


Article

# Adaptive Gaussian Kernel-Based Incremental Scheme for Outlier Detection

Panpan Zhang <sup>1</sup>, Tao Wang <sup>2</sup>, Hui Cao <sup>2,\*</sup>  and Siliang Lu <sup>3</sup>

<sup>1</sup> Engineering Research Center of Autonomous Unmanned System Technology, Ministry of Education, School of Artificial Intelligence, Anhui University, Hefei 230601, China; zppan55@gmail.com

<sup>2</sup> State Key Laboratory of Electrical Insulation and Power Equipment, School of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710049, China; taowangxjtu@gmail.com

<sup>3</sup> School of Electrical Engineering and Automation, Anhui University, Hefei 230601, China; silianglu@ahu.edu.cn

\* Correspondence: huicao@mail.xjtu.edu.cn

**Abstract:** An outlier, known as an error state, can bring valuable cognitive analytic results in many industrial applications. Aiming at detecting outliers as soon as they appear in data streams that continuously arrive from data sources, this paper presents an adaptive-kernel-based incremental scheme. Specifically, the Gaussian kernel function with an adaptive kernel width is employed to ensure smoothness in local measures and to improve discriminability between objects. The dynamical Gaussian kernel density is presented to describe the gradual process of changing density. When new data arrives, the method updates the relevant density measures of the affected objects to achieve outlier computation of the arrived object, which can significantly reduce the computational burden. Experiments are performed on five commonly used datasets, and experimental results illustrate that the proposed method is more effective and robust for incremental outlier mining automatically.

**Keywords:** outlier detection; incremental scheme; adaptive Gaussian kernel; dynamical density; incremental outlier factor



**Citation:** Zhang, P.; Wang, T.; Cao, H.; Lu, S. Adaptive Gaussian Kernel-Based Incremental Scheme for Outlier Detection. *Electronics* **2023**, *12*, 4571. <https://doi.org/10.3390/electronics12224571>

Academic Editor: Andrei Kelarev

Received: 27 September 2023

Revised: 4 November 2023

Accepted: 6 November 2023

Published: 8 November 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Outlier detection, also termed anomaly detection or abnormal detection, is an essential research work in the data-mining domain. In industrial applications, the process has gained much attention as fault detection to identify error states that appear rarely and deviate so much from others [1–6], ranging from network intrusion, video surveillance, factory production monitoring, and fraudulent transactions, etc. Static outlier-detection methods determine outliers in a dataset without new data arriving. In practice, outlier detection from streaming data is of great importance in real-time automatic monitoring. Streaming data are characterized by unbalanced data distribution or complex density regions due to uncertain sources of outliers. To meet a broad range of requirements, there are two solutions. One is the reapplication of the static method for the dataset with added data, and the other is the application of the incremental outlier-detection method. The first technique is not applicable in big data streams because of the computational expense of the objects not suffering from the newly collected data [7,8]. Hence, it is necessary to research a more competitive incremental method.

From a review of the literature, four main categories of the incremental method are presented for outlier mining [9–11], namely the model-based method, clustering-based method, distance-based method, and density-based method. The characteristics of four main categories of incremental outlier-detection methods are briefly shown in Table 1. The model-based incremental method [12–14] is easy to implement, while its excellent performance depends on sufficient data and the prior knowledge of dataset distribution, and the method is only applicable to low-dimensional datasets. The incremental multi-class

outlier-detection model [14], a model-based incremental method, employs an incremental support vector machine (ISVM) to divide the incoming query sample and detect outliers in a multi-class data stream. The clustering-based incremental method [15,16] is a low-cost and highly portable method that mostly considers the optimization of clusters. For this method, the deviation degree of the object cannot be reflected, and the detection performance is mainly determined by the adopted clustering algorithm. Incremental DBSCAN [17,18] can find clusters with arbitrary shapes and handle the noise, and it calculates the means between newly obtained data and core objects of existing clusters to assign the cluster of the data. The distance-based incremental method's conception [19] is easy to comprehend, while it is sensitive to the nearest neighbor parameter, and it is not suitable for an unbalanced density distribution dataset. Exact-Storm [20] conducts expired slide processing, new slide processing, and outlier reporting when the introduced window slides, and the window includes data storage as well as the specific structure to keep neighborhood information incrementally. All the methods mentioned above use a global perspective to recognize outliers, while the density-based incremental method employs the definition of local density to realize outlier detection in unbalanced distribution [21,22]. The classic and state-of-the-art incremental method is the incremental local outlier factor algorithm (IncLOF) [23]. IncLOF possesses the same detection results as the reapplication of the static LOF [24], which detects outliers based on local reachability density. Furthermore, IncLOF needs to recompute a small fraction of the objects affected by new arriving objects so it can reduce the time complexity of outlier mining. As with IncLOF, the incremental connectivity-based outlier factor algorithm (IncCOF) also possesses the same detection results as the reapplication of the static COF, which detects outliers based on average local connectivity [25]. IncLOF is limited in that it only considers the density difference between objects and their neighbors, while IncCOF considers both local density and connectivity. However, IncCOF may not always outperform IncLOF in outlier detection. The choice between IncLOF and IncCOF depends on the characteristics of the dataset. These two incremental methods are both sensitive to the number of nearest neighbors and do not perform well for datasets with small clusters. More variants of IncLOF are also presented, mainly considering how to tackle the memory-occupied issue, such as memory-efficient incremental LOF [26], density-summarizing incremental LOF [27], time-aware density-summarizing incremental LOF [28] and self-adaptive density-summarizing incremental LOF [29]. Memory-efficient incremental LOF considers a sliding window, which lets data profiles update within the window and indicates whether a suspected outlier is truly an outlier. Density-summarizing incremental LOF employs nonparametric Rényi divergences to improve the summarization process, where the past data are summarized, and a model is proposed for detecting outliers in a stream environment. Time-aware density-summarizing incremental LOF presents "approximate LOF" based on historical information following the discharge of out-of-data data to detect local outliers in streaming data. Self-adaptive density-summarizing incremental LOF proposes a density-based sampling technique that summarizes the historical data without prior distribution knowledge of objects so the algorithm can determine the outlier score of each object with a little memory. However, the accuracy of outlier score calculation in these variants is sacrificed to some extent to guarantee a reduction of memory consumption due to the historical estimation being imprecise. In addition, these algorithms are also sensitive to the number of nearest neighbors. It is worth noting that the kernel-density-based method can achieve smoothness and improvement of discriminability in outlier measures [30,31], where the calculation of data converts to the inner product of high-dimensional space by kernel function to enhance the difference of data. On the other hand, the existing representative incremental methods ignore the changing density of the object with its neighbors increasing to improve the different descriptions of density. Thus, the main problem of the previous density definition is the lack of interpretability of density changes and the consideration of parametric sensitivity. To leverage the advantages and overcome the drawbacks of the density-based incremental method, the proposed algorithm is motivated by the IncLOF and kernel method, and its important

research problems are the emphasis on the recognition of local outliers, incrementation, adaptability, interpretability, effectiveness, and robustness.

