881	0.928
Page	0.973	0.968	0.977	0.973	0.971	0.977	0.972	0.968	0.981
Sati	0.999	0.997	0.998	0.999	0.997	0.998	0.999	0.997	0.998
Mnist	0.865	0.866	0.850	0.860	0.866	0.856	0.859	0.873	0.844
Adult	0.908	0.908	0.898	0.911	0.909	0.896	0.908	0.901	0.894
Shut	0.993	0.993	0.994	0.993	0.993	0.994	0.993	0.992	0.994
Average	**0.913**	0.907	0.907	**0.910**	0.903	0.907	**0.909**	0.902	0.906

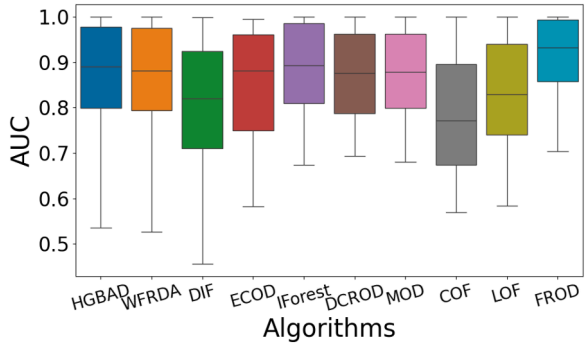


Fig. 8. The AUC boxplots for each method across 20 datasets.

FROD algorithm overlaps with those of other algorithms. This overlap complicates the determination of which algorithm is absolutely superior. Thus, we further present a table of the AUC values of the ten detection algorithms for a clearer comparison of their performances.

4.4.2. AUC value analysis

Table 6 presents the AUC values of ten algorithms across multiple datasets, with the best AUC value for each dataset highlighted in bold and underlined. This table facilitates a more intuitive comparison of the FROD algorithm's performance against other algorithms across different datasets. In terms of optimal AUC values, FROD achieved the best AUC value on 12 out of the 20 datasets. Although it did not attain the best AUC value on certain datasets, such as Zoo, Cardio, and Chess, its results remain close to the optimal values. This demonstrates the outstanding performance of FROD on the vast majority of datasets. Furthermore, FROD's average AUC value across the 20 datasets is 0.913, significantly higher than that of the other algorithms, further confirming its remarkable advantage in overall performance. To provide a holistic perspective on algorithm performance distribution and stability across all datasets, the AUC results are further visualized through boxplot analysis in Fig. 8.

Fig. 8 shows the distribution of AUC values for different detection algorithms on 20 data sets in a boxplot form. It can be observed from the boxplot that the box of the FROD algorithm is relatively compact and located above the overall distribution. Specifically, first of all, the box being located above the overall distribution means that the FROD algorithm has higher AUC values on most datasets, showing better performance than other detection algorithms. Secondly, the compactness of the box shows that the AUC value of the FROD algorithm fluctuates little. It reflects that FROD has high stability and consistency under different data sets.

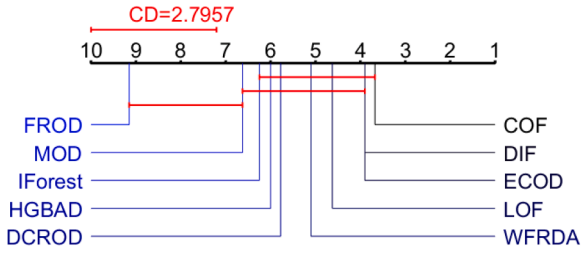


Fig. 9. Nemenyi test on AUC.

To sum up, the FROD algorithm shows superior performance from three aspects: ROC curve, AUC value and boxplot. These results show that the FROD algorithm can maintain high accuracy on different data sets, has good stability and consistency, and can provide reliable detection results in a variety of environments, proving its advantages in accuracy.

