*International Journal of Innovative Computing, Information and Control*
Volume **14**, Number **5**, October **2018**

ICIC International ©2018 ISSN 1349-4198
pp. **1633–1651**

# OUTLIER DETECTION WITH ENHANCED ANGLE-BASED OUTLIER FACTOR IN HIGH-DIMENSIONAL DATA STREAM

Zhaoyu Shou[1], Hao Tian[1], Simin Li[2] and Fengbo Zou[1]

[1]School of Information and Communication Engineering
Guilin University of Electronic Technology
[2]Key Laboratory of Cognitive Radio and Information Processing Ministry of Education
No. 1, Jinji Road, Guilin 541004, P. R. China
{ guilinshou; siminl }@guet.edu.cn; { marcus_tian; zfbb.cool }@163.com

Received January 2018; revised May 2018

Abstract. *Outlier detection over data stream is an increasingly important research in many fields. Traditional methods are no longer applicable. In this paper, a novel outlier detection algorithm with enhanced angle-based outlier factor in high-dimensional data stream (EAOF-OD) is proposed. EAOF-OD aims at improving the performance of outlier detection and reducing the consumption of memory. To measure the deviation degree of potential outliers accurately in sophisticated high-dimensional datasets, an enhanced angle-based outlier factor is introduced. To ensure the high detection rate, the proposed scheme first locates the cluster centers and divides the dataset into several clusters, and then outlier detection is carried out within each cluster. Furthermore, an efficient model based on sliding window and multiple validations is presented in order to decrease the false alarm rate, which divides data stream into uniform-sized blocks and declares a point far away from its cluster as candidate outlier. With new block joining in and historical block moving out, the sliding window reserves the most valuable information including candidate outliers which need multiple validations. Comparison experiments with existing approaches on synthetic and real datasets demonstrate that EAOF-OD outperforms some existing approaches in terms of outlier detection rate and false alarm rate.*
**Keywords:** Outlier detection, Data stream, Enhanced angle-based outlier factor (EAOF), Sliding window, Multiple validations

1. **Introduction.** Outlier detection is to quickly detect abnormal objects that do not meet the expected behavior from the complex data environment, providing deep analysis and understanding for users [1]. Outliers are usually generated by unusual mechanism, which often contain valuable information. Hence, detecting outliers from complex data environment shows great scientific and engineering importance. With the rapid development of network technology and growing popularity of society informatization, the amount of information keeps on increasing explosively. Many fields are generating high-speed, infinite and dynamic data streams. Outlier detection has been applied to many domains such as medical treatment [2], network intrusion detection [3-5], business transaction management and analysis [6,7], video surveillance [8,9], and sensor networks [10]. However, as data stream evolves during the time, traditional methods cannot perform well on them, and an outlier detection algorithm that is applied to dynamic data stream well becomes necessary.

The study of this paper aims at enhancing the outlier detection rate and decreasing the false alarm rate of data stream. The proposed algorithm EAOF-OD introduces a good way to handle the continually increasing amount of data, even in memory limited

DOI: 10.24507/ijicic.14.05.1633

situation. In real life the storage device cannot store all the increasing information forever, so the model based on sliding window and multiple validations of EAOF-OD will be useful in many applications. The outlier factor EAOF has a wide applicability in sophisticated circumstance, and the outlier estimation criterion based on mean and standard deviation is also applicable in data analysis and data mining. EAOF-OD has good performance in practical applications such as medical treatment and network intrusion detection, which will be described in experiment of Section 7.

The rest of this paper is organized as follow. Section 2 discusses related work. Section 3 presents the inspirations of EAOF-OD. Section 4 introduces related definitions. Section 5 shows the model based on sliding window and multiple validations. Section 6 describes the detail of EAOF-OD. Section 7 presents the experimental results. Finally, Section 8 concludes the paper.

2. **Related Work.** Most existing outlier detection algorithms in data stream can be categorized into four groups: distance-based algorithms; density-based algorithms; angle-based algorithms; clustering-based algorithms.

In distance-based algorithms, distance shows mutual relationship between objects. See [11] for distance-based algorithm, an object will be reported as an outlier if it does not have enough neighbors (objects within a specified distance).

Density-based algorithms [12-17] use density to evaluate the outlierness degree of each potential outlier, and update the outlier factors of each data dynamically. LOF (local outlier factor) [12] is a popular density-based algorithm used in static dataset. IncLOF (incremental LOF) [13] applies LOF iteratively after insertion of each new data. N-IncLOF [14] and I-IncLOF (improved IncLOF) [15] introduce sliding window to cut down the consumption of memory resource. Enlightened by LOCI (local correlation integral) [16], INCLOCI (incremental LOCI) algorithm [17] calculates the high-granularity deviation factor of each data to detect outliers.

To deal with the problem of dimension disaster, angle-based algorithms [18,19] introduce angle-based outlier factor (the variance of angles formed by a target object and all pairs of other objects) to measure the deviation degree of each data more precisely in high-dimensional dataset. In [18], ABOD (angle-based outlier detection) is proposed to detect outliers in static dataset. Based on ABOD, DSABOD (data stream angle-based outlier detection) algorithm [19] is presented to detect outliers on high-dimensional data stream. DSABOD updates ABOF (angle-based outlier factor) of each data and declares those data with high ABOF value as outliers. Angles are more stable than many other measurements in high-dimensional space, and the angle-based algorithm provides a new perspective to work out the estimation of outlierness degree of objects.

In clustering-based algorithms [20-30], outliers are those objects which do not belong to any cluster or deviate far away from the most objects in their clusters. Many traditional clustering algorithms such as DBSCAN (density-based spatial clustering of applications with noise) [20] work well on static datasets. Some researchers proposed many clustering algorithms [21-28] over data stream, which show good performance. Clustering is the first and important step of outlier detection in clustering-based outlier detection algorithms, and it directly affects the result of outlier detection. In [29], an unsupervised outlier detection algorithm based on weighted clustering (denoted as Algorithm Y in shorthand in the following part) is proposed, which divides the data stream into blocks. Algorithm Y clusters each block and detects outliers in each block. In clustering part, it combines DBSCAN and W-K-Mean (weighted-K-Mean clustering) [30], and updates the parameters needed. Algorithm Y is accurate but tedious. In outlier detection part, it treats small clusters as outlier groups and determines the scattered outliers based on distance.

3. **Inspirations.** Being influenced by the dimension disaster, traditional ways of measuring outlierness of data based on distance or density lose effectiveness as the dimension increases, resulting in bad performance of outlier detection. According to ABOD algorithm and DSABOD algorithm, the variance of angles can be used to evaluate the deviation degree of each object, the variance of angles formed by an outlier object and all pairs of other objects is quite small, while the variance of angles formed by an inlier object and all pairs of other objects is pretty large, which has been proved to be an effective way in high-dimensional space.

However, there are some shortcomings of traditional angle-based measurement as shown in Figure 1 and Figure 2. In Figure 1, though $E_1$ is in the center of a u-shaped cluster, and $E_2$ is located on the edge which shows less anomalous than $E_1$, the variance of angles formed by $E_1$ and pairs of other objects is rather larger than that formed by $E_2$ and others. Similar situation occurs in Figure 2 with $D_1$ located between two clusters which is more like an outlier, and $D_2$ is on the edge of one cluster, the variance of angles formed by $D_1$ and pairs of other objects is larger than that formed by $D_2$ and others.

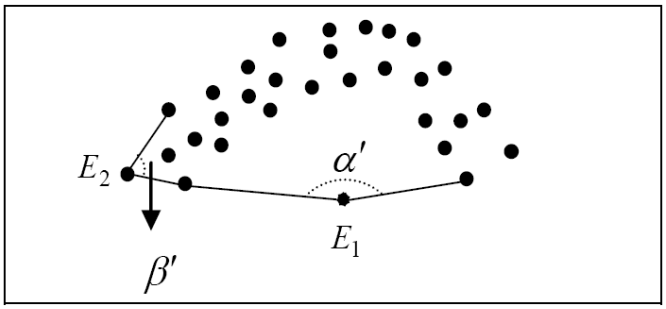


FIGURE 1. Exceptional situation 1

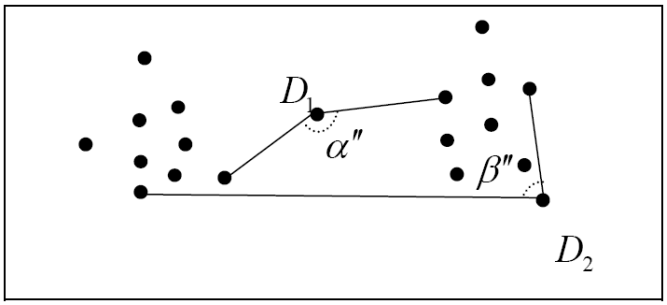


FIGURE 2. Exceptional situation 2

All of the above cases do not conform to the traditional theory of angle-based measurement. In order to improve the effectiveness of angle-based measurement, an enhanced angle-based outlier factor (EAOF) is presented which combines the advantages of distance-based measurement, density-based measurement and angle-based measurement.

An algorithm of outlier detection with enhanced angle-based outlier factor in high-dimensional data stream (EAOF-OD) is proposed in this paper. Taking advantage of the sliding window, EAOF-OD presents an efficient model to reduce the consumption of limited memory. To evaluate the outlierness of data more accurately, an enhanced angle-based outlier factor is introduced. The outliers can be determined by an efficient criterion based on mean and standard deviation which is introduced in Section 6.1.