**Table 1.** Characteristics of the four main categories of incremental outlier-detection methods.

Method Categories	Advantages	Drawbacks
Model-based incremental method	global perspective, excellent performance, easy to implement	depend on the sufficient data and the prior knowledge of dataset distribution, only applicable to low-dimensional datasets
Clustering-based incremental method	global perspective, low cost, high portability	without deviation degree, depend on the adopted clustering algorithm
Distance-based incremental method	global perspective, give outlier degree, easy to comprehend	sensitive to the nearest neighbor parameter, not suitable for unbalanced density distribution dataset
Density-based incremental method	consider local measures of objects, give outlier degree, suitable for unbalanced density distribution dataset, adapt to practical application	sensitive to the nearest neighbor parameter, ignore the density changes in objects

In this paper, an adaptive-kernel-based incremental scheme is proposed for outlier detection in a data stream with newly arriving data. The goal is to learn efficient incremental outlier detection to identify the outliers of the newly collected data. The incremental outlier factor is calculated to indicate the degree of the object being an outlier, whereas the incremental dynamical Gaussian kernel density outlier factor is presented to reflect the dynamic changes in kernel density as one nearest neighbor after another arrives. The measured kernel density is defined via the Gaussian kernel function with an adaptive width, where the kernel function improves discriminability between objects and improves robustness to the nearest neighbor size, and the adaptive width increases the discriminability further. To achieve incremental outlier detection, the method determines the affected objects and updates their helpful measures served for the computation of new objects, where the computation cost is reduced greatly. Specifically, this paper contains the following main contributions.

1. A Gaussian kernel function with an adaptive kernel width is employed to ensure smoothness in the local measures and to improve discriminability between objects.
2. The dynamical Gaussian kernel density is presented to describe the gradual process of changing density.
3. When new data arrives, the method updates the measures of the affected objects used for outlying computation of the arrived object, which can significantly reduce the computational burden.
4. The experimental results illustrate that the proposed method is more effective and robust for incremental outlier mining automatically.

## 2. Preliminaries

In this section, the computation of outlier factors and incremental learning schemes are introduced in detail.

### Computation of Outlier Factor

Ascending distance series of an object  $p$ : The ascending distance series is a merging of the object  $p$  and its nearest neighborhood, in ascending order by their Euclidean distance, denoted as  $ADS(p)$ , which is expressed as

$$ADS(p) = \{p, c_1, c_2, \dots, c_r\} \quad (1)$$

$$dist(p, c_i) \leq k\text{-distance}(p), c_i \in N_k(p), i = 1, 2, \dots, r$$

where  $r$ ,  $N_k(p)$  and  $dist(p, c_i)$  denote the size of  $N_k(p)$ , the nearest neighborhood of  $p$  and the Euclidean distance between  $p$  and  $c_i$ , respectively.  $k$ -distance( $p$ ) denotes the  $k$ -distance expressing the Euclidean distance of  $p$  from the  $k$ th nearest neighbor.

Gaussian kernel function with an adaptive width: the adaptive Gaussian kernel function achieves the calculation of data and converts to the inner product of high-dimensional space by the mapping function [30,32–35], denoted as  $Ker(o, p)$ , which can be written as

$$\begin{aligned}
 Ker(o, p, \sigma_s(p)) &= \langle \Phi(o), \Phi(p) \rangle \\
 &= \sum_{s=1}^{SS} \frac{1}{SS} \exp(-\|o - p\|^2 / \sigma_s^2(p)) \\
 \sigma_s(p) &= 2^{s-1} \sigma(p), \quad s = 1, 2, \dots, SS \\
 \sigma(p) &= \alpha [dist_{max} + dist_{min} + \varepsilon - dist_k(p)]
 \end{aligned} \tag{2}$$

where  $\langle, \rangle$ ,  $\Phi()$ ,  $\sigma(p)$  ( $\sigma(p) > 0$ ) and  $SS$  denote the inner product, the mapping function, the adaptive kernel width, and the multiscale kernel parameter, respectively.  $dist_k(p) = \frac{1}{k} \sum_{i=1}^r dist(p, c_i)$  is the neighborhood distance of  $p$ .  $dist_{min} = \min\{dist_k(p_j) | j = 1, 2, \dots, r\}$  and  $dist_{max} = \max\{dist_k(p_j) | j = 1, 2, \dots, r\}$  are the largest and the smallest neighborhood distances of all objects, respectively.  $\varepsilon$  is a given positive to ensure non-zero for the width, and  $\alpha$  is a compensation parameter for smoothness control.

Notably, the multiscale kernel method is intended to find a set of kernel functions with multiscale representation capabilities [33,34]. The synthetic kernel method, a multiscale kernel method, is proposed based on the linear combination of the single kernel function. However, there is no theory for the parameter setting and the combination type of synthetic kernel to solve the uneven distribution of objects, which limits the representation ability of the synthetic kernel method. The Gaussian kernel function possesses a typical multiscale character, which is widely used because of its universal approximation ability. The width of the Gaussian kernel function depends on the object’s location. The large width is well suited for an object positioned within a high-density region. Conversely, the width of the object should be set to a small value when it is situated in a low-density region. In this paper,  $ss$ ,  $\alpha$  and  $\varepsilon$  are set as 1, 0.5, and  $10^{-6}$ , respectively.

Calculation of kernel distance: According to the theory of Gaussian kernel function, the distance of an object in the high-dimensional space and the dynamical Gaussian kernel distance of an object in the nearest neighborhood, denoted as  $k\_dis(o, p)$  and  $dk\_dis(p, c_i)$ , are calculated as follows:

$$\begin{aligned}
 k\_dis(o, p) &= \sqrt{\|\Phi(o) - \Phi(p)\|^2} \\
 &= \sqrt{\Phi(o)\Phi(o) - 2\Phi(o)\Phi(p) + \Phi(p)\Phi(p)} \\
 &= \sqrt{Ker(o, o, \sigma_s(o)) - 2Ker(o, p, \sigma_s(p)) + Ker(p, p, \sigma_s(p))} \\
 &= \sqrt{2 - 2 \sum_{s=1}^{SS} \frac{1}{SS} \exp(-\|o - p\|^2 / \sigma_s^2(p))}
 \end{aligned} \tag{4}$$

$$dk\_dis(p, c_i) = \sum_{ii=1 \wedge k\_dis(p, c_{ii}) \neq k\_dis(p, c_{i'i}), i'i < ii}^i k\_dis(p, c_{ii}) \tag{5}$$

Calculation of kernel density: According to the theory of Gaussian kernel function and the formula of kernel distance, a new definition of Gaussian kernel density is presented to express the density between the object and its nearest neighbor, denoted as  $gk\_den(p, c_i)$ , which is calculated as

$$gk\_den(p, c_i) = \frac{1 + i}{\sum_{ii=1}^i dk\_dis(p, c_{ii})} \tag{6}$$

Furthermore, dynamical Gaussian kernel density and dynamical Gaussian kernel density fluctuation are both presented to indicate Gaussian kernel density changes in an

object with its neighbors increasing, respectively, denoted as  $gk\_DD(p)$  and  $gk\_DDF(p)$ , which are calculated as follows:

$$gk\_DD(p) = \sum_{i=1}^r \frac{\left( \frac{gk\_den(p,c_i) - gk\_den(p,c_{i+1})}{gk\_den(p,c_i)} \right)^2}{i} \quad (7)$$

$$gk\_DDF(p) = \sum_{i=1}^r \frac{\left( \frac{gk\_DD(p) - gk\_DD(c_i)}{gk\_DD(p)} \right)^2}{i} \quad (8)$$

Please note that the estimated contribution of  $gk\_DD(p)$  and that of  $gk\_DDF(p)$  are more obtained by the earlier objects in  $ADS(p)$ .