4.5. Statistical analysis

This subsection employs the Friedman test and Nemenyi post-hoc test for statistical analysis of experimental results across different algorithms [45]. The Friedman test ranks the AUC values of each algorithm across all datasets to examine significant performance differences among algorithms. It corely calculates the τ_F statistic to determine the validity of the null hypothesis. If the null hypothesis is rejected, the Nemenyi post-hoc test is further used to analyze specific differences. The key of the Nemenyi test lies in computing the critical difference (CD), which intuitively reveals significant differences between algorithms through the connection of horizontal line segments in the test plot. Algorithms connected by the same horizontal line segment have no significant differences.

A total of 10 algorithms and 20 datasets were used in the experiments. The τ_F conforms to the F distribution with degrees of freedom 9 and 171. According to Friedman's test, the computed $\tau_F = 8.1911$ exceeds the critical value of 1.6684 at the significance level $\alpha = 0.1$, indicating that there is a significant difference between the different algorithms. Therefore, the specific algorithmic differences were further analysed using the Nemenyi post hoc test.

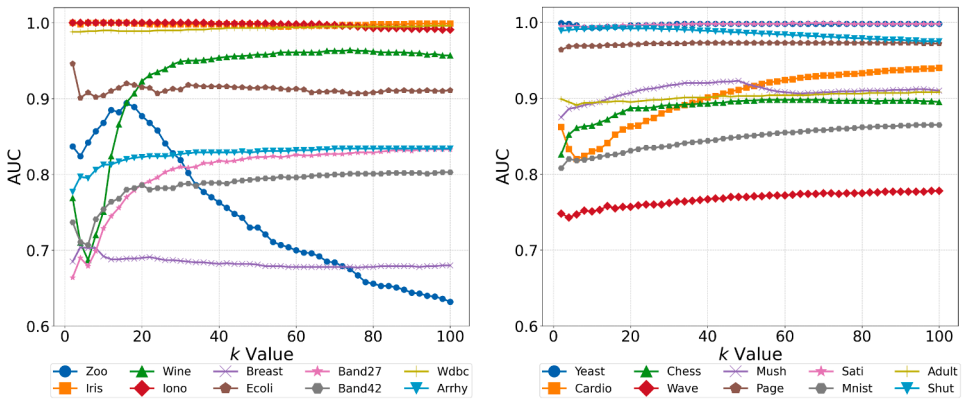
In the Nemenyi test, the significance level is set at $\alpha = 0.1$, resulting in a critical difference of $CD_{0.1} = 2.7957$. A Nemenyi test plot can be generated by plotting the line between the average ranking of the algorithm and the CD. As shown in Fig. 9, if certain methods are not connected by the red horizontal segment CD, it indicates a significant statistical difference between these methods. Otherwise, there is none. From Fig. 9, it can be observed that the FROD algorithm is not connected by a line segment to other algorithms (such as HGBAD, DCROD, WFRDA, etc.) except for the MOD algorithm. This indicates that there is a statistically significant difference between the FROD algorithm and the other algorithms, excluding the MOD algorithm.

4.6. Parameter sensitivity analysis

In this subsection, we investigate the sensitivity of FROD to the parameter k, where k represents the size of the neighborhood around the centers of the granular-balls. Fig. 10 illustrates the variation curves of the parameter k and AUC values across different datasets. From the figure, it is observed that in most datasets, when $k < 20$, the AUC values fluctuate significantly and are unstable. This instability arises because a

Table 6
Experimental comparison results on AUC.