4. **Related Definitions.** EAOF-OD has the following definitions.

**Definition 4.1.** *(Angle-based outlier factor)*

*Given a d-dimensional dataset $S^d$, a point $\vec{A} \in S^d$ $\left(\vec{A} = (A_1, A_2, \ldots, A_d)\right)$. For two points $\vec{B}$, $\vec{C} \in S^d$ and $\vec{B}$, $\vec{C} \neq \vec{A}$, $\overline{AB} = \vec{B} - \vec{A}$ denotes the difference vector, $\left\|\overline{AB}\right\|$ represents the Euclidean distance between $\vec{A}$ and $\vec{B}$, and $\left\langle \overline{AB}, \overline{AC} \right\rangle$ signifies the scalar product between $\overline{AB}$ and $\overline{AC}$. The angle-based outlier factor $V\left(\vec{A}\right)$ is the variance over the angles between the difference vectors of $\vec{A}$ to all pairs of points in $S^d$ weighted by the distance of the points:*

$$\begin{aligned} V\left(\vec{A}\right) &= VAR_{\vec{B},\vec{C}\in S^d}\left(\frac{\left\langle \overline{AB}, \overline{AC} \right\rangle}{\left\|\overline{AB}\right\|^2 \cdot \left\|\overline{AC}\right\|^2}\right) \\ &= \frac{\sum\limits_{\vec{B}\in S^d}\sum\limits_{\vec{C}\in S^d} \frac{1}{\left\|\overline{AB}\right\|\cdot\left\|\overline{AC}\right\|} \cdot \left(\frac{\left\langle \overline{AB}, \overline{AC} \right\rangle}{\left\|\overline{AB}\right\|^2\cdot\left\|\overline{AC}\right\|^2}\right)^2}{\sum\limits_{\vec{B}\in S^d}\sum\limits_{\vec{C}\in S^d} \frac{1}{\left\|\overline{AB}\right\|\cdot\left\|\overline{AC}\right\|}} \\ &\quad - \left(\frac{\sum\limits_{\vec{B}\in S^d}\sum\limits_{\vec{C}\in S^d} \frac{1}{\left\|\overline{AB}\right\|\cdot\left\|\overline{AC}\right\|} \cdot \frac{\left\langle \overline{AB}, \overline{AC} \right\rangle}{\left\|\overline{AB}\right\|^2\cdot\left\|\overline{AC}\right\|^2}}{\sum\limits_{\vec{B}\in S^d}\sum\limits_{\vec{C}\in S^d} \frac{1}{\left\|\overline{AB}\right\|\cdot\left\|\overline{AC}\right\|}}\right)^2 \end{aligned} \tag{1}$$

**Definition 4.2.** *(Local density)*

*Assuming that $N_r(p)$ is the neighbor dataset of p, where points lay within a distance of r to the center p. $V_{N_r}(p)$ is the angle-based outlier factor constructed by p and points in $N_r(p)$. The local density of p is defined below:*

$$\rho(p) = \sum_{q\in N_r(p)} e^{-\left(V_{N_r}(q)\right)^2} \tag{2}$$

From Equation (2), the local density of a point is associated with its position and the number of its nearest neighbors. A point can get larger local density when it is nearer to the center and has more neighbors.

**Definition 4.3.** *(Cluster dissimilarity)*

*According to Equation (2), local density of each point can be acquired, and listed in descending order as $\rho(p_1) \geq \rho(p_2) \geq \cdots \geq \rho(p_n)$, $\{p_i\}_{i=1}^n$ are the corresponding sequence numbers of points in dataset, and n is the total number of points. $d(p_i, p_j)$ denotes the Euclidean distance between $p_i$ and $p_j$. The cluster dissimilarity $\delta(p_i)$ is the distance between $p_i$ and the nearest point among those points with higher local density than $p_i$:*

$$\delta(p_i) = \begin{cases} \min\limits_{p_j(j<i)} \left(d(p_i, p_j)\right), & i \geq 2 \\ \max\limits_{j\geq 2} \left(\delta(p_j)\right), & i = 1 \end{cases} \tag{3}$$

**Definition 4.4.** *(Cluster centrality factor)*

*With local density and cluster dissimilarity, the cluster centrality factor $\tau(p)$ can be definite as:*

$$\tau(p) = \rho(p) \cdot \delta(p) \tag{4}$$

The cluster centrality factor is applied to evaluating the centrality degree of a point. The bigger cluster centrality factor is, the more likely the point is located in the center of a cluster. Figure 3, Figure 4 and Figure 5 show how the cluster centrality factor works. Figure 3 shows the distribution of a dataset, obviously there are two clusters, and point 13 and point 25 are the centers.