The calculation of incremental outlier factor: based on calculations on kernel density, the incremental dynamical Gaussian kernel density outlier factor of an object is estimated by the ratio of its dynamical Gaussian kernel density fluctuation and the average dynamical Gaussian kernel density fluctuation of its neighbors, denoted as  $IncDGOF(p)$ , which is calculated as

$$IncDGOF(p) = \frac{|N_k(p)| \cdot gk\_DDF(p)}{\sum_{o \in N_k(p)} gk\_DDF(o)} \quad (9)$$

The  $IncDGOF$  value is a local deviate factor to indicate the outlier degree of the object, and the higher degree of the object indicates a larger value of  $IncDGOF$ .

### 3. Incremental Learning Scheme

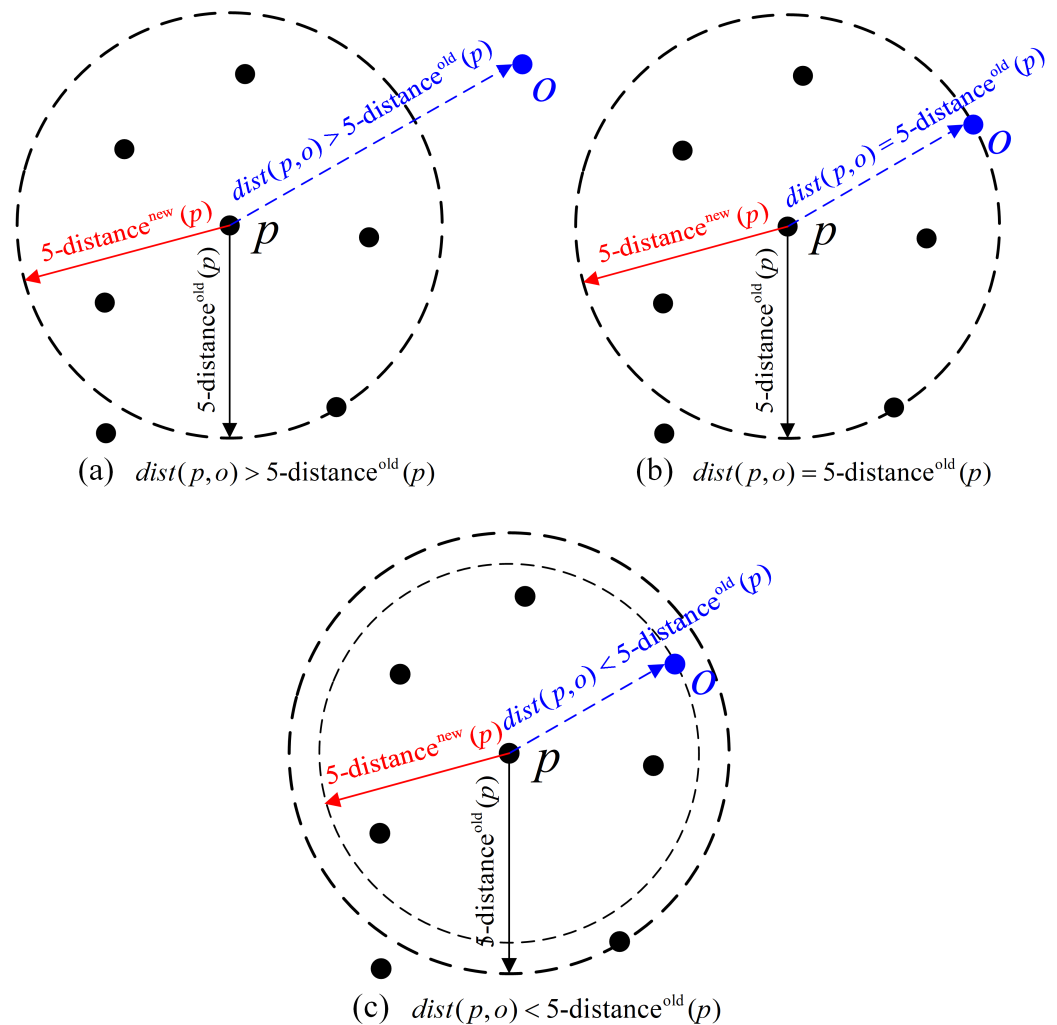
An incremental learning scheme is proposed based on a Gaussian kernel function with an adaptive kernel width and updating the measures of the affected objects along with the new objects arriving.

The detection task has two phases for incremental outlier detection of streaming data. The first phase is the model-training phase to learn the normal pattern of the training dataset, and the second phase is the model-testing phase to compare newly arrived objects with the previously learned normal pattern. The requirement of the task is the training dataset consisting of normal objects. Notably, the fundamental assumption of the incremental learning scheme is that normal objects of the detection system are not changing over time; in other words, the change in normal objects is negligible in the incremental detection period of the detection system. Theoretically, the model can be retrained regularly to absorb the normal change emerging in the incremental detection system.

To update the measures of the affected objects, they are both determining the affected objects and recalculating their measures when the new object arrives. There are two orders of the affected objects, including the objects whose dynamical Gaussian kernel density caused by the  $k$ -distance and the nearest neighborhood are changed and whose dynamical Gaussian kernel density fluctuation caused by dynamical Gaussian kernel density is changed. When a new object  $o$  arrives, the first affected object  $p$  will be updated with a new  $k$ -distance and a new nearest neighborhood. Figure 1 shows three situations of incremental updating of the  $k$ -distance and the nearest neighborhood after the arrival of a new object.

Three situations are analyzed in detail as follows:

- (a) If  $dist(p, o) > k\text{-distance}^{\text{old}}(p)$ , then do not update.
- (b) If  $dist(p, o) = k\text{-distance}^{\text{old}}(p)$ , then update  $N_k(p)$ .
- (c) If  $dist(p, o) < k\text{-distance}^{\text{old}}(p)$ , then update  $k\text{-distance}(p)$  and  $N_k(p)$ .



**Figure 1.** Three situations of incremental updating of  $k\text{-distance}(p)$  and  $N_k(p)$  after a new object  $o$  arrived ( $k = 5$ ).