Datasets	Others					*k*-NNs-based				
	HGBAD	WFRDA	DIF	ECOD	IForest	DCROD	MOD	COF	LOF	FROD
Zoo	0.536	0.527	0.800	0.582	0.749	0.792	0.829	**0.906**	0.730	0.894
Iris	**1.000**	**1.000**	0.757	0.977	0.984	**1.000**	0.995	**1.000**	**1.000**	**1.000**
Wine	0.912	0.915	0.456	0.733	0.842	0.883	0.940	0.892	0.932	**0.964**
Iono	0.996	0.993	0.998	0.994	**1.000**	**1.000**	**1.000**	0.991	0.995	**1.000**
Breast	0.675	0.670	0.590	0.656	0.674	0.694	0.700	0.671	0.689	**0.704**
Ecoli	0.886	0.875	0.856	0.781	0.872	0.874	0.875	0.869	0.906	**0.946**
Band27	0.784	0.802	0.743	0.806	0.818	0.773	0.763	0.686	0.756	**0.833**
Band42	0.737	0.762	0.700	0.750	0.741	0.759	0.773	0.674	0.772	**0.803**
Wdbc	**0.999**	**0.999**	0.825	0.959	0.991	0.986	0.995	0.984	0.986	0.996
Arrhy	0.831	0.826	0.816	0.807	0.810	0.802	0.808	0.810	0.807	**0.834**
Yeast	0.998	0.998	0.987	0.995	0.997	0.990	**1.000**	**1.000**	0.994	0.999
Cardio	**0.945**	0.922	0.917	0.935	0.931	0.878	0.905	0.610	0.850	0.940
Chess	0.848	0.794	0.599	0.865	0.884	**0.908**	0.830	0.883	0.904	0.898
Wave	0.700	0.704	0.707	0.608	0.722	0.748	0.738	0.757	0.743	**0.778**
Mush	0.895	0.887	0.875	0.900	0.903	**0.936**	0.934	0.621	0.654	0.923
Page	0.952	0.944	0.971	0.938	0.970	0.959	0.951	0.779	0.887	**0.973**
Sati	0.974	0.972	0.996	0.965	0.994	0.969	**0.999**	0.761	0.962	**0.999**
Mnist	0.804	0.792	0.853	0.746	0.806	0.819	0.832	0.765	0.803	**0.865**
Adult	0.881	0.861	0.712	0.898	**0.925**	0.867	0.881	0.671	0.708	0.908
Shut	0.988	0.986	0.945	0.993	**0.996**	0.697	0.681	0.569	0.584	0.993
Average	0.867	0.861	0.805	0.844	0.880	0.867	0.871	0.795	0.833	**0.913**



(a) Datasets 1-10 (b) Datasets 11-20

Fig. 10. AUC results with varying values of k.

smaller k restricts the discrimination of each object to a limited number of neighboring granular-ball centers, making it sensitive to local data variations and leading to unstable outlier detection. However, when $k \geq 20$, as the value of k increases, the AUC values for most datasets tend to stabilize. A notable exception is the Zoo dataset, where performance declines with increasing k. This is due to the dataset's small size and discrete attribute space, which result in a few well-separated, small granular-balls. A large k causes the neighborhood of a granular-ball center to extend beyond its boundary, incorporating dissimilar objects from other granular-balls and affecting the region consistency estimation. Therefore, it impacts the detection performance of the algorithm. Overall, the FROD algorithm demonstrates good stability across a broad range of k values in most datasets.

4.7. Detection efficiency analysis

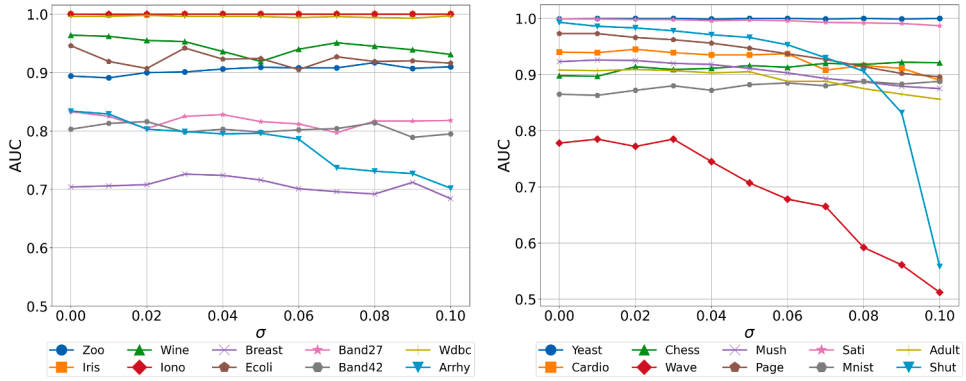
This subsection evaluates the running efficiency of the FROD algorithm by comparing the running times of different algorithms across 20 datasets, where the dataset sizes increase from Zoo to Shut. The following observations can be made from Table 7.