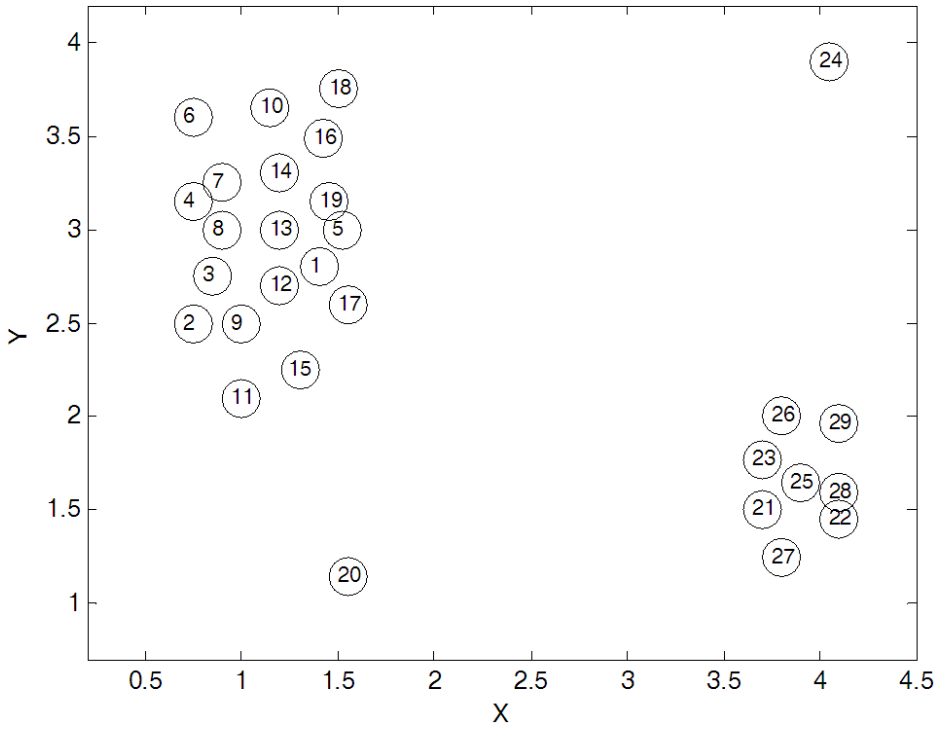


FIGURE 3. Distribution of a dataset

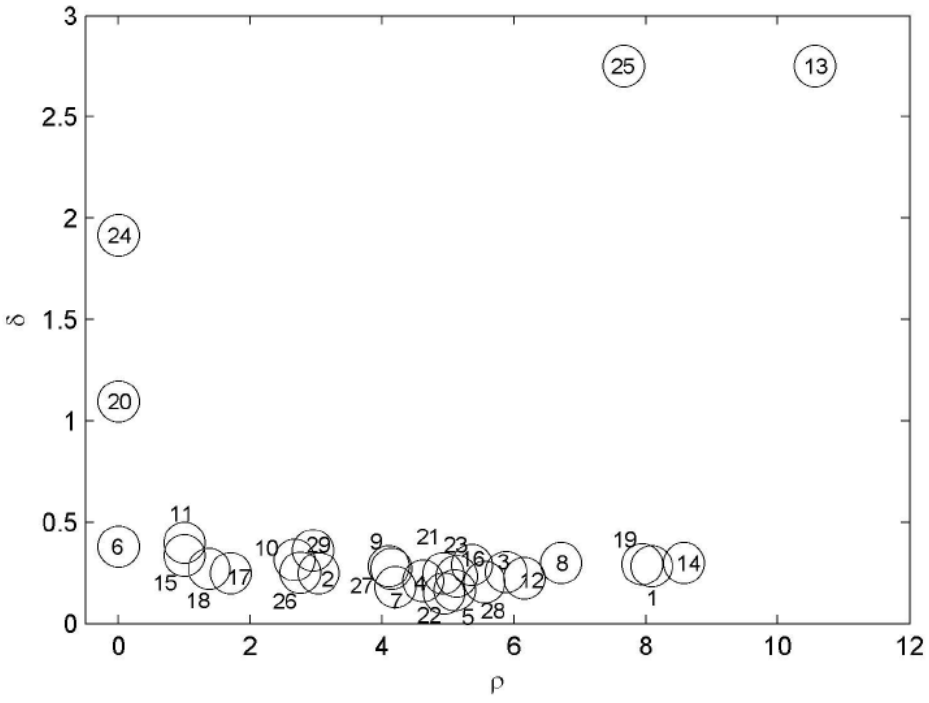


FIGURE 4. Distribution of $\rho$-$\delta$

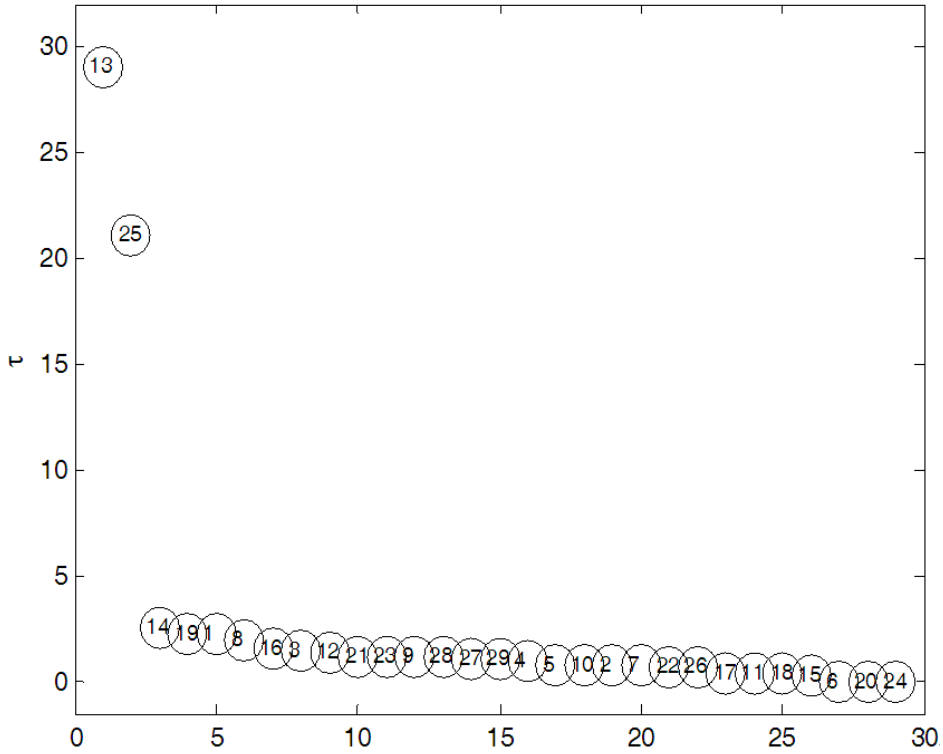


FIGURE 5. The cluster centrality factors in descending order

The information about local density ($\rho$) and cluster dissimilarity ($\delta$) of each point is shown in Figure 4. It can be observed that both $\rho$ and $\delta$ of point 13 and point 25 are much larger than the other points. Figure 5 shows the cluster centrality factors of all points in descending order. Standing out from the rest points, point 13 and point 25 have the largest cluster centrality factors, which apparently should be identified as cluster centers. Therefore, cluster centrality factor is an efficient way to find cluster centers, which is a crucial step of clustering.

**Definition 4.5.** *(Belonging matrix)*

*Belonging matrix is employed to record the belonging relationships among all points. Belonging matrix is expressed as* $F = [f_1, f_2, \ldots, f_n]$, *where* $f_i$ $(i = 1, 2, \ldots, n)$ *is the sequence number of the point which is the nearest point of point* $i$ *among those points with larger cluster centrality factors than point* $i$. *List cluster centrality factors as* $\tau(q_1) \geq \tau(q_2) \geq \cdots \geq \tau(q_n)$, *where* $\{q_j\}_{j=1}^{n}$ *are the corresponding sequence numbers of points in dataset.* $f_{q_j}$ $(j = 1, 2, \ldots, n)$ *is the* $q_j$*th element in the belonging matrix, which can be derived by the method below:*

$$f_{q_j} = \begin{cases} q_1, & j = 1 \\ \{q_i | dist\,(q_i, q_j) = \min\{dist\,(q_i, q_j), i = 1, 2, \ldots, j-1\}\}, & else \end{cases} \tag{5}$$

**Definition 4.6.** *(Local increment)*

*Assuming that* $S^d$ *is divided into* $m$ *clusters* $C_1, C_2, \ldots, C_m$ *after clustering.* $N_{r_c}(p)$ *denotes the subset formed by points which belong to the same cluster as* $p$ *and located in its* $r$*-neighborhood with* $p$ *as its center and* $r$ *as its radius.* $U_{N_{r_c}(p)}(p)$ *represents the number of points in* $N_{r_c}(p)$. *The local increment* $H(p)$ *can be calculated as:*

$$H(p) = \sum_{q \in N_{r_c}(p)} U_{N_{r_c}(q)}(q) \tag{6}$$

**Definition 4.7.** *(Sum of distances between k nearest neighbors)*

*For point p, assuming that the k neighborhood formed by its k nearest neighbors is denoted as* $N_{k\text{-}distance(p)}(p)$*, then the sum of distances between p and its k nearest neighbors is measured by:*

$$L(p) = \sum_{q \in N_{k\text{-}distance(p)}(p)} dist(p, q) \tag{7}$$

**Definition 4.8.** *(EAOF: enhanced angle-based outlier factor)*

*Assuming that o is the center of cluster to which p belongs,* $dist(o, p)$ *is the distance between point p and the cluster center* $V_C(p)$ *is the angle-based outlier factor calculated with points in* $N_{rc}(p)$*. The enhanced angle-based outlier factor* $EAOF(p)$ *is defined as:*

$$EAOF(p) = \frac{dist(o, p) \cdot L(p)}{V_C(p) \cdot H(p)} \tag{8}$$

The enhanced angle-based outlier factor not only remains the outstanding performance of traditional angle-based outlier factor in high-dimensional space, but also combines the advantages of distance-based measurement and density-based measurement, which greatly improves the accuracy of estimating deviation degree of point in complex data environment.

5. **Model Based on Sliding Window and Multiple Validations.** Data stream consists of a series of data, which is infinite, dynamic and arrives continuously. Due to the limited memory resource, reserving all the information of the data stream not only is impossible, but also increases the time and space complexity. To work out the problem, an efficient model based on sliding window and multiple validations is presented in this part. The coming data stream is divided into uniform-sized blocks. Several blocks form a sliding window. Different from the traditional sliding window [14,15] which moves from point to point and keeps a constant width. The sliding window constructed in this paper moves from data block to data block and its width may change a little depending on conditions. Data stream is being loaded into memory with new block joining in and historical block moving out and the sliding window only reserves the valuable data. Due to the dynamic nature of data stream, data behavior may change during the time. As shown in Figure 6(a) and Figure 6(b), at the time of $t_1$, $P'$ shows up like an outlier. While as the sliding window moves and new data block loaded in, $P'$ belongs to a new dense cluster at $t_2$. So evaluating an object for outlierness when it arrives may lead to wrong decisions.

Multiple validations are employed in this framework. Declare those new coming data which deviate far away from the most other data as candidate outliers, reserve them in

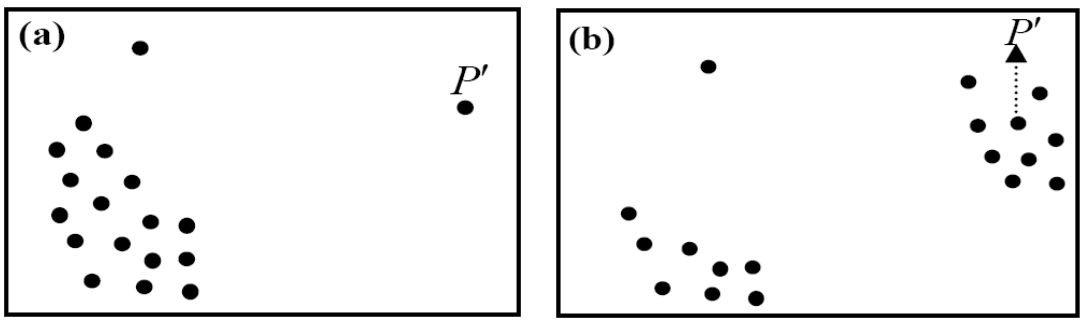


FIGURE 6. Data distribution in sliding window: (a) data distribution in sliding window at the time of $t_1$; (b) data distribution in sliding window at the time of $t_2$

the sliding window and examine their outlierness when the following blocks move in. If candidate outliers remain anomalous after specified times of multiple validations, declare them as real outliers, otherwise remove them from memory as normal data.

Figure 7 provides an insight about the way how the efficient model based on the sliding window and multiple validations works. $B_0, B_1, B_2, \ldots$ are the data blocks divided, $\varepsilon$ ($\varepsilon = 2$ in Figure 7) blocks form a sliding window. As the sliding window $W_i$ at the time of $T_i$ moves to sliding window $W_{i+1}$ at the time of $T_{i+1}$, block $B_{i+1}$ joins in and the historical block $B_{i-1}$ moves out. At the same time, the candidate outliers in $W_i$ validated at the time of $T_i$ are kept in the sliding window for the next validation.