Moreover, the neighborhood distance of the new object may be exceedingly large in the model-testing phase. According to Formula (3), its positivity requirement could probably be violated by a negative Gaussian kernel width. To meet the adaptive of the width, the updating function is proposed as

$$\sigma(p) = \begin{cases} \alpha[\text{dist}_{\min} + \varepsilon], & \text{dist}_k(p) > \text{dist}_{\max} \\ \alpha[\text{dist}_{\max} + \text{dist}_{\min} + \varepsilon - \text{dist}_k(p)], & \text{otherwise} \end{cases} \quad (10)$$

where  $\text{dist}_{\min}$  and  $\text{dist}_{\max}$  have been fixed in the model-training phase.

Simplistically, if a few objects arrive at the same time, at this point, we can focus on the newly arrived objects to update the measures of objects, including the nearest neighborhood of the newly arrived objects, the nearest neighborhood of the former nearest neighborhood, and the nearest neighborhood of the former nearest neighborhood further.

#### 4. The Proposed Method

In this section, the proposed incremental outlier-detection algorithm is formulated in detail, and we analyze the time complexity of the proposed algorithm.

##### 4.1. IncDGOF Algorithm

For streaming data, the status of objects would be changed when new objects arrive. This paper proposes the IncDGOF algorithm for outlier detection in a data stream. To

achieve efficient incremental outlier mining, the IncDGOF algorithm updates the measures of the affected objects used for outlying computation of the arrived object. This algorithm includes the computation of outlier factors and an incremental learning scheme consisting of finding affected objects and updating their measures. The detailed IncDGOF algorithm is formulated in Algorithm 1. First, calculate the Euclidean distance between the newly collected object and each object of the existing dataset and compute its ascending distance series, kernel width, and kernel density. Next, find the first-order affected objects and update their measures regarding the calculation of dynamical Gaussian kernel density. Then, find the second-order affected objects and update their measures regarding the calculation of dynamical Gaussian kernel density fluctuation. Finally, determine the incremental outlier factor of the collected object.

---

**Algorithm 1:** IncDGOF algorithm
 

---

**Input:** Dataset  $D$ , new collected object  $o$ , nearest neighbor number  $k$

**Output:**  $IncDGOF$  value of  $o$

```

1: Scale  $o$  to zero mean and unit variance
2: for all  $p \in D$  do
3:   Compute  $dist(o, p), dist(p, o) = dist(o, p)$ 
4: end for
5: Compute  $ADS(o)$ 
6: Compute  $\sigma(o)$  using Formula (3)
7: Compute  $gk\_den(o, a), a \in N_k(o)$  using Formula (6)
/*Find 1st order affected objects and update their measures*/
8: for all  $p \in D$  do
9:   if  $dist(p, o) \leq k\text{-distance}(p)$  then
10:     $S_{affect} \leftarrow p$ 
11:   end if
12: for all  $d_s \in S_{affect}$  do
13:   Update  $\sigma(d_s)$  using Formula (10)
14:   Update  $gk\_den(d_s, c_i), c_i \in N_k(d_s)$  using Formula (6)
15:   Update  $gk\_DD(d_s)$  using Formula (7)
16: end for
17: Compute  $gk\_DD(o)$  using Formula (7)
/*Find 2nd order affected objects and update their measures*/
18: for all  $p \in D$  do
19:   if  $d_s \in S_{affect} \wedge d_s \in N_k(p)$  then
20:     $SI_{affect} \leftarrow p$ 
21:     $SA_{affect} \leftarrow S_{affect} \cup SI_{affect}$ 
22:   end if
23: end for
24: for all  $d_s \in SA_{affect}$  do
25:   Update  $gk\_DDF(d_s)$  using Formula (8)
26: end for
27: Compute  $gk\_DDF(o)$  using Formula (8)
28: Compute  $IncDGOF(o)$  using Formula (9)

```

---

#### 4.2. Time Complexity Analysis

For the proposed algorithm, let  $n$ ,  $d$ , and  $k$  be the size of the dataset, the dimension, and the nearest neighbor, respectively. Moreover, let  $a$  and  $b$  be the size of the object whose dynamical Gaussian kernel density is changed and whose dynamical Gaussian kernel density fluctuation is changed, respectively. According to the description of Algorithm 1, the time complexity of the algorithm can be considered to be two main parts with two sub-steps. The two sub-steps of each part determine the affected objects and the updating of their related

measures when the new object arrives. First, the time complexity of determining in  $S_{affect}$  is  $O(nd)$ , and the time complexity of updating kernel width, Gaussian kernel density, and dynamical Gaussian kernel density of the objects in  $S_{affect}$  is  $O(a + a + ak)$ . Second, the time complexity of determining in  $SA_{affect}$  is  $O(ank)$ , and the time complexity of updating dynamical Gaussian kernel density fluctuation of the objects in  $SA_{affect}$  is  $O(bk)$ . Therefore, the time complexity of the whole process is  $O(nd + a + a + ak + ank + bk)$ . Although  $a, b, k, d \ll n$ , the whole time complexity of the algorithm can be written as  $O(n)$ .

## 5. Experimental Setup and Analysis

In this section, IncDGOF, IncLOF, and IncCOF are performed on five real datasets with different sizes to verify their performance. Furthermore, the experiments evaluate the involved scaling factors to demonstrate the suggestion value of  $\alpha$  and the suggestion value of  $SS$  for the proposed method.

### 5.1. Experimental Datasets and Implementation Details

Five frequently adopted datasets are taken from the UCI Machine-Learning Repository (<http://archive.ics.uci.edu/ml>, accessed on 15 March 2021), which are the Wine dataset, Ionosphere dataset, Phoneme dataset, Vowel dataset, and Cup 99 Smtip dataset [29,31,32]. Concretely, experiments are performed on datasets to search their rare class, and some objects are randomly eliminated from the classes of the Wine dataset, Ionosphere dataset, Phoneme dataset, and Smtip dataset to build an uneven distribution. This operation is commonly used by many researchers to evaluate the performance of outlier-detection algorithm [36,37]. The experiments contain two parts. For the first part, algorithms are performed on the training dataset, which comprises 80% of the dataset. For the second part, algorithms are performed on the rest of the objects of the dataset, consisting of all outliers and the other normal objects of the dataset, expressed as the newly arriving dataset. The summary of details for each dataset is provided in Table 2.

**Table 2.** Characteristics of five experimental datasets.

Datasets	Objects	Training Dataset	New Arriving Dataset	Attributes	Outliers
Wine	81	65	26	12	10
Ionosphere	245	196	34	5	20
Phoneme	500	400	100	5	50
Vowel	1456	1167	289	12	50
Smtip	5000	4000	1000	3	30

The effectiveness of algorithms is estimated by four metrics, which are the precision, the recall, the rank power, and the area under the receiver's operational characteristic (ROC) curve [37], denoted as Pr, Re, RP, and AUC, respectively. In addition, AUC is also employed to analyze the robustness of algorithms to the tuning parameters, where the value of  $k$  ranges in [1, 50] [38]. Pr represents the proportion of abnormal samples within the first  $m$  samples detected. As precision increases, so does the number of detected outliers in the sample. Re refers to the proportion of samples predicted to be outliers out of the actual abnormal samples. A higher recall rate indicates a greater proportion of correctly detected abnormal samples. RP assesses the positions of returned outliers. An outlier positioned earlier in the returned list has a greater contribution to the rank power compared to one placed later in the list. AUC is a better metric used to quantify an algorithm's capacity to distinguish outliers. The ROC curve's horizontal axis represents the false positive rate, while the vertical axis represents the true positive rate. When an algorithm exhibits higher classification accuracy, the ROC curve approaches the top-left corner, and the AUC value approaches 1. Conversely, when the algorithm's classification accuracy is lower, the ROC curve tends to be closer to the lower-right corner, resulting in



a lower AUC value. In summary, the larger the value of Pr, Re, RP, and AUC, the more efficient the outlier-detection algorithm.

