ISSN: 2664-9640

Fast Outlier Detection Algorithm Based on Local Density and Connectivity

Ziming Guo[1, a, *]

[1]School of Logistics Engineering College, Shanghai Maritime University, Shanghai 201306, China.

[a]Correspondence: 18018980232@163.com

Abstract

The dissimilarity measurement and outlier factor calculation based on the k-nearest neighbor field have always been important research directions in outlier detection algorithms. The existing k-neighbor-based density outlier detection algorithm LOF is between the point to be measured and the k-nearest neighbor. The outlier factor of this point is calculated by the density comparison, but the method is prone to misjudgment of boundary points when the detection data is irregularly distributed. Although the INFLO algorithm improves this situation by introducing inverse k-nearest neighbors, this method not only increases the amount of calculation but also does not improve the situation where the LOF is sensitive to the value of k. When the value of k changes, the detection result of INFLO may be biased . To solve this problem, an outlier detection algorithm based on local density and connectivity (LDCO) is proposed. By combining the concept of mutual nearest neighbor points, the correlation between the various neighbor points and the degree of outlier of the data points is analyzed , and a new pruning strategy is proposed. The dissimilarity measure based on connectivity is introduced to improve the algorithm's ability to handle irregular data sets. The theoretical analysis and experimental results prove that the accuracy and efficiency of the LDCO algorithm is higher than LOF and INFLO.

Keywords

Outlier detection, Mutual nearest neighbors, Pruning, Connectivity, LDCO.

1. Introduction

Outlier detection is an important research direction of data mining. It originates from the "noise" of preprocessing in data mining. With the rapid development of the Internet and information technology, these "noise" that we previously ignored may record some anomalies. The behavior also contains a lot of valuable information, such as: machine failure reasons, network attack information, credit card fraud information, pre-disease symptoms, etc. Therefore, it is possible to find outliers in the data set that may be generated by abnormal mechanisms containing useful information Become an important task in the data mining process.

The purpose of outlier detection is to find a small amount of data in the data set that is different from most of the data. The definition of outliers that are now generally accepted is given by Hawkins: "Outliers are those that are different from most of the Obvious differences make it suspect that they are produced by different mechanisms[1]. " Through the definition of Hawkins, we can understand that in outlier detection algorithms, outliers are generally considered to have two points in common: 1. Smallness: That is, the number of outliers only accounts for a small part of the entire data set, and is generally considered not to exceed About

ISSN: 2664-9640

5%; 2. Isolation: that is, most of the outliers are far from the group or deviate from the surrounding groups[2].

With the development of outlier detection, a large number of outlier detection algorithms have been proposed. The distance-based detection method was first proposed by Knorr and Ng in 1998[3], this method detects outliers by calculating the distance between all objects and their nearest neighbors. Johnson et al. Proposed a depth-based algorithm[4], Breunig et al.'S Local Outlier Factor Algorithm (LOF)[5], Yu et al. Proposed an outlier mining algorithm (LIC) based on local isolation coefficients[6]. Among them, density-based outlier detection is an important direction for outlier detection research.

The density-based method is to overcome the problems caused by the distance-based method when facing different density clusters. The basic idea of the density-based method is: "The density of outliers is very different from the density around them. Compare the density of and its domain density. If there is a significant difference between the densities, you can declare the object as an outlier. " The most classic density-based method is the Local Outlier Factor Algorithm (LOF) proposed by Breunig et al. Aiming at the shortcomings of LOF, Jin et al. proposed a local outlier detection algorithm based on reverse K nearest neighbors (INFLO) [7]. Although the INFLO algorithm overcomes some of the shortcomings of the LOF algorithm, it still has sensitivity to the parameter K, a large amount of calculation, and is not suitable for some special data sets [8].

In view of the above problems, this paper proposes an outlier detection algorithm based on local density and connectivity (LDCO). Combine the concept of mutual K nearest neighbors to better analyze the distribution of data points, screen out suspected outliers to achieve the effect of pruning, and thereby improve the efficiency of the algorithm. The introduction of the concept of connectivity detection can better deal with special data sets, thereby improving the scalability of the algorithm.

The rest of the paper is organized as follows. In Section2, we analyzed the problem. In Section3, we introduced the related work and the proposed algorithm. In Section4, we performed experimental verification. Finally, we conclude in Section 5.