- **Small-scale datasets:** When $n < 10^3$, the performance of the FROD algorithm is inferior to that of most other algorithms, with relatively longer detection times. This is mainly because the FROD algorithm requires additional time to generate granular-balls when processing data.
- **Medium-scale datasets:** When $10^3 < n < 10^4$, as the data size increases, for instance, within the range from Mush to Mnist datasets, the detection time of FROD gradually becomes lower than that of other algorithms, except for ECOD and IForest. Moreover, as the data size continues to grow, the advantages of FROD become even more pronounced.
- **Large-scale datasets:** When $n > 10^4$, although the detection time of the FROD algorithm is still higher than that of ECOD and IForest, it is significantly lower than that of other k-NNs algorithms as well as non-k-NNs algorithms, demonstrating a considerable advantage.

In summary, the average detection time of the FROD algorithm is 0.654 seconds, which is significantly better than that of other k-NNs algorithms and non-k-NNs algorithms, except for ECOD and IForest. This advantage becomes even more apparent when the dataset sizes are larger.

Table 7
Execution time (Seconds) ($k = 50$).

Data size n	Datasets	Others					k-NNs-based				
		HGBAD	WFRDA	DIF	ECOD	IForest	DCROD	MOD	COF	LOF	FROD
$n < 10^3$	Zoo	1.339	0.002	1.189	0.004	0.096	0.038	1.28	0.035	0.023	0.152
	Iris	1.008	0.002	1.505	0.003	0.097	0.032	0.141	0.037	0.022	0.155
	Wine	3.667	0.009	1.330	0.003	0.104	0.039	0.146	0.058	0.026	0.071
	Iono	20.922	0.009	1.829	0.007	0.093	0.084	3.056	0.103	0.057	0.305
	Breast	4.181	0.005	1.777	0.003	0.097	0.076	0.257	0.107	0.058	0.110
	Ecoli	4.837	0.005	2.381	0.005	0.094	0.094	0.323	0.149	0.074	0.139
	Band27	34.817	0.013	2.144	0.008	0.094	0.110	4.532	0.151	0.078	0.264
	Band42	36.913	0.015	2.157	0.007	0.098	0.116	5.173	0.148	0.081	0.300
	Wdbc	33.788	0.024	2.495	0.011	0.091	0.126	5.268	0.170	0.094	0.375
	Arrhy	236.386	0.208	2.931	0.75	0.088	0.302	6.012	0.287	0.155	0.556
$10^3 < n < 10^4$	Yeast	24.169	0.062	5.887	0.004	0.098	0.260	0.896	0.719	0.367	0.261
	Cardio	115.685	0.473	9.636	0.014	0.115	0.416	14.858	1.676	0.859	0.457
	Chess	74.591	1.002	10.463	0.023	0.257	0.480	17.888	2.486	1.077	0.513
	Wave	284.982	2.067	16.347	0.045	0.129	0.716	27.601	5.699	3.017	0.717
	Mush	240.135	2.648	20.508	0.033	0.147	0.890	37.845	11.005	5.536	0.739
	Page	261.877	2.234	24.309	0.030	0.138	1.128	4.333	14.837	6.791	0.653
	Sati	939.107	7.995	25.564	0.109	0.155	1.282	47.667	31.503	14.765	0.949
	Mnist	2889.739	46.045	34.997	0.332	0.195	2.708	67.18	94.21	45.907	1.188
$n > 10^4$	Adult	5520.465	714.604	114.717	0.217	0.445	16.716	38.432	921.906	550.56	2.264
	Shut	8638.112	2606.791	145.961	0.209	0.535	11.645	41.816	6481.043	3043.307	2.921
	Average	968.336	169.210	21.406	0.091	0.158	1.863	16.235	378.316	183.643	0.654



(a) Datasets 1-10 (b) Datasets 11-20

Fig. 11. The variation curves of AUC values with increasing standard deviation σ.