5.2. Experimental Results and Discussions of the Methods

For effectiveness estimating, the nearest neighbor size  $k$  is set at 5% of the object number of the dataset. The values of  $k$  are 4, 12, 25, 73, and 250 for the Wine dataset, Ionosphere dataset, Phoneme dataset, Vowel dataset, and Cup 99 Smtip dataset, respectively. The proposed method is performed with the fixed empirical parameters in the experiments, where the values of  $\alpha$  and that of  $SS$  are set to 0.5 and 1, respectively. Experimental results of IncLOF, IncCOF, and IncDGOF for five different datasets are shown in Table 3, where the top  $m$  denotes the number of the observed outlier candidates and  $m$  is set as multiples of the number of outliers in the dataset within four multiples. For five different datasets, the experimental results of IncDGOF are always superior to those of IncLOF and IncCOF, except for the RP values of the Smtip dataset. The RP values of IncDGOF are just inferior to those of IncLOF for the Smtip dataset. According to the incremental scheme of IncLOF and IncCOF, IncLOF values and IncCOF values of all affected objects will be recomputed in the training dataset. However, the training dataset consists of normal objects and the outlier degree of the object should be fixed in the training dataset, i.e., the outlier degree of the training dataset should not be influenced by the newly collected objects, particularly the anomalies. Nevertheless, in the incremental scheme of IncDGOF, the outlier degree of the training dataset is retained, and outlier factors of the affected objects are just updated to serve outlier computation of the newly collected objects. Thus, IncDGOF can reduce the computational burden. The proposed incremental scheme is more suitable to real application data and is more efficient in application.

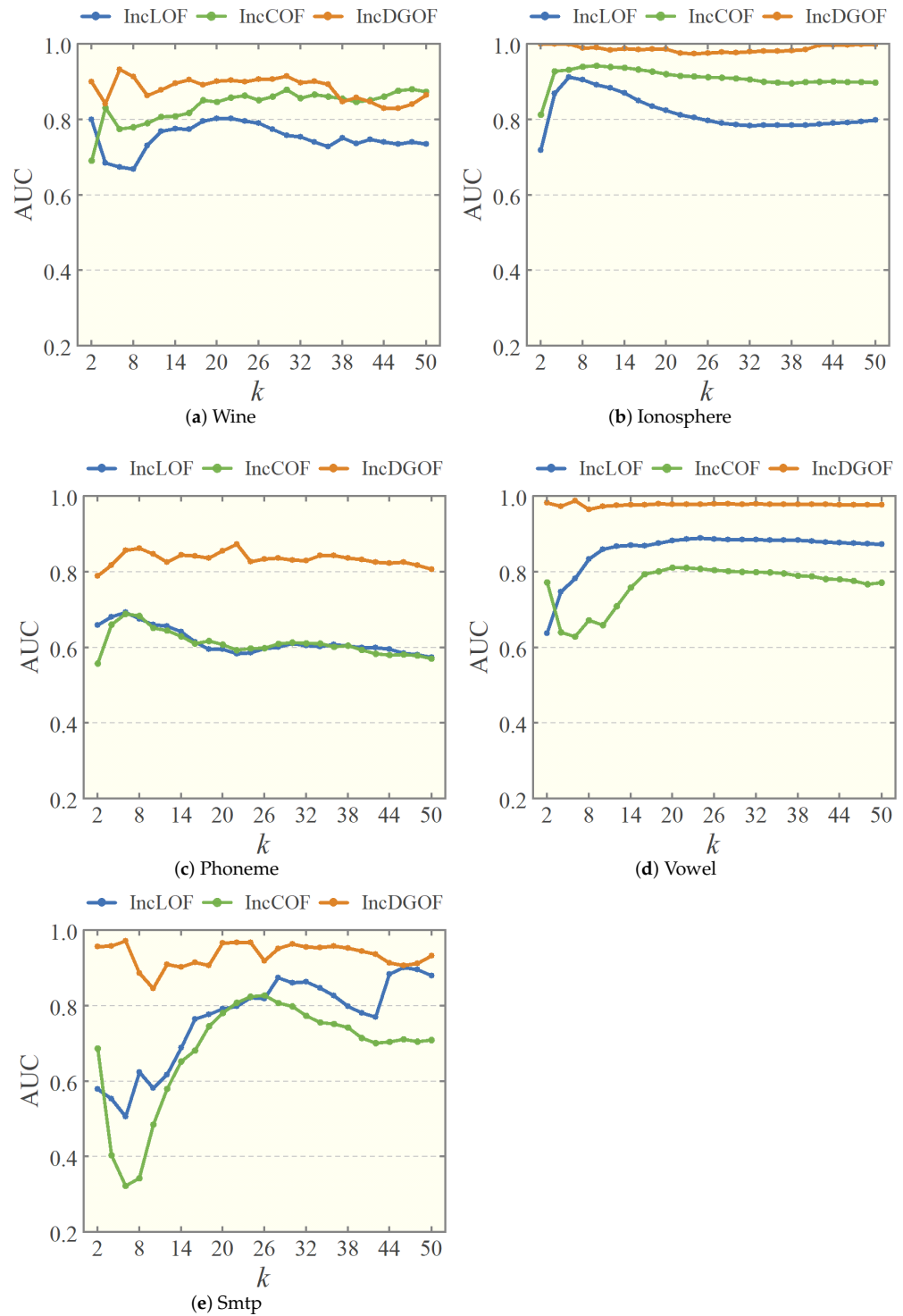
Table 3. Experimental results of IncLOF, IncCOF, and IncDGOF for all datasets.