2. Problem Analysis

2.1. Related Information

Local outlier factor algorithm (LOF) is a very classic density-based outlier detection algorithm. This algorithm analyzes the local density of the measured points and the distribution of surrounding sample points to give each data point a characterization of the data The factor of the outlier degree of the object, and then find the data points of the outlier degree top-n, and these top-n data points are the detected outlier points. Some basic definitions of the LOF algorithm are as follows:

Definition 1 (K-distance of data point p)[5] For a data set *D*, if there are data point *p* and data point *o* in the data set, the Euclidean distance between the two points is recorded as *d (p, o)*, and the data point o meets the following conditions: there are at least k data points $q \in D/q$ such that $d(p.q) \le d(p, o)$; there are at most k-1 data points $q \in D/q$ such that $d(p.q) \le d(p, o)$. Then *d (p, o)* is called the k distance of the data point *p*, and it is recorded as *k-distance (p)*.

Definition 2 (K-nearest neighborhood of data point p)[5] In the data set *D*, the *k-distance (p)* of the data point *p* in the data set *D* is known, and the set of all data points in the data set *D* whose distance *p* point does not exceed *k-distance (p)* is the *k-distance* field of *p* point, denoted as $N_k(p)$:

ISSN: 2664-9640

$$N_k(p) = \{q | q \in D, d(p,q) \le k\text{- distance}(p)\} \quad (1)$$

The selection of the "neighbors" of the test points can greatly affect the detection results. From the above, we know that the algorithm represented by LOF simply treats the k points closest to the test point as the test point. The detection field is prone to detection errors. Therefore, Jin et al. Reverse neighbors and k-nearest neighbors construct an extended detection field.

2.2. Problem Description

The LOF algorithm simply considers the k points closest to the point to be measured as "neighbors", and then uses these neighboring points to determine the degree of outlier of the point to be measured. However, the nearest "neighbors" of some data points are distributed in clusters of different densities, which can easily lead to erroneous detection results, so it is not accurate to simply use k-nearest neighbors as the field for detecting outliers. The selection of the "neighbors" of the test points can greatly affect the detection results. From the above, we know that the algorithm represented by LOF simply treats the k points closest to the point to be measured as the detection area where detection errors are prone to occur. For this reason, the INFLO algorithm proposed by Jin et al. An extended detection field is constructed to the nearest neighbors and k nearest neighbors. Compared with the LOF algorithm, INFLO not only considers the k-nearest neighbors of the measured points, but also considers the reverse neighbors of the measured points, which can better reflect the distribution around the measured points. But doing so increased the amount of calculation and did not completely solve the problem.

As shown in Figure. 1, there are two different densities of C1 and C2. When K = 5, although the Q point belongs to the C2 cluster, most of it's K neighbors belong to the C1 cluster. If you use the LOF algorithm to calculate the outlier degree of the Q point, an error will occur. Since most of the reverse K nearest neighbors of the Q point belong to the C2 cluster, the detection effect of the INFLO algorithm is improved. However, for the boundary point P in the high-density cluster C1, there are many reverse K nearest neighbors in the C2 cluster. The INFLO algorithm may produce a large deviation when calculating the outlier degree of P.

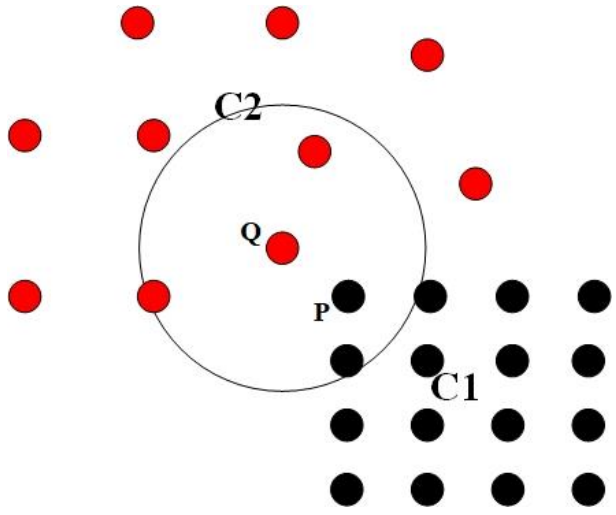

Figure 1. Examples for illustration

In addition to the above problems, LOF is not good at processing special-shaped data sets (such as: linear, crescent, etc). As shown in Figure. 2, the graph consists of a linear cluster R1, a circular cluster R2, and an outlier A. If LOF and INFLO are used as the detection methods based on local density, the outlier degree of point A may be lower than R2, which is incorrect. In

ISSN: 2664-9640

addition, both LOF and INFLO need to calculate the outlier degree of the full data set and then find outliers according to the ordering, which makes it inefficient to process high-dimensional large data.