The block diagram of EAOF-OD is given in Figure 8. It is shown in Figure 8. Fast coming data stream is divided into uniform-sized blocks, and $\varepsilon$ ($\varepsilon = 2$ in Figure 8) blocks form a sliding window. Outlier detection is carried out on the sliding window. Update the times of multiple validations and the times of being declared to be candidate outlier of

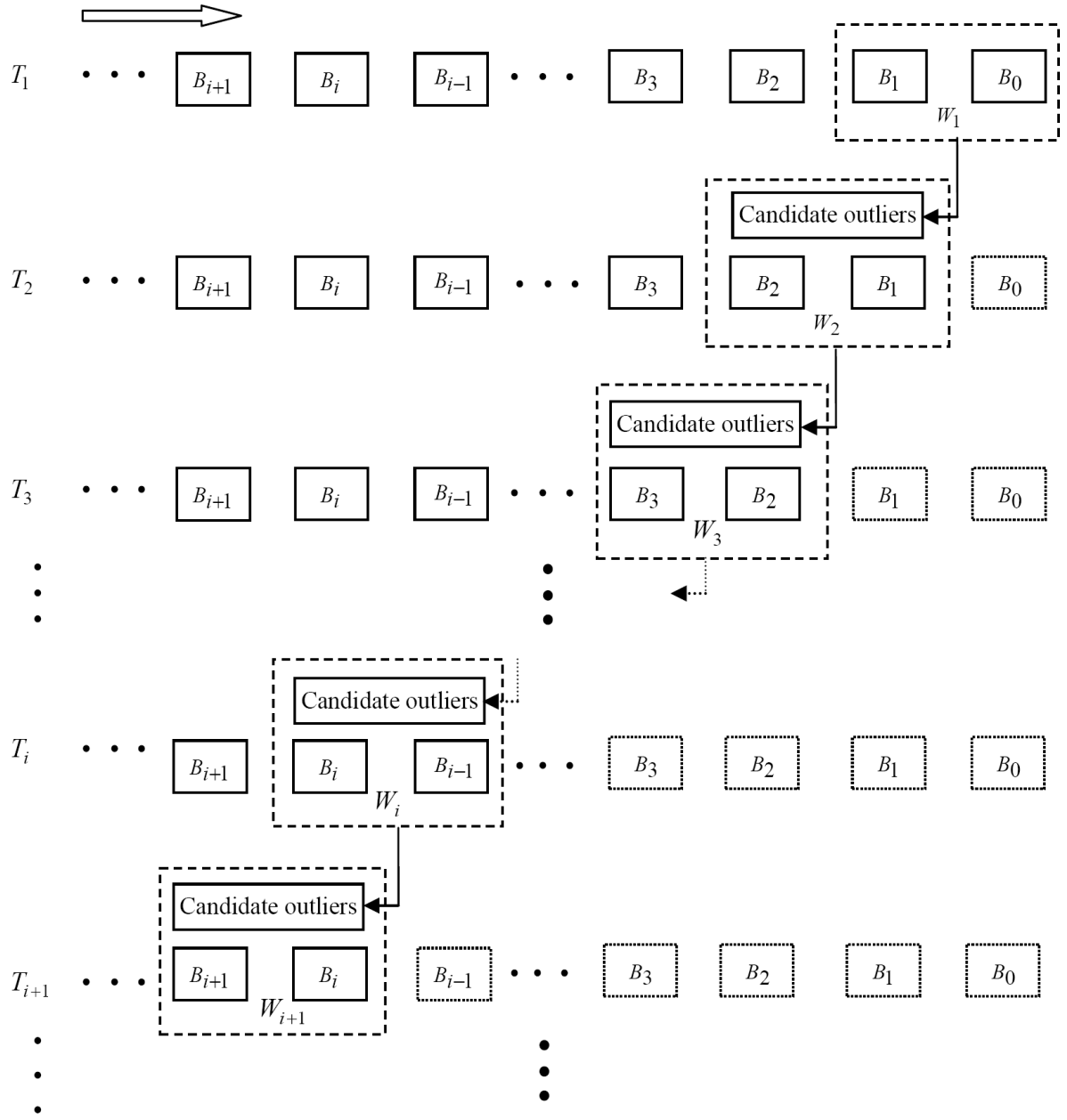


FIGURE 7. Model based on sliding window and multiple validations

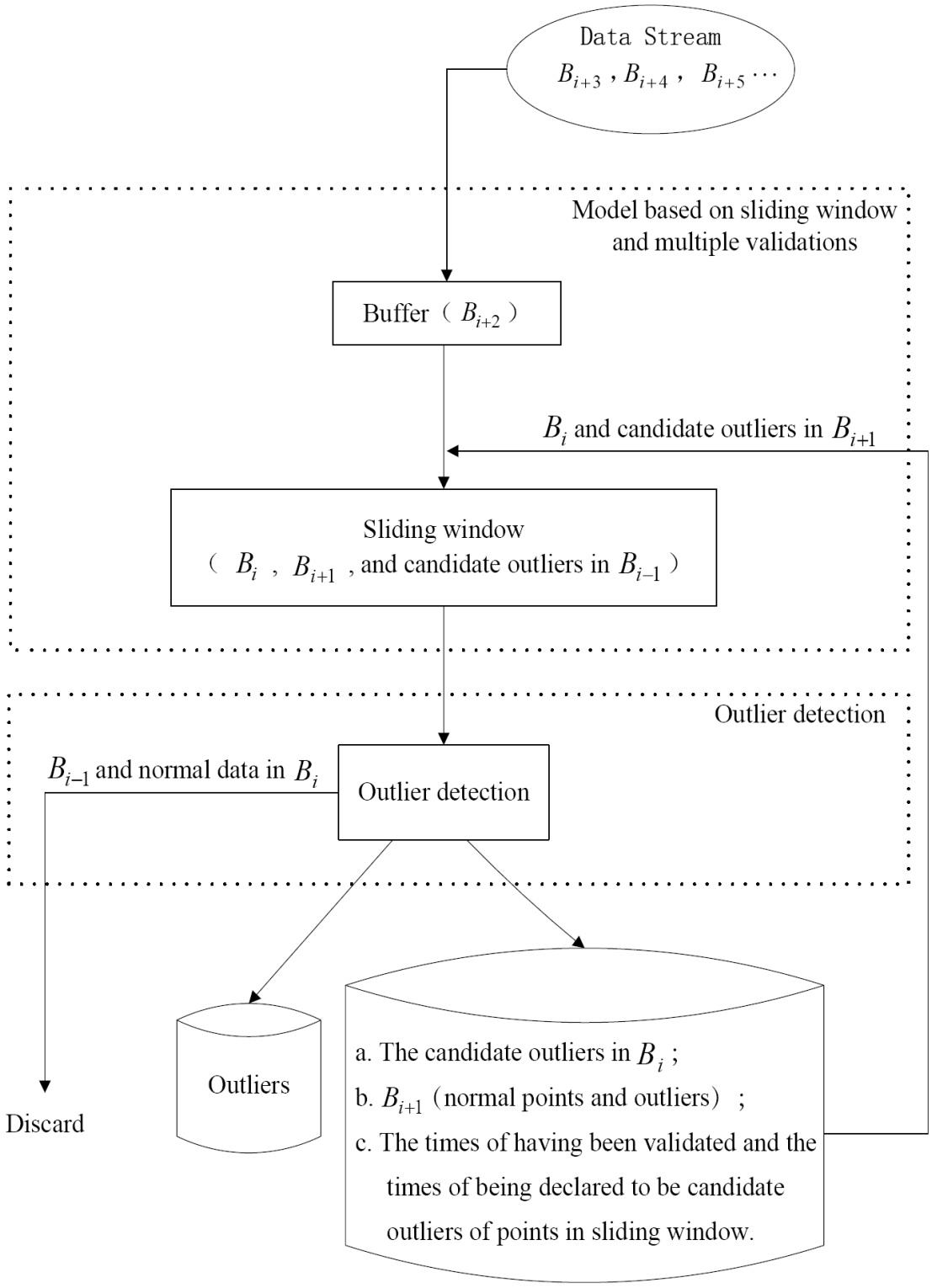


FIGURE 8. Block diagram of EAOF-OD

each point in the sliding window. And declare the points as real outliers if they meet the condition of multiple validations (described in Algorithm 1 in Section 6.2). The candidate outliers in the historical blocks which do not satisfy the conditions of being discarded are kept in the sliding window, waiting for next validation. This framework can not only deal with data stream effectively but also get high detection rate and low false alarm rate.

## 6. Outlier Detection.

### 6.1. Outlier estimation criterion based on mean and standard deviation.

**Corollary 6.1.** *Let $\mu$ denote the mean of the angle-based outlier factors of all points in the same cluster as point p, $\sigma$ is the standard deviation, and $\xi$ is a specified coefficient. Point p is declared to be an outlier if $EAOF(p)$ meets the following condition:*

$$EAOF(p) \geq \mu + \xi \cdot \sigma \tag{9}$$

**Proof:** According to Chebyshev's inequality it follows that:

$$\Pr\{EAOF(p) - \mu \geq \xi \cdot \sigma\} \leq \Pr\{|EAOF(p) - \mu| \geq \xi \cdot \sigma\} \leq \frac{\sigma^2}{(\xi \cdot \sigma)^2} = \frac{1}{\xi^2}$$

where Pr{ } denotes the probability. It is proved that the Chebyshev's inequality holds in any case. It can be concluded that, in a cluster the possibility that the EAOF of a point is bigger than the sum of mean and $\xi$ times the value of standard deviation is no more than $1/\xi^2$, that is rare event. Note that, in real life, the frequency of anomalous patterns is rare (ranging from 5% to less than 0.01% depending on the application) [13,15]. If the cluster satisfies Gauss distribution, the value of $\xi$ can be set to 3. For other distributions, in order to reduce the false alarm rate, $\xi$ is suggested to be set to $2 \sim 3$ depending on specific conditions.