4.8. Detection robustness analysis

This subsection evaluates the robustness of the FROD algorithm for outlier detection by injecting Gaussian noise of varying intensities into the normalized attributes of inliers, with AUC adopted as the performance metric. Specifically, for each inlier object x_i, Gaussian noise is injected into its normalized attribute values following the formula $a'_{i,j} = a_{i,j} + \mathcal{N}(0, \sigma^2)$, where $a_{i,j}$ represents the normalized value of the i-th object on the j-th attribute. The Gaussian noise standard deviation σ is increased from 0 to 0.10 in steps of 0.01. This range is appropriate for attributes normalized to $[0, 1]$, covering a test interval from slight to moderate noise.

Fig. 11 illustrates the effect of increasing σ on the AUC values of different datasets. As shown in the figure, as the noise intensity gradually increases, the AUC values of most datasets demonstrate good stability. For example, the AUC values of datasets such as Iris, Iono, Yeast, and Sati remain largely unaffected by noise, consistently staying above 0.99, with no significant decrease even when the σ reaches 0.1. On datasets such as Zoo, Chess, and Mnist, the AUC values exhibit only minor fluctuations with increasing noise, with overall changes not exceeding 0.05, and they still maintain a high level of performance. In contrast, the AUC values of the Wave and Shut datasets show a significant decline as noise increases, especially when $\sigma \geq 0.04$, where the downward trend becomes notably more pronounced. The reason for this phenomenon may be that the distributions of inliers and outliers in these two datasets have a high degree of overlap, and as the noise intensity increases, the boundary between the two types of objects becomes further blurred. Overall, in the face of progressively increasing noise, the FROD algorithm maintains stable outlier detection performance across most datasets, demonstrating strong robustness.

5. Conclusion and future work

In response to the inefficiency and noise sensitivity issues of traditional k-nearest neighbor methods in outlier detection, this paper proposes a fast and robust outlier detection (FROD) method based on granular-ball center isolation and region consistency. Experimental results demonstrate that FROD performs excellently across 20 datasets, effectively mitigating noise interference while achieving high detection accuracy and faster processing speed. However, the stability of FROD is compromised when the parameter k is set to small values. Although increasing k stabilizes its performance, it also incurs additional compu-

tational overhead. Therefore, future research should focus on developing adaptive parameter adjustment mechanisms to select a k value that achieves a balance between efficiency and performance. Moreover, the numerical processing of categorical data in FROD may lead to information loss. Subsequent studies should prioritize exploring more effective strategies for handling categorical data to better preserve its intrinsic structure and enhance the overall performance of the algorithm.

CRediT authorship contribution statement

Rongxiang Wang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization; **Jihong Wan:** Writing – review & editing, Supervision, Resources, Funding acquisition; **Xiaoping Li:** Writing – review & editing, Supervision, Funding acquisition; **Shuaishuai Tan:** Writing – review & editing, Supervision.

Data availability

Data will be made available on request.

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.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 62406079, the Key Laboratory of Ecological Security Monitoring and Governance at Sichuan Minzu College of Sichuan Province under Grant ESMK2025004, and the National Key Research and Development Program of China under Grant 2022YFB3305500.