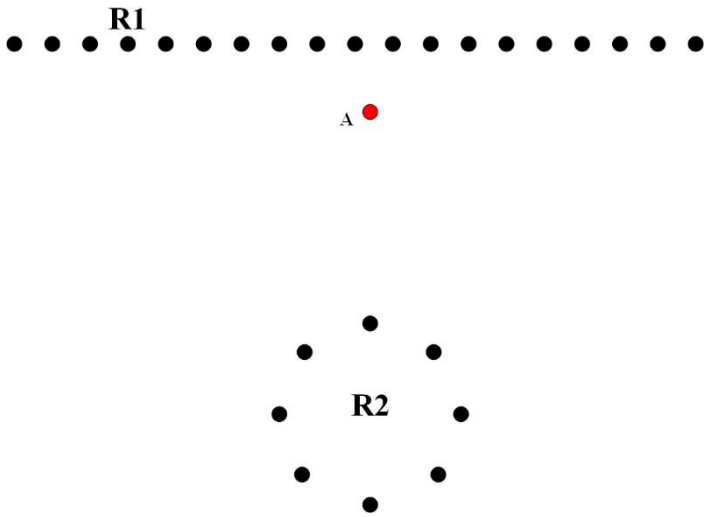

Figure 2. Failure of Outliers detection for LOF [9]

3. Relate Work

From the above analysis, we can see that the traditional density-based algorithms have problems in scalability, stability, and efficiency. Related work in this paper falls into three main categories: mutual K nearest neighbors and Connectivity-based approach, LDCO algorithm related concepts, algorithm structure.

3.1. Mutual K Nearest Neighbors and Connectivity

Definition 3 (Mutual K nearest neighbor of data point p) [10] For a data set D, if there are data point p and data point q in the data set, and $q \in N_k(p)$ $p \in N_k(q)$. Then p and q are mutual K nearest neighbor relations, Recorded as $q \in MNN_k(p), p \in MNN_k(q)$.

Mutual k-nearest neighbors are an improvement on k-nearest neighbors and inverse k-nearest neighbors. Compared to k-nearest neighbors and inverse k-nearest neighbors, mutual k-nearest neighbors are more capable of describing the distribution around points.

As shown in Figure. 3, there are two clusters and six outliers. The contour map of the number of mutual K nearest neighbors of each point in the data set is shown in Figure. 4. It can be seen that the point at the most central position of the two data clusters is used for the most mutual K-nearest neighbors, and then the smaller it goes to the edge. This shows that the number of mutual K neighbors can reflect the position of the point in the data set to some extent, which is an advantage that other neighbor relationships do not have.

The large outlier algorithm is inefficient because it often needs to calculate the degree of outliers at all points, which causes unnecessary waste. The pruning operation is to filter out the interior points of each cluster in the data set. These points must not be outliers, avoiding unnecessary calculations. The most common method is to use outlier pruning using a clustering algorithm [11]. In this way, the effect of profit group detection depends heavily on the effect of clustering, which may cause unnecessary trouble. Through the above analysis, we know that the interior points will have more mutual K neighbors, while the edge points and outliers will have fewer mutual K neighbors. Outliers in the data set generally do not exceed 5%, so we can filter out some points with fewer K-nearest neighbors. These points must not be interior points, so that the pruning effect is achieved without being affected by the clustering results.