The mean and standard deviation can automatically adjust according to the current sliding window, so the outlier estimation criterion shows good adaption to dynamic data stream.

### 6.2. EAOF-OD: Outlier detection with enhanced angle-based outlier factor in high-dimensional data stream.

Due to the dynamic nature of data stream, outlier detection algorithm is supposed to have strong adaptive capacity, be able to deal with datasets with various distributions, and be applied to high-dimensional data environment. EAOF-OD first identifies the cluster centers and then assigns each data point to the proper cluster. EAOF (enhanced angle-based outlier factor) is calculated to evaluate the outlierness of each data point in the scope of each corresponding cluster, and data points with high EAOF are identified as (candidate) outliers. The idea of performing outlier detection after clustering can improve the accuracy of detection. Only when candidate outliers are identified to be outliers during the whole multiple validations can they be declared as real outliers. The whole scheme of EAOF-OD combines the model based on sliding window and multiple validations with efficient outlier detection algorithm, not only can deal with the data stream in real time, but also has low time and space complexity.

Let $S$ be the set of data in sliding window at a certain time, $n$ be the number of data points in $S$, and $S$ can be described as $S = \{X_1, X_2, \ldots, X_n\}$. The detailed steps of EAOF-OD are presented in Algorithm 1.

In Algorithm 1, there are several parameters which need to be set in the first step. They are the number of nearest neighbors $k$, the radius of spatial neighborhood $r$, the number of data blocks contained in a sliding window $\varepsilon$, the times of multiple validations $\lambda$, the coefficient for outlier estimation criterion $\xi$. For the number of nearest neighbors $k$ and the radius of spatial neighborhood $r$, DBSCAN provides good recommendations: $k$ is set to 4 and $r$ is set to the value of the first "valley" in the $k$-dist graph (mapping each point to the distance from its $k$-th nearest neighbor). The number of data blocks contained in a sliding window $\varepsilon$ can be set depending on conditions. When the memory is large enough, the blocks contained in a sliding window can be a bit more. In this paper, $\varepsilon$ is recommended to $2 \sim 5$. The times of multiple validations $\lambda$ can be set to 3, and too

many multiple validations can have bad effect on the real time. As introduced in Section 6.1 the coefficient for outlier estimation criterion can be set to $2 \sim 3$.

6.3. **Effectiveness.** Two synthetic datasets (shown in Figure 9(a) and Figure 10(a)) are created to show the effectiveness of EAOF-OD intuitively. There are three normal clusters (N1, N2, N3) and two outlier clusters (N4, N5) in the 2-dimensional dataset. The 3-dimensional dataset consists of three normal clusters and one outlier cluster. Figures 9(b)-9(d) and Figures 10(b)-10(d) are the performance after applying EAOF-OD on these two synthetic datasets.

Algorithm 1. EAOF-OD: outlier detection with enhanced angle-based outlier factor in high-dimensional data stream

Input: $S$: the set of data in sliding window at a certain time
$k$: the number of nearest neighbors
$r$: the radius of spatial neighborhood
$\varepsilon$: the number of data blocks contained in a sliding window
$\lambda$: the times of multiple validations
$\xi$: the coefficient for outlier estimation criterion
Output: $O$: outliers set

Begin
1: Initialize the parameters $k$, $r$, $\varepsilon$, $\lambda$, $\xi$;
2: According to Equations (1), (2), (3), (4), calculate the cluster centrality factor $\tau(X_i)$ of every point;
3: List all $\tau(X_i)$ in descending order $\tau(X_{q_1}) \geq \tau(X_{q_2}) \geq \cdots \geq \tau(X_{q_n})$, get the belonging matrix $F = [f_1, f_2, \ldots, f_n]$ by Equation (5);
4: Identify the cluster centers:
Let $C_{center\_id}$ be the cluster center ID, $C_{cluster\_label}$ be the cluster label.
①: Initialize $C_{center\_id} = 0$ and $C_{cluster\_label}(X_{q_1}) = 0$;
②: **for** $i = 2 : n$ **do**
③: **while** the distances between $X_{q_i}$ and $X_{q_j}$ $(j = 1, 2, \ldots, i-1)$ meet $dist\left(X_{q_i}, X_{q_j}\right) > r$ **do**
④: $C_{center\_id} = C_{center\_id} + 1$;
⑤: $C_{cluster\_label}(X_{q_i}) = C_{center\_id}$;
⑥: **end** //end of while
⑦: **end** //end of for
5: Cluster:
①: **for** $i = 1 : n$ **do**
②: $C_{cluster\_label}(i) = C_{cluster\_label}(f_i)$;
③: **end** //end of for
④: Cluster points with the same cluster labels into one cluster, get $C_{center\_id}$ clusters $C_1, C_2, \ldots, C_m$;
6: Outlier detection:
Let $G$ be the candidate outlier set which stores the candidate outliers.
①: **for** $i = 1 : C_{center\_id}$, and with respect to each cluster $C_i$ **do**
②: Use Equations (1), (6), (7), (8) to obtain the $EAOF(X_j)$ of each point $X_j$ in $C_i$;
③: Compute the mean $\mu$ and standard deviation $\delta$ of all $EAOF$s of points in $C_i$, identify the suspicious points, put them in candidate outlier set $G$;
④: **end** //end of for

7: Multiple validations

Let $\gamma(X_i)$ be the times of being validated of point $X_i$, $\alpha(X_i)$ be the times of showing suspicious, initialize them to zero.

a: **if** $\gamma(X_i) < \varepsilon$ **then**
b: **if** $X_i$ is detected as normal in the validation **then**
c: $\gamma(X_i) = \gamma(X_i) + 1$, declare $X_i$ as normal point, remain $X_i$ in the sliding window;
d: **else**
e: $\gamma(X_i) = \gamma(X_i) + 1$, $\alpha(X_i) = \alpha(X_i) + 1$, declare $X_i$ as candidate outlier, keep $X_i$ in the sliding window for next validation;
f: **end** //end of if
g: **else if** $\gamma(X_i) = \varepsilon$ and $\alpha(X_i) < \gamma(X_i)$ **then**
h: Stop the validation of $X_i$, declare $X_i$ as normal data point, delete $X_i$ from memory;
i: **else if** $\varepsilon \leq \gamma(X_i) < \lambda$ **then**
j: **if** $X_i$ shows suspicious in the validation **then**
k: $\gamma(X_i) = \gamma(X_i) + 1$, $\alpha(X_i) = \alpha(X_i) + 1$, declare $X_i$ as candidate outlier, remain $X_i$ in the sliding window for next validation;
l: **else**
m: End the validation of $X_i$, declare $X_i$ as normal data point, remove $X_i$ from memory;
n: **end** //end of if
o: **else if** $\gamma(X_i) = \lambda$, $\alpha(X_i) = \lambda$ **then**
p: Stop the validation of $X_i$, declare $X_i$ as real outlier, remove $X_i$ from the candidate outlier set $G$, store $X_i$ in the real outlier set $O$, output $O$
q: **else**
r: Stop the validation of $X_i$, declare $X_i$ as normal data point, delete $X_i$ from memory;
s: **end** //end of if

Exit.

It can be observed from Figures 9(a)-9(d) that 2 outlier clusters and 15 scattered outliers in 2-dimensional dataset can be detected effectively by EAOF-OD with none outlier missed. From Figure 9(d), it is obvious that there are 3 normal objects falsely detected as outliers. Though these 3 normal points are generated from Gaussian distributions which belong to normal clusters, they deviate quite far from normal clusters and show suspicious during all the $\lambda$ ($\lambda = 3$) times of multiple validations.

From Figures 10(a)-10(d) it is clear that the performance of EAOF-OD in 3-dimensional dataset is still satisfying. Outliers including the outlier cluster and 47 out of 48 scattered outliers are detected with just one scattered outlier left out. The reason which leads to the missing outlier is that this scattered outlier is located so close to normal cluster that it is declared as normal point during multiple validations.