References

[1] R. Domingues, M. Filippone, P. Michiardi, J. Zouaoui, A comparative evaluation of outlier detection algorithms: Experiments and analyses, Pattern Recognit. 74 (2018) 406–421.
[2] Z. Zhang, Q. Feng, J. Huang, Y. Guo, J. Xu, J. Wang, A local search algorithm for k-means with outliers, Neurocomputing 450 (2021) 230–241.
[3] C.G. Tekkali, K. Natarajan, RDQN: ensemble of deep neural network with reinforcement learning in classification based on rough set theory for digital transactional fraud detection, Complex Intell. Syst. 9 (5) (2023) 5313–5332.
[4] X. Wang, M.M. Ahmed, M.N. Husen, Z. Qian, S.B. Belhaouari, Evolving anomaly detection for network streaming data, Inf. Sci. 608 (2022) 757–777.
[5] B. Wang, Z. Mao, Outlier detection based on Gaussian process with application to industrial processes, Appl. Soft Comput. 76 (2019) 505–516.
[6] R.D.H. Devi, M.I. Devi, Outlier detection algorithm combined with decision tree classifier for early diagnosis of breast cancer, Int. J. Adv. Eng. Technol. 93 (2016) 98–103.
[7] M. Kontaki, A. Gounaris, A.N. Papadopoulos, K. Tsichlas, Y. Manolopoulos, Efficient and flexible algorithms for monitoring distance-based outliers over data streams, Inf. Syst. 55 (2016) 37–53.
[8] 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.
[9] L.A.S. Arias, C.W. Oosterlee, P. Cirillo, AIDA: analytic isolation and distance-based anomaly detection algorithm, Pattern Recognit. 141 (2023) 109607.
[10] H. Liu, S. Zhang, Z. Wu, X. Li, Outlier detection using local density and global structure, Pattern Recognit. 157 (2025) 110947.
[11] Y. Zhang, N. Meratnia, P. Havinga, Outlier detection techniques for wireless sensor networks: A survey, IEEE Commun. Surv. Tut. 12 (2) (2010) 159–170.
[12] S. Xia, Y. Liu, X. Ding, G. Wang, H. Yu, Y. Luo, Granular ball computing classifiers for efficient, scalable and robust learning, Inf. Sci. 483 (2019) 136–152.
[13] 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.
[14] D. Samariya, A. Thakkar, A comprehensive survey of anomaly detection algorithms, Annals Data Sci. 10 (3) (2023) 829–850.
[15] X. Su, Z. Yuan, B. Chen, D. Peng, H. Chen, Y. Chen, Detecting anomalies with granular-ball fuzzy rough sets, Inf. Sci. 678 (2024) 121016.
[16] D. Chakraborty, V. Narayanan, A. Ghosh, Integration of deep feature extraction and ensemble learning for outlier detection, Pattern Recognit. 89 (2019) 161–171.
[17] X. Tan, J. Yang, J. Chen, S. Rahardja, S. Rahardja, MSS-PAE: saving Autoencoder-based Outlier Detection from Unexpected Reconstruction, Pattern Recognit. 163 (2025) 111467.
[18] H. Tang, C. Yuan, Z. Li, J. Tang, Learning attention-guided pyramidal features for few-shot fine-grained recognition, Pattern Recognit. 130 (2022) 108792.
[19] S. Yan, H. Tang, L. Zhang, J. Tang, Image-specific information suppression and implicit local alignment for text-based person search, IEEE Trans. Neural Netw. Learn. Syst. 35 (12) (2023) 17973–17986.
[20] H. Tang, Z. Li, D. Zhang, S. He, J. Tang, Divide-and-conquer: Confluent triple-flow network for RGB-T salient object detection, IEEE Trans. Pattern Anal. Mach. Intell. 47 (3) (2024) 1958–1974.
[21] S. Ramaswamy, R. Rastogi, K. Shim, Efficient algorithms for mining outliers from large data sets, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of data, 2000, pp. 427–438.
[22] K. Zhang, M. Hutter, 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, Springer, 2009, pp. 813–822.