ISSN: 2664-9640

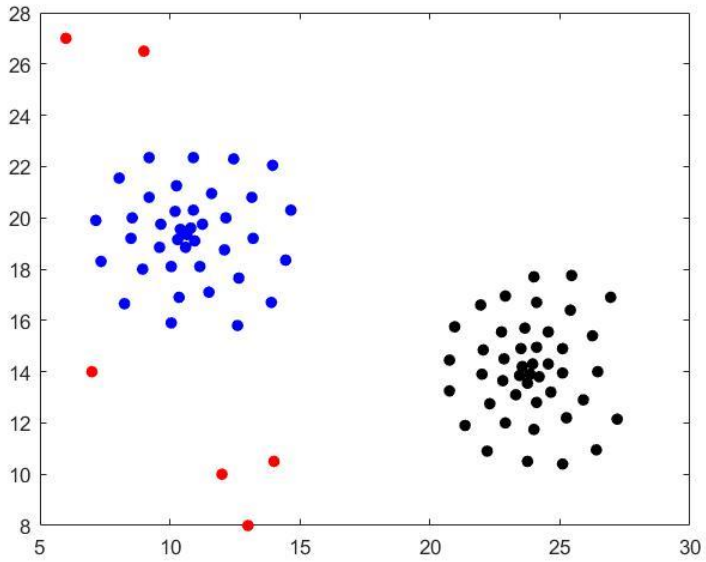

Figure 3. A sample example

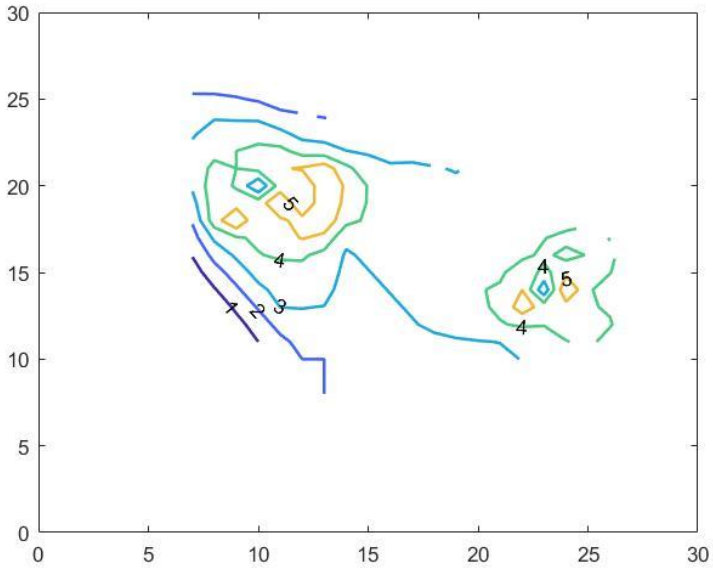

Figure 4. Contour map

Density-based outlier detection algorithms will fail when processing special data sets. In 2002, Tang et al. Proposed a connectivity-based outlier factor (COF)[9]. This method calculates outliers based on the overall connectivity of the data points to the dataset. Although the connectivity-based method improves this situation well, it does not work well when faced with datasets with large differences in density.

3.2. LDCO Algorithm

From the above, the basic idea of the density-based method is: "The density of outliers is very different from the density around them. Compare the density of the data point with its field density. If there is a significant difference between the densities, You can declare the object as an outlier". This method is suitable for data sets with a spherical distribution, and is not good at processing some data sets with other morphological distributions. Connectivity-based methods are used to investigate the degree of connection between the points and the data set

ISSN: 2664-9640

to calculate the isolation of the points, so as to filter outliers. In order to combine the advantages of the two, a new outlier factor calculation method is designed by combining density and connectivity.

The density estimation method in this paper uses the kernel density estimation method proposed by Latecki et al [12]. The density-based outlier factor is obtained by using the ratio of the kernel density of the point to be measured and its neighbor's kernel density. The neighborhood area is expanded to obtain extended density-based outliers as Equation 2.

$$\mathrm{ELDF}(p)=\frac{\sum_{q\in S_k(p)}\frac{\mathrm{LDE}(q)}{|S_k(p)|}}{\mathrm{LDE}(p)+c\sum_{q\in S_k(p)}\frac{\mathrm{LDE}(q)}{|S_k(q)|}} \tag{2}$$

$$S_k(p)=\{q|q\in N_k(p)\cup \mathrm{FNN}_k(p)\} \tag{3}$$

Where c is a scaling factor ranging from 0 to 1, $\mathrm{FNN}_k(p)$ is the inverse K nearest neighbors, and LDE represents the kernel density at each point.

The calculation method of connectivity is shown in Equation 4, among them $s=|S_k(p)|$. Estimate the connectivity-based outlier factor LCO by comparing the ratio of the measured points to the neighboring points' connectivity, as shown in Equation 5.