7. **Experimental Result and Comparative Analysis.** To verify the effectiveness of the EAOF-OD, the quality evaluation and time complexity evaluation experiments are performed over several synthetic and real datasets with EAOF-OD, Algorithm Y, I-IncLOF and DSABOD. The information about the datasets is shown in Table 1. The 8 datasets represent different kinds of datasets with different number of dimensions, different

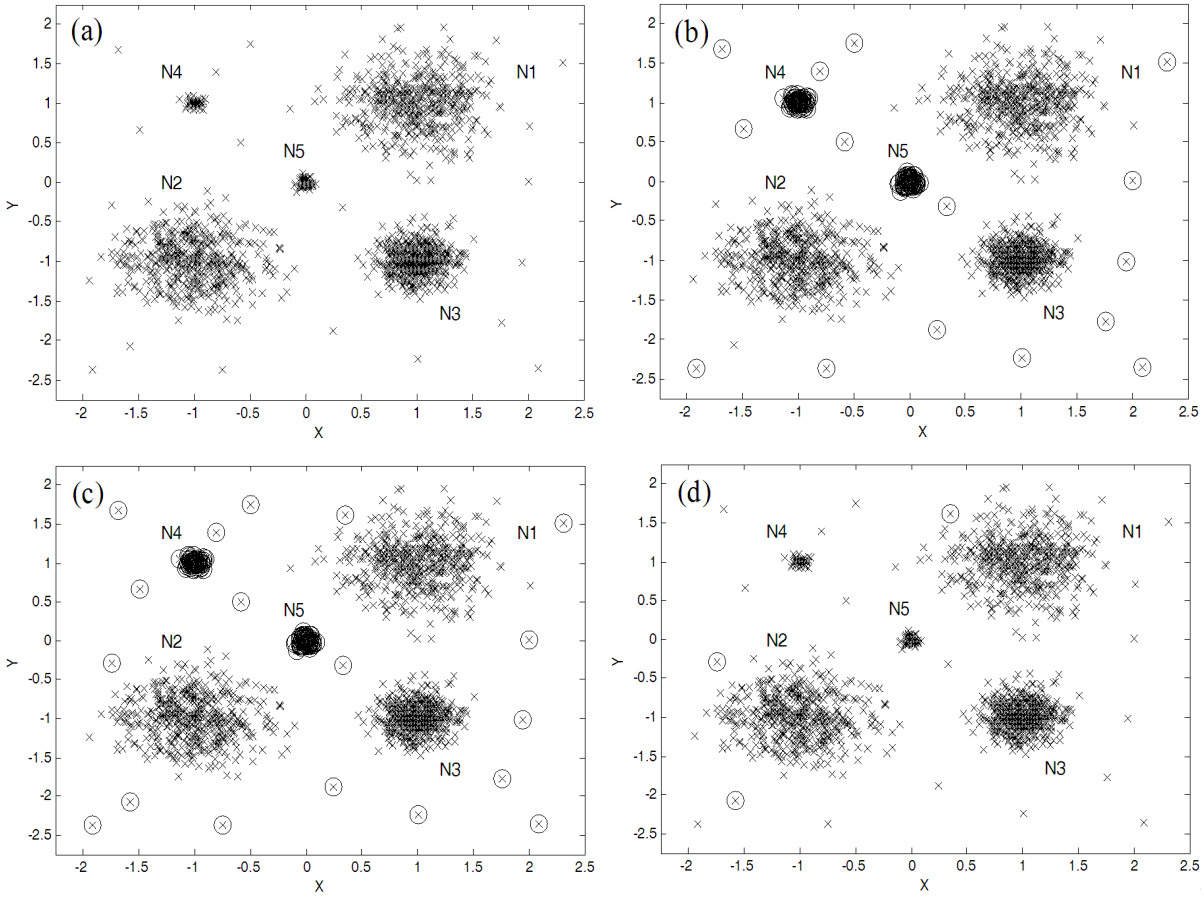


FIGURE 9. Performance of EAOF-OD on a 2-dimensional synthetic dataset: (a) distribution of the 2-dimensional synthetic dataset; (b) real outliers in the 2-dimensional synthetic dataset (denoted by circles); (c) detected outliers (denoted by circles); (d) falsely detected outliers (denoted by circles)

TABLE 1. Characteristics of the datasets

| Dataset name | Number of instances | Number of attributes |
|---|---|---|
| Synthetic dataset 1 | 1615 | 2 |
| Synthetic dataset 2 | 860 | 3 |
| Yeast | 1484 | 8 |
| Abalone | 4177 | 8 |
| Breast Cancer | 699 | 10 |
| Mushroom | 8124 | 22 |
| Ionosphere | 351 | 34 |
| KDD1999 | 973959 | 41 |

sizes, different distributions. All experiments were conducted in matlab R2014a on Intel Core i5-3230M, 2.6GHz with 4GB memory running on Windows $10 \times 64$.

Synthetic dataset 1 is composed of 1615 data instances of which there are 1600 data records generated from 5-modal mixture of 2-dimensional Gaussian distributions and 15 scattered data records. The dataset consists of three normal clusters each containing 500 data records, two outlier clusters each having 50 data instances, and 15 scattered

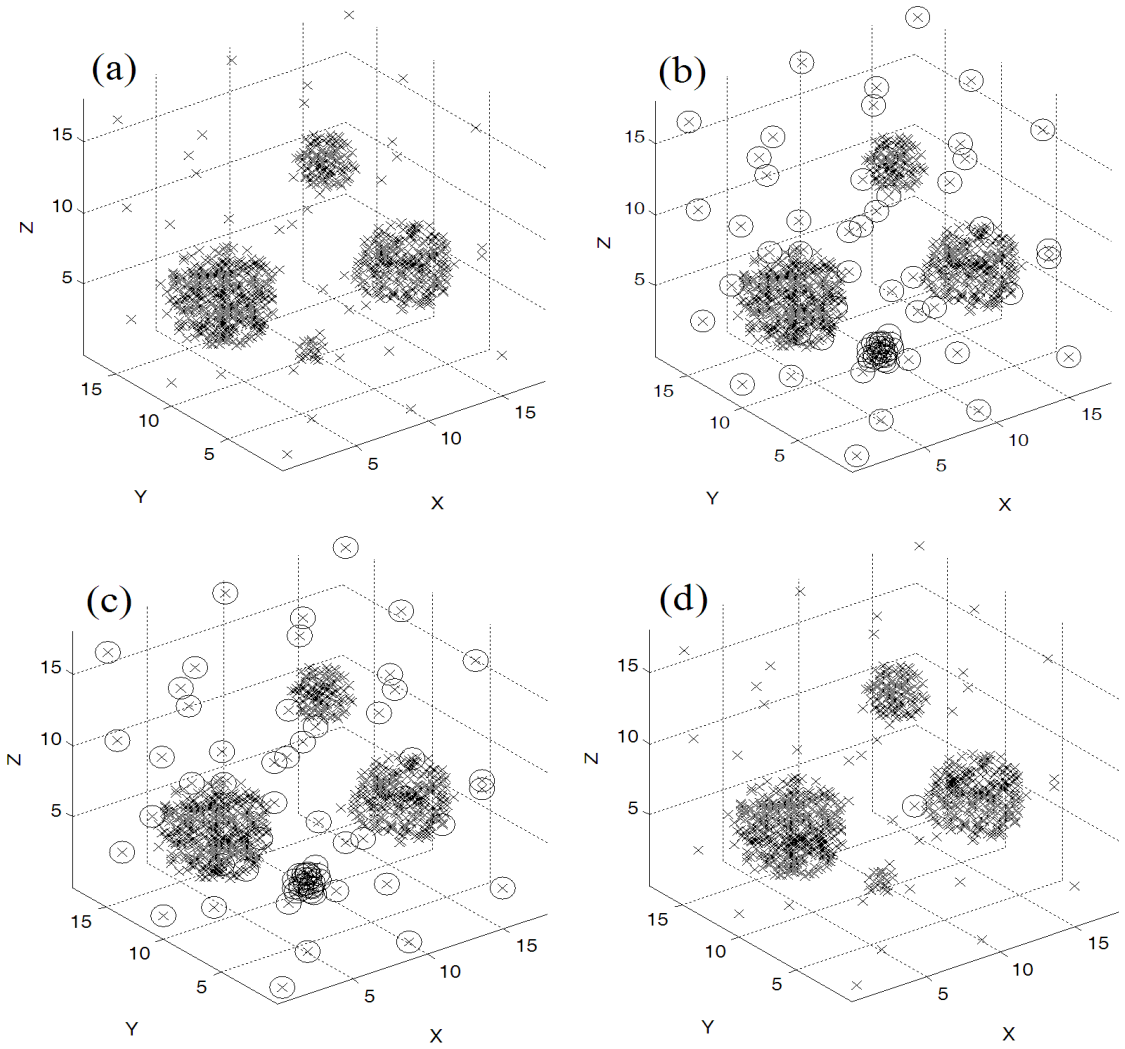


FIGURE 10. Performance of EAOF-OD on a 3-dimensional synthetic dataset: (a) distribution of the 3-dimensional synthetic dataset; (b) real outliers in the 3-dimensional synthetic dataset (denoted by circles); (c) detected outliers (denoted by circles); (d) missing outliers (denoted by circles)

data records being generated based upon domain knowledge (statistical characteristics like mean, standard deviation, class distribution, etc.).

Synthetic dataset 2 is composed of 860 data records of which there are 791 data records forming 3 normal clusters and 21 data records constituting an outlier cluster. Besides 48 scattered outliers are planted in Synthetic dataset 2 based on the statistical characteristics (min, max, mean and standard deviation) of attributes.

Real datasets are taken from UCI machine learning repository [31]. There are total 1484 data instances in Yeast dataset forming 10 classes. CYT, NUC, MIT, ME3, cover maximum of data records and other classes cover small portion. In experiments 50% records of ME2, ME1, EXC, VAC, POX are removed and the rest 50% are treated as outliers. Abalone contains 4177 examples with 8 attributes, there are 28 classes in the dataset and those classes containing instances less than 60 are treated as outlier clusters (groups of outliers). In Breast Cancer dataset there are 699 data records making up 2 normal classes. 4.86% additional outlier objects are planted into the dataset for a better performance analysis based on the statistical characteristics of attributes. The Mushroom

dataset consists of two classes 4208 in Edible and 3916 in Poisonous. 2.15% of Poisonous are selected as outliers. Ionosphere dataset has 351 data records, wherein 225 data records are Good and 126 data records are Bad. In experiments, 8.73% of Bad remain as outliers. KDD1999 contains the records of 7 weeks of network traffic. There are 972781 instances of normal data, whereas the number of attack records is too high to be considered as outliers (3925650). In order to make the dataset more realistic, the rare attack types (U2R and R2L) are selected as outliers so that outliers become a small ratio of normal instances.