Datasets	Methods	IncLOF				IncCOF				IncDGOF			
		Pr	Re	RP	AUC	Pr	Re	RP	AUC	Pr	Re	RP	AUC
Wine	Top 10	0.30	0.30	<b>0.67</b>		0.30	0.30	0.40		<b>0.40</b>	<b>0.40</b>	0.50	
	Top 20	0.20	0.40	0.34		0.30	0.60	0.37		<b>0.40</b>	<b>0.80</b>	<b>0.44</b>	
	Top 30	0.20	0.60	0.26	0.68	0.27	0.80	0.34	0.83	<b>0.30</b>	<b>0.90</b>	<b>0.44</b>	<b>0.84</b>
	Top 40	0.18	0.70	0.23		0.25	1.00	0.31		<b>0.23</b>	<b>0.90</b>	<b>0.44</b>	
	parameter $k$	4	4	4	4	4	4	4	4	4	4	4	4
Ionosphere	Top 20	0.30	0.30	0.35		0.55	0.55	0.72		<b>0.90</b>	<b>0.90</b>	<b>0.96</b>	
	Top 40	0.33	0.65	0.34		0.40	0.80	0.60		<b>0.48</b>	<b>0.95</b>	<b>0.95</b>	
	Top 60	0.25	0.75	0.33	0.88	0.30	0.90	0.52	0.94	<b>0.32</b>	<b>0.95</b>	<b>0.95</b>	<b>0.98</b>
	Top 80	0.24	0.95	0.29		0.24	0.95	0.50		<b>0.24</b>	<b>0.95</b>	<b>0.95</b>	
	parameter $k$	12	12	12	12	12	12	12	12	12	12	12	12
Phoneme	Top 50	0.08	0.08	0.08		0.12	0.12	0.10		<b>0.46</b>	<b>0.46</b>	<b>0.54</b>	
	Top 100	0.17	0.34	0.14		0.13	0.26	0.13		<b>0.36</b>	<b>0.72</b>	<b>0.47</b>	
	Top 150	0.15	0.44	0.14	0.59	0.13	0.40	0.13	0.59	<b>0.26</b>	<b>0.76</b>	<b>0.44</b>	<b>0.87</b>
	Top 200	0.14	0.54	0.14		0.14	0.54	0.14		<b>0.21</b>	<b>0.82</b>	<b>0.39</b>	
	parameter $k$	25	25	25	25	25	25	25	25	25	25	25	25
Vowel	Top 50	0.38	0.38	0.66		0.40	0.40	0.48		<b>0.58</b>	<b>0.58</b>	<b>0.75</b>	
	Top 100	0.26	0.52	0.44		0.26	0.52	0.38		<b>0.41</b>	<b>0.82</b>	<b>0.57</b>	
	Top 150	0.20	0.60	0.36	0.74	0.21	0.62	0.32	0.86	<b>0.31</b>	<b>0.92</b>	<b>0.50</b>	<b>0.98</b>
	Top 200	0.17	0.68	0.29		0.16	0.62	0.32		<b>0.24</b>	<b>0.94</b>	<b>0.49</b>	
	parameter $k$	73	73	73	73	73	73	73	73	73	73	73	73
Smtip	Top 30	<b>0.67</b>	<b>0.67</b>	<b>1</b>		<b>0.67</b>	<b>0.67</b>	0.98		<b>0.67</b>	<b>0.67</b>	0.98	
	Top 60	<b>0.33</b>	<b>0.67</b>	<b>1</b>		<b>0.33</b>	<b>0.67</b>	0.98		<b>0.33</b>	<b>0.67</b>	0.98	
	Top 90	0.22	0.67	<b>1</b>	0.87	0.22	0.67	0.98	0.78	<b>0.22</b>	<b>0.67</b>	0.72	<b>0.92</b>
	Top 120	0.17	0.67	<b>1</b>		0.17	0.67	0.98		<b>0.18</b>	<b>0.70</b>	0.72	
	parameter $k$	250	250	250	250	250	250	250	250	250	250	250	250

The bolder ones mean better.

Outlier-detection algorithms adopt  $k$ -nearest neighbors to calculate the outlier factor, so they are probably sensitive to the parameter  $k$ . Figure 2 shows AUC values with  $k$  ranging from 1 to 50 for five different datasets, which clarifies the influence of the nearest neighbor size for the algorithms. Compared with the AUC curves of IncLOF and that of IncCOF, the AUC curves of IncDGOF are always smoother and steadier for each dataset, and IncDGOF possesses larger AUC values for various  $k$  values in five different datasets. Gaussian kernel function with adaptive width can realize that the inseparable issue of low-dimensional space is transformed into a linearly separable issue of high-dimensional space,

and it can increase the adaptivity of the algorithm to parameter  $k$ . On the other hand, the presented dynamical Gaussian kernel density improves the difference description between the objects. Therefore, the proposed algorithm not only builds a precise incremental scheme but is also robust to the nearest neighbor size for the computation of outlier degrees.

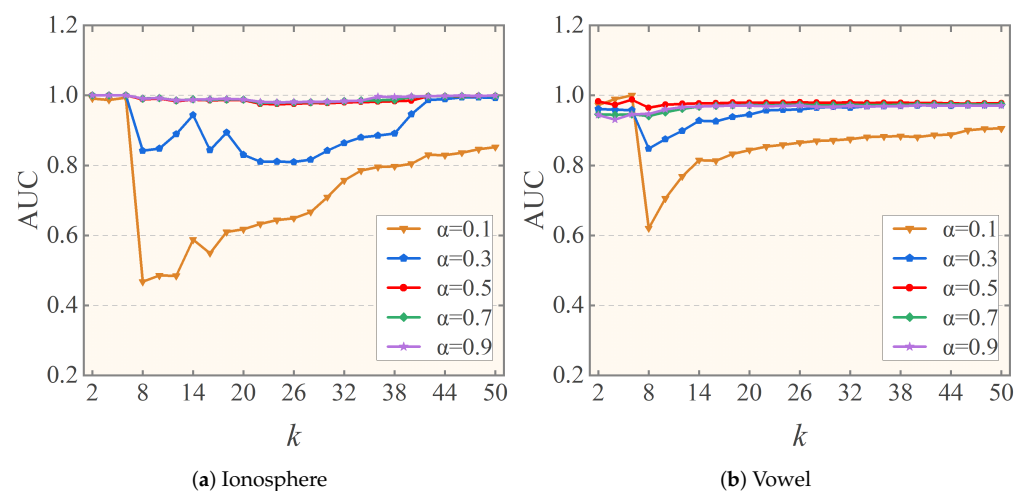


**Figure 2.** AUC values of IncLOF, IncCOF and IncDGOF for (a) Wine dataset, (b) Ionosphere dataset, (c) Phoneme dataset, (d) Vowel dataset, and (e) Smtpt dataset, respectively.

### 5.3. Experimental Results and Discussions for the Involved Scaling Factors

The Ionosphere and Vowel datasets are adopted to evaluate the involved scaling factors of the proposed algorithm by comparing AUC values. Moreover, the suggestion value of  $\alpha$  and the suggestion value of  $SS$  are determined for using density estimation in the proposed method. For the smoothness experiments regarding the controlling parameter,  $SS$  of the proposed method is set as 1, and  $\alpha$  of the proposed method is, respectively, set as 0.1, 0.3, 0.5, 0.7, and 0.9. For the parameter experiments regarding multiscale kernel function,  $\alpha$  of the proposed method is set as 0.5, and  $SS$  of the proposed method is, respectively, set as 1, 2, 3, 4, and 5.