3.3. LDCO Algorithm Structure

The process steps of the LDCO algorithm are shown in follow.

Algorithm: outlier detection algorithm based on local density and connectivity

Input: Data set D, parameter K,m

Output: top-m Outlier according to LDCO

1: for all p∈D, finding sorted distance from p to all neighbor q.

2: for all p∈D, Get K nearest neighbors, reverse K nearest neighbors, and mutual K nearest neighbors at p

3: $\forall p \in D$, if the number of mutual K nearest neighbors of p points is 0.8K or more

4: the p-point is the interior point of the cluster, not an outlier

5: else

6; add p points to the candidate outlier set

7: end

8: for all p∈the candidate outlier set

9: Calculate LDCO(p)

10: end

11: sort all obtained LDCO values

12: get the point of LDCO value top-m

13: output m outliers

14: end function

$$\mathrm{ac}\text{-}\mathrm{dist}(p)=\sum_{i=1}^{s-1}\frac{2(s-i)}{s(s-1)}\mathrm{dist}(e_i) \tag{4}$$

ISSN: 2664-9640

$$LCO(p) = \frac{|S_k(p)| ac\text{-}dist(p)}{\sum_{q \in S_k(p)} ac\text{-}dist(q)} \tag{5}$$

Combine the above-mentioned density and connectivity-based outlier factor (LDCO) as shown in Equation 6, among them $\alpha = \frac{K}{|S_k(p)|}$.

$$LDCO(p) = \frac{\alpha \cdot ELDF(p) + (1-\alpha) LCO(p)}{2} \tag{6}$$

4. Experiment Analysis

We will verify the performance of LDCO on stability, accuracy and efficiency on an artificial data set and several UCI data sets.

4.1. Artificial Data Set

As shown in Figure. 5, the artificial data set contains 3 clusters and 1 outlier, of which 2 clusters with different densities are closer and the other is a polyline cluster.

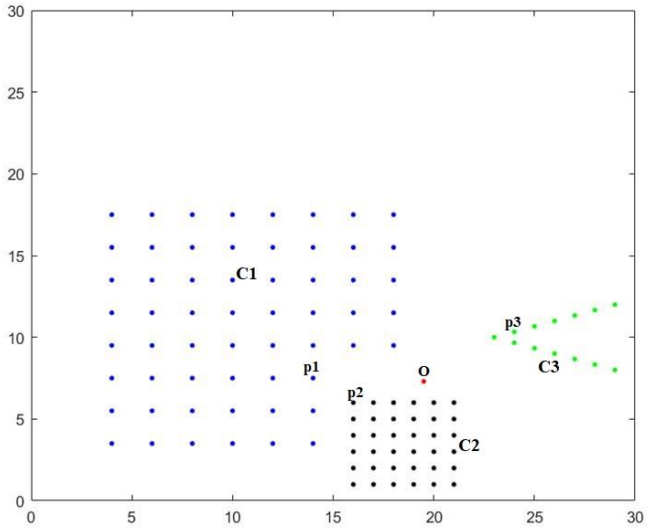

Figure 5. Artificial data set

Select one point on the contact edge of the *C1* and *C2* clusters, which is *p1* and *p2*. Take a random point *p*3 in the C3 cluster and add the outlier point o. When K = 4-12, use LOF, INFLO, LDCO to calculate the degree of outlier of these 4 points, The result is shown in Figure. 6.

As K is worth changing, the outliers calculated by the LOF and INFLO algorithms continue to fluctuate as K is worth changing. In many cases, the outlier point o is less than the normal point, which is incorrect. Compared with them, the LDCO algorithm is very stable in the face of K-worth changes, and there is no misjudgment.