7.1. **Quality evaluation.** Outlier detection rate and false alarm rate are used to evaluate the quality performance of algorithms. Detection rate refers to the ratio between the number of correctly detected outliers to the total number of actual outliers. False alarm rate is the ratio between the number of normal objects that are misinterpreted as outlier to the total number of alarms. In the experiment, EAOF-OD is applied with $\varepsilon = 2$, $\lambda = 3$, $\xi = 2.5$, and parameters $k$ and $r$ are set as DBSCAN described in Section 6.2.

Table 2 shows detailed comparative experiment results of EAOF-OD vs. Algorithm Y, I-IncLOF and DSABOD on 8 synthetic and real datasets. The graphical comparison results are shown in Figure 11 and Figure 12.

Table 2. EAOF-OD vs. Algorithm Y, I-IncLOF and DSABOD

| Dataset | Algorithm Y | | I-IncLOF | | DSABOD | | EAOF-OD | |
|---|---|---|---|---|---|---|---|---|
| | detection rate/% | false alarm rate/% | detection rate/% | false alarm rate/% | detection rate/% | false alarm rate/% | detection rate/% | false alarm rate/% |
| Synthetic dataset 1 | 98.26 | 0.40 | 97.39 | 5.58 | 37.39 | 23.53 | **100.00** | **0.20** |
| Synthetic dataset 2 | 97.10 | 0.50 | 95.65 | 0 | 89.86 | 2.63 | **98.55** | 0 |
| Yeast | **95.13** | 2.39 | 93.94 | 4.35 | 75.76 | 20.83 | 94.57 | **1.46** |
| Abalone | 70.12 | 2.25 | 69.23 | 5.93 | 65.38 | 11.76 | **86.94** | **1.25** |
| Breast Cancer | 91.17 | 0.43 | 82.35 | 7.22 | 91.18 | 18.92 | **94.12** | **0** |
| Mushroom | 92.28 | 12.38 | 97.73 | 2.24 | 97.15 | 24.81 | **98.69** | **1.02** |
| Ionosphere | 72.73 | **9.09** | 72.73 | 54.55 | **91.67** | 31.25 | 83.33 | 16.67 |
| KDD1999 | 91.09 | 14.02 | 86.14 | 32.56 | 57.43 | 31.58 | **91.27** | **9.56** |

It can be observed from Table 2, Figure 11 and Figure 12 that generally EAOF-OD outperforms Algorithm Y, I-IncLOF and DSABOD with higher outlier detection rate and lower false alarm rate. It is because EAOF-OD utilizes more accurate way to estimate outlierness, as well as multiple validations. Algorithm Y updates parameters including weight of each attribute when new instance joins in, which improve its effectiveness and ability of adaptation to data stream in multi-dimensional space. Algorithm Y does well in some datasets. Although in Yeast dataset its detection rate is higher than EAOF-OD, its false alarm rate is higher than EAOF-OD, too. And in higher dimensional space, the detection rate of Algorithm Y is worse than EAOF-OD due to its distance-based and density-based nature. As is shown in Table 2, I-IncLOF achieves 97.39% as the outlier detection rate and 5.58% as the false alarm rate on Synthetic dataset 1, which is satisfying. However, when coming to Abalone dataset with 10 attributes, outlier detection rate decreases to 69.23%, and to Ionosphere dataset with 34 attributes the false alarm rate increases to 54.55%, and to KDD1999 dataset with 41 attributes the false alarm

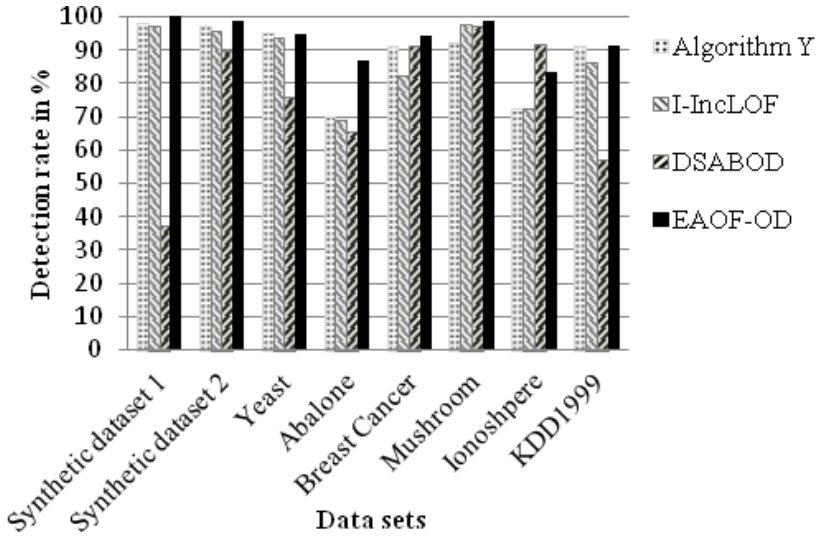


FIGURE 11. Outlier detection rate

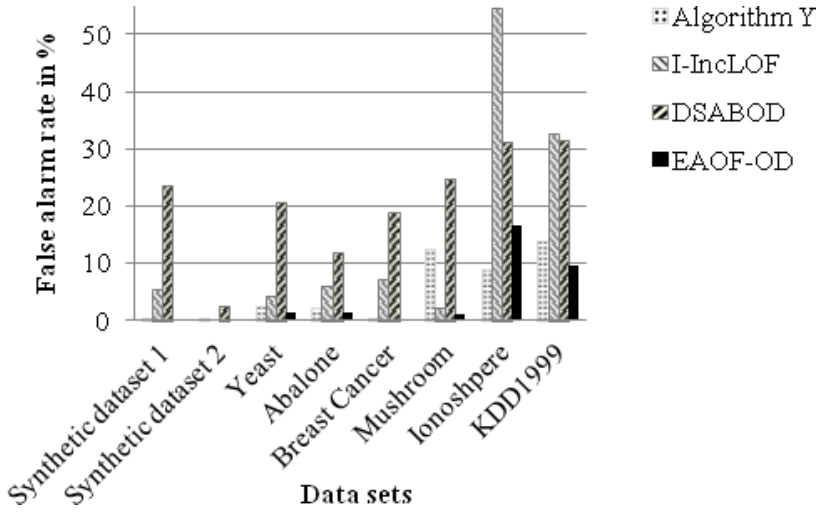


FIGURE 12. False alarm rate

rate gets to 32.56%. And I-IncLOF has poor adaptation, when the characteristics of data stream vary (such as the Abalone dataset and the Breast Cancer dataset), I-IncLOF performs worse. From the results of EAOF-OD and DSABOD, it can be concluded that generally when dimension increases the advantage of angle-based algorithm gets more prominent. However, when there are outlier clusters in dataset (such as Synthetic dataset 1, Synthetic dataset 2, Yeast and Abalone) and the distribution is unbalanced (such as the KDD1999), the traditional angle-based algorithm DSABOD loses effectiveness due to the shortcomings described in Section 3. Although the dimension increases and characteristics vary, generally EAOF-OD performs well and outperforms Algorithm Y, I-IncLOF and DSABOD quite a lot.

7.2. **Time complexity evaluation.** The EAOF-OD performs better than the other three algorithms in terms of detection rate and false alarm rate, but this performance benefit does not come without cost. The time complexity of EAOF-OD is $O(n^3)$. The time complexity of Algorithm Y is $O(n^2)$. The time complexity of I-IncLOF is $O(n \cdot \log n)$. And for DSABOD, when a new data record comes, the outlier factors of all the history data records need to be updated, so the time complexity of DSABOD is $O(n^3)$. It is

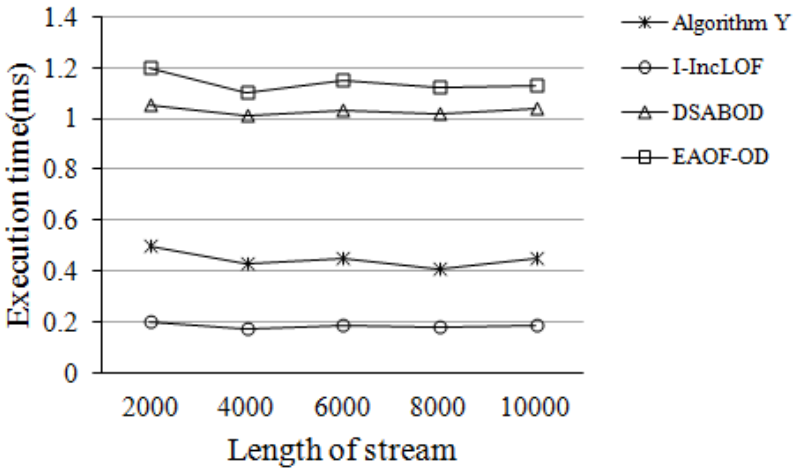


FIGURE 13. Comparison of execution time

obvious that the time complexity of EAOF-OD is the same as DSABOD, and higher than Algorithm Y and I-IncLOF. EAOF-OD needs to cluster the data first, which DSABOD does not have to do so. Although the time complexity of EAOF-OD is the same as DSABOD, EAOF-OD is a little more time-consuming than DSABOD. Figure 13 shows the execution time for the four algorithms when experiment on KDD1999. The time is recorded every $2000^{\text{th}}$ data. In the experiment, EAOF-OD is applied with $\varepsilon = 2$, $\lambda = 3$, $\xi = 2.5$, and parameters $k$ and $r$ are set as DBSCAN described in Section 6.2.