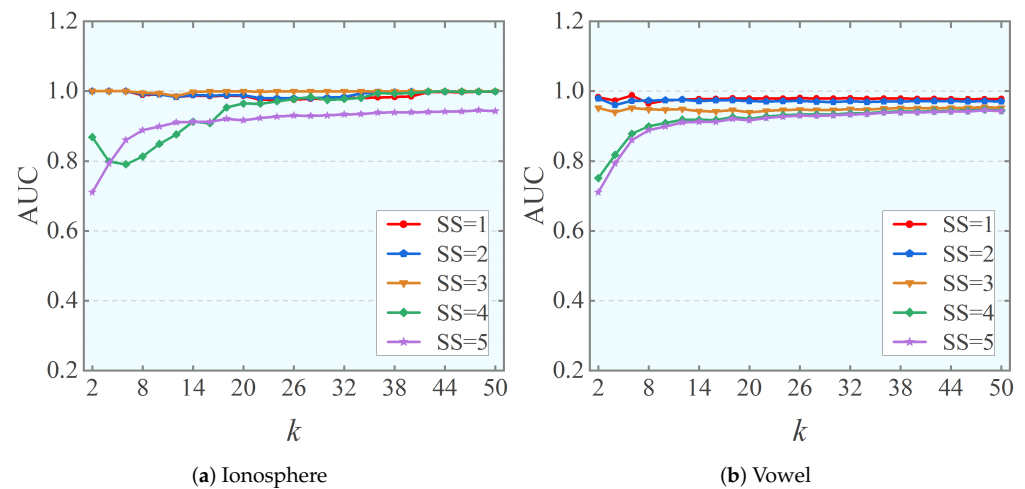
For the Ionosphere and Vowel datasets, AUC values of the proposed algorithm with different values of  $\alpha$  are shown in Figure 3, and the AUC values with  $k$  ranging from 1 to 50 reflect how the value of  $\alpha$  works for outlier detection. The proposed algorithm with various  $k$  values has close and larger AUC values when the value of  $\alpha$  is 0.5, 0.7, and 0.9. Furthermore, the proposed algorithm has similar stable AUC curves when the value of  $\alpha$  is 0.5, 0.7, and 0.9. The AUC values of the proposed algorithm are greatly changed and inferior when the value of  $\alpha$  is 0.1 and 0.3. Therefore, the suggestion value of  $\alpha$  can be chosen in [0.5, 1] for the proposed algorithm. This conclusion satisfies that  $\alpha$  can be determined by “Silverman’s rule-of-thumb” with regard to the density estimation issue where the suggestion value is selected in the interval [0.5, 1] [32,39].



**Figure 3.** AUC values of the proposed method with  $\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$  for (a) Ionosphere dataset, and (b) Vowel dataset, respectively.

For the Ionosphere and Vowel datasets, AUC values of the proposed algorithm with different values of  $SS$  are shown in Figure 4, and the AUC values with  $k$  ranging from 1 to 50 reflect how the value of  $SS$  works for outlier detection. The proposed algorithm performed similarly with stable and higher AUC values along with  $k$  increasing when  $SS$  is set as 1, 2, and 3. The proposed algorithm has obvious fluctuation and relatively worse values of the AUC curve when  $SS$  is set as 4 and 5. Therefore, the suggestion value of  $SS$  can be given as 1, 2, and 3 for the proposed algorithm.

According to the experimental results of parameters  $\alpha$  and  $SS$  on two different characteristic datasets, the suggestion value of  $\alpha$  can be chosen in [0.5, 1], and that of  $SS$  can be set to 1, 2, and 3.



**Figure 4.** AUC values of the proposed method with  $SS = 1, 2, 3, 4, 5$  for (a) Ionosphere dataset, and (b) Vowel dataset, respectively.

## 6. Conclusions

This paper has proposed an adaptive-kernel-based incremental scheme for industrial outlier detection. In data streams, the automatic monitoring of outliers integrates the computation of outlier factors and adaptive incremental strategy. The definition of outlier factor indicates the gradual process of changing density, which enhances the discriminability between objects and the explanation of the density change. Gaussian kernel function with an adaptive kernel width is employed to ensure smoothness in the local measures and to improve discriminability between objects, and dynamical Gaussian kernel density is presented to describe the gradual process of changing density. When new data arrives, the method updates the measures of the affected objects used for outlying computation of the arrived object, which can significantly reduce the computational burden. According to the experimental results, IncDGOF is superior in terms of detection capability and robustness to the nearest neighbor size compared with IncLOF and IncCOF. For the scaling factors of IncDGOF, the suggestion value of  $\alpha$  is selected in the interval  $[0.5, 1]$ , and the suggestion value of  $SS$  can be given as 1, 2, and 3. Moreover, incremental strategy reinforces its applicability in the industrial field. The proposed method holds significant potential for widespread application in industrial complex data streams characterized by varying density regions because it can indicate the gradual process of changing density. In the industrial domain, engineers can efficiently manage objects that display high outlier factor values in real time in accordance with specific actual requirements.

The multiple kernel function can be built based on various kernel functions. Considering the diversity of the kernel function and its difference description of separability in high-dimensional space, the estimation of an outlier factor combined with the multiple kernel function is also an interesting future research issue for improving detection performance.

**Author Contributions:** Conceptualization, P.Z. and H.C.; methodology, P.Z. and H.C.; software, P.Z. and T.W.; validation, P.Z. and T.W.; investigation, P.Z. and S.L.; writing—original draft preparation, P.Z.; writing—review and editing, P.Z. and S.L.; project administration, H.C.; funding acquisition, P.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Natural Science Foundation of China (Grant No. 62303013).