ISSN: 2664-9640

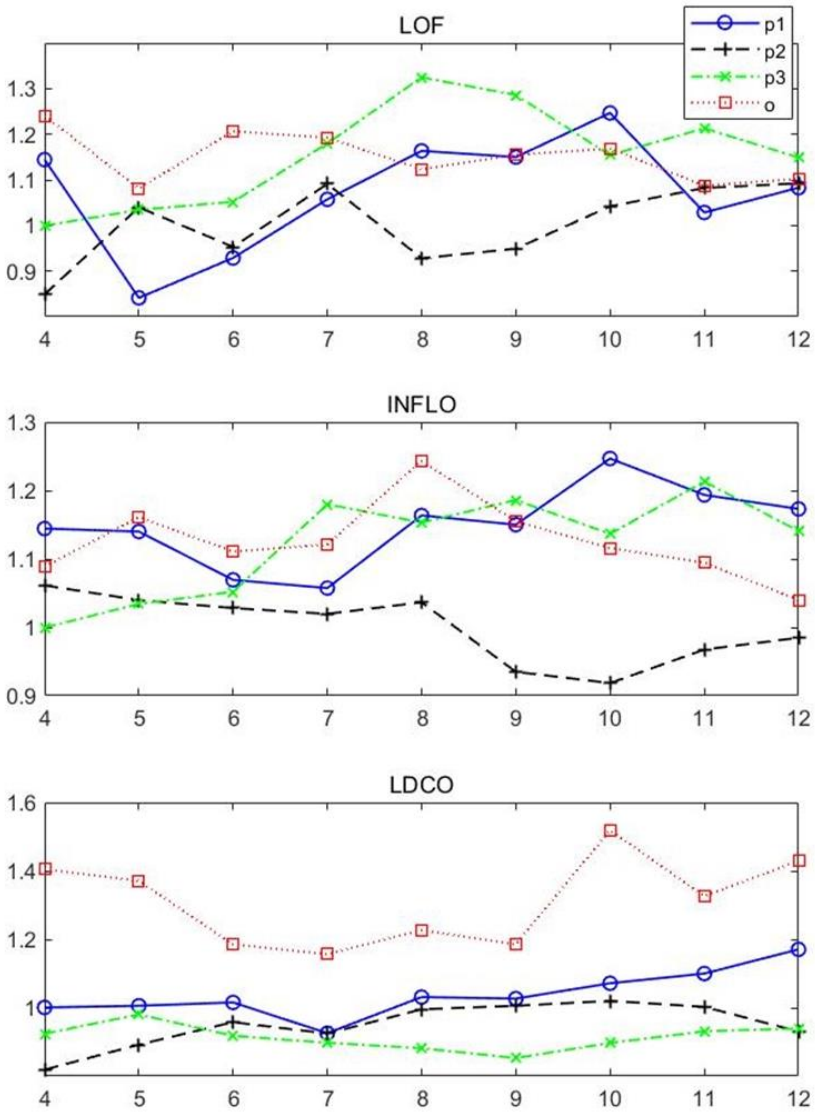

Figure 6. Compare Results

4.2. UCI Data Set

The Shuttle dataset contains 14,500 objects, each of which has 9 features and 1 label, of which about 80% of the data are of type 1 labels. From the Shuttle data set, 1000 data points were selected, of which 95% were classified as normal points and 1% were labeled as outliers. The results after using the three algorithms are as Table 1.

Table 1. Performance comparison on the Shuttle dataset with fixed m=20(K=10)

Algorithm	Pr	Re	RP	Time
LOF	0.55	0.26	0.40	1.47
INFLO	0.70	0.28	0.76	1.83
LDCO	0.95	0.38	0.99	0.71

Where m indicates that the algorithm outputs outlier top-m points. Pr represents precision, precision represents the percentage of true outliers in the top m objects returned by the method. Re means Recall, Recall is the percentage of total outliers contained in the top-m points. RP(rank power) measures the position of the outliers in the top-m points of the output, Suppose

ISSN: 2664-9640

there are n outliers in top-m, and Wi represents the position of these points, then RP is defined as:

$$RP=\frac{n(n+1)}{2\sum_{i=1}^{n} W_i} \tag{7}$$

5. Conclusion

In this paper, we propose a new density and connectivity-based outlier detection algorithm. Experimental results on artificial data sets and UCI data sets show that the algorithm is superior to the LOF and INFLO algorithms in stability, accuracy, and efficiency.

References

[1] A. C. Atkinson, D. M. Hawkins. Identification of Outliers[J]. Biometrics, 1981, 37(4):860.

[2] Enderlein G. Hawkins, D. M.: Identification of Outliers. Chapman and Hall, London – New York 1980, 188 S. £ 14, 50[J]. 2010, 29(2):198-198.

[3] Edwin M. Knorr, Raymond T. Ng, Vladimir Tucakov. Distance-based outliers: algorithms and applications[J]. Vldb Journal, 8(3-4):237-253.