[23] J. Xie, Z. Xiong, Q. Dai, X. Wang, Y. Zhang, A local-gravitation-based method for the detection of outliers and boundary points, Knowl. Based Syst. 192 (2020) 105331.
[24] J. Yang, S. Rahardja, P. Fränti, Mean-shift outlier detection and filtering, Pattern Recognit. 115 (2021) 107874.
[25] A. Boukerche, L. Zheng, O. Alfandi, Outlier detection: Methods, models, and classification, ACM Comput. Surv. (CSUR) 53 (3) (2020) 1–37.
[26] M.M. Breunig, H. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of data, 2000, pp. 93–104.
[27] H. Cao, R. Ma, H. Ren, S.S. Ge, Data-defect inspection with kernel-neighbor-density-change outlier factor, IEEE Trans. Autom. Sci. Eng. 15 (1) (2016) 225–238.
[28] B. Tang, H. He, A local density-based approach for outlier detection, Neurocomputing 241 (2017) 171–180.
[29] Z. Zhang, K. Wang, J. Dong, S. Li, SDROF: outlier detection algorithm based on relative skewness density ratio outlier factor, Applied Intell. 55 (1) (2025) 1–21.
[30] Y. Chen, P. Wang, X. Yang, J. Mi, D. Liu, Granular ball guided selector for attribute reduction, Knowl. Based Syst. 229 (2021) 107326.
[31] 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.
[32] S. Xia, S. Zheng, G. Wang, X. Gao, B. Wang, Granular ball sampling for noisy label classification or imbalanced classification, IEEE Trans. Neural Netw. Learn. Syst. 34 (4) (2021) 2144–2155.
[33] X. Peng, P. Wang, S. Xia, C. Wang, W. Chen, VPGB: a granular-ball based model for attribute reduction and classification with label noise, Inf. Sci. 611 (2022) 504–521.
[34] D. Cheng, S. Liu, S. Xia, G. Wang, Granular-ball computing-based manifold clustering algorithms for ultra-scalable data, Expert Syst. Appl. 247 (2024) 123313.
[35] X. Su, X. Wang, D. Peng, H. Chen, Y. Chen, Z. Yuan, Granular-ball computing guided anomaly detection for hybrid attribute data, Int. J. Mach. Learn. Cybern. 16 (5) (2025) 2869–2884.
[36] S. Xia, H. Zhang, W. Li, G. Wang, E. Giem, Z. Chen, GBNRS: a novel rough set algorithm for fast adaptive attribute reduction in classification, IEEE Trans. Knowl. Data Eng. 34 (3) (2020) 1231–1242.
[37] J.C. Bezdek, N.R. Pal, Some new indexes of cluster validity, IEEE Trans. Syst. Man Cybern. Part B (Cybernetics) 28 (3) (1998) 301–315.
[38] H. Yu, Y. Cheng, Theoretical study on the upper bound of the number of clusters, Sci. China Series F: Inf. Sci. 44 (3) (2001) 198–207.
[39] D. Cheng, Y. Li, S. Xia, G. Wang, J. Huang, S. Zhang, A fast granular-ball-based density peaks clustering algorithm for large-scale data, IEEE Trans. Neural Netw. Learn. Syst. 35 (12) (2023) 17202–17215.
[40] 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.
[41] F.T. Liu, K.M. Ting, Z. Zhou, Isolation-based anomaly detection, ACM Trans. Knowl. Discov. Data (TKDD) 6 (1) (2012) 1–39.
[42] H. Xu, G. Pang, Y. Wang, Y. Wang, Deep isolation forest for anomaly detection, IEEE Trans. Knowl. Data Eng. 35 (12) (2023) 12591–12604.
[43] 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.
[44] J. Tang, Z. Chen, A.W. Fu, D.W. Cheung, Enhancing effectiveness of outlier detections for low density patterns, in: Advances in Knowledge Discovery and Data Mining: 6th Pacific-Asia Conference, PAKDD 2002 Taipei, Taiwan, May 6–8, 2002 Proceedings 6, Springer, 2002, pp. 535–548.
[45] J. Wan, H. Chen, T. Li, W. Huang, M. Li, C. Luo, R2CI: information theoretic-guided feature selection with multiple correlations, Pattern Recognit. 127 (2022) 108603.