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zou, L.; Wang, Z.; Geng, H.; Liu, X. Set-membership filtering subject to impulsive measurement outliers: A recursive algorithm. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 377–388. [[CrossRef](#)]
2. Pan, Z.; Wang, Y.; Yuan, X.; Yang, C.; Gui, W. A classification-driven neuron-grouped SAE for feature representation and its application to fault classification in chemical processes. *Knowl. Based Syst.* **2021**, *230*, 107350. [[CrossRef](#)]
3. Yu, T.; Hu, J.; Yang, J. Intrusion detection in intelligent connected vehicles based on weighted self-information. *Electronics* **2023**, *12*, 2510. [[CrossRef](#)]
4. Kim, S.; Hwang, C.; Lee, T. Anomaly based unknown intrusion detection in endpoint environments. *Electronics* **2020**, *9*, 1022. [[CrossRef](#)]
5. Cai, S.; Huang, R.; Chen, J.; Zhang, C.; Liu, B.; Yin, S.; Geng, Y. An efficient outlier detection method for data streams based on closed frequent patterns by considering anti-monotonic constraints. *Inform. Sci.* **2021**, *555*, 125–146. [[CrossRef](#)]
6. Slavakis, K.; Banerjee, S. Robust hierarchical-optimization RLS against sparse outliers. *IEEE Signal Process. Lett.* **2020**, *27*, 171–175. [[CrossRef](#)]
7. Degirmenci, A.; Karal, O. Robust incremental outlier detection approach based on a new metric in data streams. *IEEE Access* **2021**, *9*, 160347–160360. [[CrossRef](#)]
8. Li, A.; Xu, W.; Liu, Z.; Shi, Y. Improved incremental local outlier detection for data streams based on the landmark window model. *Knowl. Inf. Syst.* **2021**, *63*, 2129–2155. [[CrossRef](#)]
9. Taha, A.; Hadi, A.S. Anomaly detection methods for categorical data: A review. *ACM Comput. Surv.* **2019**, *52*, 38. [[CrossRef](#)]
10. Cai, S.; Li, Q.; Li, S.; Yuan, G.; Sun, R. WMFP-Outlier: An efficient maximal frequent-pattern-based outlier detection approach for weighted data streams. *Inf. Technol. Control* **2019**, *48*, 505–521. [[CrossRef](#)]
11. Gao, J.; Ji, W.; Zhang, L.; Li, A.; Wang, Y.; Zhang, Z. Cube-based incremental outlier detection for streaming computing. *Inform. Sci.* **2020**, *517*, 361–376. [[CrossRef](#)]
12. Ozkan, H.; Ozkan, F.; Kozat, S.S. Online anomaly detection under markov statistics with controllable type-i error. *IEEE Trans. Signal Process.* **2015**, *64*, 1435–1445. [[CrossRef](#)]
13. Ruff, L.; Kauffmann, J.R.V.; Vandermeulen, R.A.; Montavon, G.; Samek, W.; Kloft, M.; Dietterich, T.G.; Müller, K. A unifying review of deep and shallow anomaly detection. *Proc. IEEE* **2021**, *109*, 756–795. [[CrossRef](#)]
14. Degirmenci, A.; Karal, O. iMCOD: Incremental multi-class outlier detection model in data streams. *Knowl. Based Syst.* **2022**, *258*, 109950. [[CrossRef](#)]
15. Deshmukh, M.M.K.; Kapse, A.S. A survey on outlier detection technique in streaming data using data clustering approach. *Int. Eng. Comput. Sci.* **2016**, *5*, 15453–15456.
16. Khan, I.; Huang, J.Z.; Ivanov, K. Incremental density-based ensemble clustering over evolving data streams. *Neurocomputing* **2016**, *191*, 34–43. [[CrossRef](#)]
17. Azhir, E.; Navimipour, N.J.; Hosseinzadeh, M.; Sharifi, A.; Darwesh, A. An efficient automated incremental density-based algorithm for clustering and classification. *Future Gener. Comput. Syst.* **2021**, *114*, 665–678. [[CrossRef](#)]
18. Bakr, A.M.; Ghanem, N.M.; Ismail, M.A. Efficient incremental density-based algorithm for clustering large datasets. *Alexandria Eng. J.* **2015**, *54*, 1147–1154. [[CrossRef](#)]
19. Tran, L.; Fan, L.; Shahabi, C. Distance-based outlier detection in data streams. *Proc. VLDB Endow.* **2016**, *9*, 1089–1100. [[CrossRef](#)]
20. Angiulli, F.; Fassetto, F. Detecting distance-based outliers in streams of data. In Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, Lisbon, Portugal, 6–10 November 2007; pp. 811–820.
21. Alghushairy, O.; Alsini, R.; Soule, T.; Ma, X. A review of local outlier factor algorithms for outlier detection in big data streams. *Big Data Cogn. Comput.* **2020**, *5*, 1. [[CrossRef](#)]
22. Degirmenci, A.; Karal, O. Efficient density and cluster based incremental outlier detection in data streams. *Inf. Sci.* **2022**, *607*, 901–920. [[CrossRef](#)]
23. Pokrajac, D.; Lazarevic, A.; Latecki, L.J. Incremental local outlier detection for data streams. In Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining, Honolulu, HI, USA, 1–5 April 2007; pp. 504–515.
24. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 16–18 May 2000; pp. 93–104.
25. Pokrajac, D.; Reljin, N.; Pejic, N.; Lazarevic, A. Incremental connectivity-based outlier factor algorithm. In Proceedings of the Visions of Computer Science-BCS International Academic Conference, London, UK, 22–24 September 2008; pp. 211–223.
26. Karimian, S.H.; Kelarestaghi, M.; Hashemi, S. I-inclof: Improved incremental local outlier detection for data streams. In Proceedings of the CSI International Symposium on Artificial Intelligence and Signal Processing, Fars, Iran, 2–3 May 2012; pp. 23–28.
27. Dupuis, P.; Katsoulakis, M.A.; Pantazis, Y.; Rey-Bellet, L. Sensitivity analysis for rare events based on Rényi divergence. *Ann. Appl. Probab.* **2020**, *30*, 1507–1533. [[CrossRef](#)]
28. Huang, J.W.; Zhong, M.X.; Jaysawal, B.P. Tadilof: Time aware density-based incremental local outlier detection in data streams. *Sensors* **2020**, *20*, 5829. [[CrossRef](#)]
29. Singh, M.; Pamula, R. ADINOF: Adaptive density summarizing incremental natural outlier detection in data stream. *Neural Comput. Appl.* **2021**, *33*, 9607–9623. [[CrossRef](#)]

30. Zhang, L.; Lin, J.; Karim, R. Adaptive kernel density-based anomaly detection for nonlinear systems. *Knowl. Based Syst.* **2018**, *139*, 50–63. [[CrossRef](#)]
31. Zhang, P.; Cao, H.; Zhang, Y.; Wang, J.; Jia, J.; Hu, F. Adjoint dynamical kernel density for anomaly detection. *Neurocomputing* **2022**, *499*, 81–92. [[CrossRef](#)]
32. Wahid, A.; Rao, A.C.S. Rkdos: A relative kernel density-based outlier score. *IETE Tech. Rev.* **2020**, *37*, 441–452. [[CrossRef](#)]
33. Hoi, S.C.; Jin, R.; Zhao, P.; Yang, T. Online multiple kernel classification. *Mach. Learn.* **2013**, *90*, 289–316. [[CrossRef](#)]
34. Pinar, A.J.; Rice, J.; Hu, L.; Anderson, D.T.; Havens, T.C. Efficient multiple kernel classification using feature and decision level fusion. *IEEE Trans. Fuzzy Syst.* **2016**, *25*, 1403–1416. [[CrossRef](#)]
35. Hang, H.; Steinwart, I.; Feng, Y.; Suykens, J. Kernel Density Estimation for Dynamical Systems. *J. Mach. Learn. Res.* **2016**, *19*, 1–49.
36. Aggarwal, C.C.; Sathe, S. Theoretical foundations and algorithms for outlier ensembles. *ACM Sigkdd Explor. Newsl.* **2015**, *17*, 24–47. [[CrossRef](#)]
37. Cao, H.; Ma, R.; Ren, H.; Ge, S.S. Data-defect inspection with kernel-neighbor-density-change outlier factor. *IEEE Trans. Autom. Sci. Eng.* **2016**, *15*, 225–238. [[CrossRef](#)]
38. Tang, B.; He, H. A local density-based approach for outlier detection. *Neurocomputing* **2017**, *241*, 171–180. [[CrossRef](#)]
39. Silverman, B.W. *Density Estimation for Statistics and Data Analysis*; Chapman and Hall: New York, NY, USA, 1986.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.