[4] Johnson, T., I. Kwok and R. Ng. Fast computation of 2-dimensional depth contours. in International Conference on Knowledge Discovery & Data Mining. 1998.

[5] Breunig, M.M., et al. LOF: Identifying Density-Based Local Outliers. in Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA. 2000.

[6] Yu B , Song M , Wang L . Local Isolation Coefficient-Based Outlier Mining Algorithm[C]. International Conference on Information Technology & Computer Science. IEEE, 2009.

[7] Jin W , Tung A K H , Han J , et al. Ranking Outliers Using Symmetric Neighborhood Relationship[J]. Lecture Notes in Computer Science, 2006.

[8] Wang X , Wang X L , Ma Y , et al. A fast MST-inspired kNN-based outlier detection method[J]. Information Systems, 2015, 48:89-112.

[9] J. Tang, Z. Chen, W. C. Fu, and W. L. Cheung, “Enhancing effectiveness of outlier detections for low density patterns,” 2002.

[10] Divya Sardana, Raj Bhatnagar. Graph Clustering Using Mutual K-Nearest Neighbors[C], International Conference on Active Media Technology. Springer International Publishing, 2014.

[11] T. jin, “Outlier detection method based on clustering and density(in chinese),” Ph.D. dissertation, 2014.

[12] Latecki L J , Lazarevic A , Pokrajac D . Outlier Detection with Kernel Density Functions[C] International Workshop on Machine Learning and Data Mining in Pattern Recognition. Springer, Berlin, Heidelberg, 2007.

[13] Yang Maolin, Lu Yansheng. Mining outliers for massive data based on pruning [J](in chinese). Journal of Frontiers of Computer Science, 2012, 39 (10): 152-156.

[14] Zhu Q , Feng J , Huang J . Natural Neighbor: a self-adaptive neighborhood method without parameter K[J]. Pattern Recognition Letters, 2016, 80.

[15] Tang B , He H . ENN: Extended Nearest Neighbor Method for Pattern Recognition [Research Frontier][J]. Computational Intelligence Magazine, IEEE, 2015, 10(3):52-60.

[16] Angiulli F , Basta S , Lodi S , et al. Distributed Strategies for Mining Outliers in Large Data Sets[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(7):1520-1532.

[17] Amol Ghoting, Srinivasan Parthasarathy, Matthew Eric Otey. Fast mining of distance-based outliers in high-dimensional datasets[J]. Data Mining & Knowledge Discovery, 16(3):349-364.

ISSN: 2664-9640

[18] Albert J. Hoglund, Kimmo Hatonen, A.S. Sorvari. A computer host-based user anomaly detection system using the self-organizing map[C], Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on. IEEE, 2000.

[19] Wang, Wei, Lu, Peizhong. An Efficient Switching Median Filter Based on Local Outlier Factor[J]. IEEE Signal Processing Letters, 18(10):551-554.

[20] Dragoljub Pokrajac, Natasa Reljin, Nebojsa Pejcic. Incremental Connectivity-Based Outlier Factor Algorithm[C], Visions of Computer Science - BCS International Academic Conference, Imperial College, London, UK, 22-24 September 2008. DBLP, 2008.

[21] Tang B , He H . A local density-based approach for outlier detection[J]. Neurocomputing, 2017, 241:171-180.

[22] Zhang, Ke, Hutter, Marcus, Jin, Huidong. A New Local Distance-Based Outlier Detection Approach for Scattered Real-World Data[C], Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Berlin, Heidelberg, 2009.

[23] Bai, Mei, Wang, Xite, Xin, Junchang. An efficient algorithm for distributed density-based outlier detection on big data[J]. Neurocomputing, 181:19-28.

[24] Pei Y, Zaane O R, Gao Y. An efficient reference-based approach to outlier detection in large datasets[J]. 2006.

[25] Garima Singh V K. An Efficient Clustering and Distance Based, Approach for Outlier Detection[J]. 2013, 4(7).

[26] Outlier detection using neighborhood rank difference[J]. Pattern Recognition Letters, 2015, 60-61:24-31.

[27] S.S. Sabade, D.M. Walker. IC Outlier Identification Using Multiple Test Metrics[J]. IEEE Design & Test of Computers, 2005, 22(6):586-595.