As is shown in Figure 13, execution times of algorithms fluctuate within a small range, which is because the sliding window keeps the number of data records almost unchanged. On average EAOF-OD takes 1.5 times more than Algorithm Y, 5 times more than I-IncLOF and 0.04 millisecond more than DSABOD. Generally the execution time of EAOF-OD is less than 1.2 milliseconds. The data frequency lower than 1.2 millisecond is impractical for most of the current data stream applications, so the execution time of EAOF-OD is quite acceptable. The little more extra time is worthy for EAOF-OD because it offers a good performance improvement over Algorithm Y, I-IncLOF and DSABOD in terms of outlier detection rate and false alarm rate.

8. **Conclusion and Future Work.** In this paper, an outlier detection algorithm with enhanced angle-based outlier factor in high-dimensional data stream (EAOF-OD) is proposed. EAOF-OD fast identifies cluster centers and clusters the dataset in time. Enhanced angle-based outlier factor (EAOF) is described to evaluate the deviation degree accurately in high-dimensional data space. Efficient model based on sliding window and multiple validations is utilized to decrease the false alarm rate. Besides, EAOF-OD reserves and removes the records selectively, which makes sure that the limited memory is consumed properly. Experiments on several synthetic as well as real datasets demonstrate that EAOF-OD outperforms some existing algorithms with higher outlier detection rate and less false alarm rate especially in high-dimensional data environment. For future work, more efforts will be put into the investigation of less dependence of parameters.

**Acknowledgements.** This work was supported by the following foundations: the National Natural Science Foundation of China (61662013, 61362021, U1501252); Natural Science Foundation of Guangxi Province (2016GXNSFAA380149); Guangxi Innovation-Driven Development Project (Science and Technology Major Project) (AA17202024);

the Key Laboratory of Cognitive Radio and Information Processing Ministry of Education (2011KF11); Innovation Project of GUET Graduate Education (2016YJCXB02, 2017YJCX34, 2018YJCX37); the Guilin Scientific Research and Technological Development Project (2016010404-4).

## REFERENCES

[1] Z. Shou, M. Li and S. Li, Outlier detection based on multi-dimensional clustering and local density, *Journal of Central South University*, vol.24, no.6, pp.1299-1306, 2017.

[2] M. A. Jaffar, N. Naveed, S. Zia, B. Ahmed and T.-S. Choi, DCT features based malignancy and abnormality type detection method for mammograms, *International Journal of Innovative Computing, Information and Control*, vol.7, no.9, pp.5459-5513, 2011.

[3] J. Zhang and M. Zulkernine, Anomaly based network intrusion detection with unsupervised outlier detection, *IEEE International Conference on Communications (ICC)*, vol.5, pp.2388-2393, 2006.

[4] A. R. Vasudevan and S. Selvakumar, Local outlier factor and stronger one class classifier based hierarchical model for detection of attacks in network intrusion detection dataset, *Frontiers of Computer Science*, vol.10, no.4, pp.755-766, 2016.

[5] C. Bae, W.-C. Yeh, M. A. M. Shukran, Y. Y. Chung and T.-J. Hsieh, A novel anomaly-network intrusion detection system using ABC algorithms, *International Journal of Innovative Computing, Information and Control*, vol.8, no.12, pp.8231-8248, 2012.

[6] J. Chen, D. J. Dewitt, F. Tian and Y. Wang, NiagaraCQ: A scalable continuous query system for Internet databases, *ACM SIGMOD International Conference on Management of Data*, vol.29, no.2, pp.379-390, 2000.

[7] Y. Zhu and D. Shasha, Statstream: Statistical monitoring of thousands of data streams in real time, *Proc. of the 28th International Conference on Very Large Data Bases (VLDB)*, pp.358-369, 2002.

[8] G. Medioni, I. Choen, F. Brémond, S. Hongeng and R. Nevatia, Event detection and analysis from video streams, *IEEE Trans. Pattern Analysis & Machine Intelligence*, vol.23, no.8, pp.873-889, 2001.

[9] S. Chen, M. Shyu and C. Zhang, Multimedia data mining for traffic video sequence, *Journal of Intelligent Information Systems*, vol.19, no.1, pp.61-77, 2002.

[10] P. Bonnet, J. Gehrke and P. Seshadri, Towards sensor database systems, *Proc. of the 2nd International Conference on Mobile Data Management*, pp.3-14, 2001.

[11] E. M. Knorr, R. T. Ng and V. Tucakov, Distance-based outliers: Algorithms and applications, *The VLDB Journal*, vol.8, nos.3-4, pp.237-253, 2000.

[12] M. M. Breunig, H.-P. Kriegel, R. T. Ng and J. Sander, LOF: Identifying density-based local outliers, *Proc. of the 2000 ACM SIGMOD International Conference on Management of Data*, vol.29, no.2, pp.93-104, 2000.

[13] D. Pokrajac, A. Lazarevic and L. J. Latecki, Incremental local outlier detection for data streams, *IEEE Symposium on Computational Intelligence and Data Mining*, pp.504-515, 2007.

[14] K. Gao, F. J. Shao and R. C. Sun, $n$-INCLOF: A dynamic local outlier detection algorithm for data streams, *The 2nd International Conference on Signal Processing Systems (ICSPS)*, vol.2, pp.179-183, 2010.

[15] S. H. Karimian, M. Kelarestaghi and S. Hashemi, I-IncLOF: Improved incremental local outlier detection for data streams, *The 16th CSI International Symposium on Artificial Intelligence and Signal Processing*, pp.23-28, 2012.

[16] P. Spiros, K. Hiroyuki and P. B. Gibbons, LOCI: Fast outlier detection using the local correlation integral, *Proc. of the 19th International Conference on Data Engineering (ICDE)*, pp.315-326, 2003.

[17] X. Lu, T. Yang and Z. Liao, Incremental outlier detection in data streams using local correlation integral, *The 2009 ACM Symposium on Applied Computing (SAC)*, pp.1520-1521, 2009.

[18] H.-P. Kriegel, M. S. Hubert and A. Zimek, Angle-based outlier detection in high-dimensional data, *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.444-452, 2008.

[19] H. Ye, H. Kitagawa and J. Xiao, Continuous angle-based outlier detection on high-dimensional data streams, *Proc. of the 19th International Database Engineering & Applications Symposium*, pp.162-167, 2015.

[20] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, *Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp.226-231, 2008.

[21] R. T. Ng and J. Han, Efficient and effective clustering methods for spatial data mining, *Proc. of the 20th International Conference on Very Large Data Bases (VLDB)*, pp.144-155, 1994.

[22] M. A. Belabbas and P. J. Wolfe, Spectral methods in machine learning and new strategies for very large datasets, *The National Academy of Sciences of the United States of America*, vol.106, no.2, pp.369-374, 2009.

[23] C. C. Aggarwal, J. Han, J. Wang and P. S. Yu, A framework for clustering evolving data streams, *Proc of the 29th International Conference on Very Large Data Bases*, vol.29, no.3, pp.81-92, 2003.

[24] C. C. Aggarwal, J. Han, J. Wang and P. S. Yu, A framework for projected clustering of high dimensional data streams, *Proc. of the 30th International Conference on Very Large Data Bases (VLDB)*, pp.852-863, 2004.

[25] I. Ntoutsi, A. Zimek, T. Palpanas and H.-P. Kriegel, Density-based projected clustering over high dimensional data streams, *Proc. of the 12th SIAM International Conference on Data Mining*, pp.987-998, 2012.

[26] F. Cao, M. Ester, W. Qian and A. Zou, Density-based clustering over an evolving data stream with noise, *Proc. of the 6th SIAM International Conference on Data Mining*, pp.328-339, 2006.

[27] B. Zhang, S. Qin, W. Wang, D. Wang and L. Xue, Data stream clustering based on fuzzy c-mean algorithm and entropy theory, *Signal Processing*, vol.126, no.2, pp.111-116, 2016.

[28] M. Hassani, P. Spaus, M. M. Gaber and T. Seidl, Density-based projected clustering of data streams, *Proc. of the 6th International Conference on Scalable Uncertainty Management*, pp.311-324, 2012.

[29] Y. Thakran and D. Toshniwal, Unsupervised outlier detection in streaming data using weighted clustering, *The 12th International Conference on Intelligent Systems Design and Applications (ISDA)*, pp.947-952, 2012.

[30] J. Z. Huang, M. K. Ng, H. Rong and Z. Li, Automated variable weighting in k-means type clustering, *IEEE Trans. Pattern Analysis & Machine Intelligence*, vol.27, no.5, pp.657-668, 2005.

[31] M. Lichman, *UCI Machine Learning Repository*, http://archive.ics.uci.edu/m.