Research Article

An Efficient Outlier Detection Approach for Streaming Sensor Data Based on Neighbor Difference and Clustering

Saihua Cai ^{1,2}, Jinfu Chen ¹, Baoquan Yin,³ Ruizhi Sun,⁴ Chi Zhang,¹ Haibo Chen,¹ Jingyi Chen,¹ and Min Lin¹

¹School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China

²Jiangsu Key Laboratory of Security Technology for Industrial Cyberspace, Jiangsu University, Zhenjiang 212013, China

³Yantai Academy of China Agricultural University, Yantai 264670, China

⁴College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China

Correspondence should be addressed to Jinfu Chen; jinfuchen@ujs.edu.cn

Received 14 September 2021; Revised 15 January 2022; Accepted 29 January 2022; Published 27 February 2022

Academic Editor: Shifeng Sun

Copyright © 2022 Saihua Cai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In wireless sensor networks (WSNs), the widely distributed sensors make the real-time processing of data face severe challenges, which prompts the use of edge computing. However, some problems that occur during the operation of sensors will cause unreliability of the collected data, which can result in inaccurate results of edge computing-based processing; thus, it is necessary to detect potential abnormal data (also known as outliers) in the sensor data to ensure their quality. Although the clustering-based outlier detection approaches can detect outliers from the static data, the feature of streaming sensor data requires the detection operation in a one-pass fashion; in addition, the clustering-based approaches also do not consider the time correlation among the streaming sensor data, which leads to its low detection accuracy. To solve these problems, we propose an efficient outlier detection approach based on neighbor difference and clustering, namely, ODNDC, which not only quickly and accurately detects outliers but also identifies the source of outliers in the streaming sensor data. Experiments on a synthetic dataset and a real dataset show that the proposed ODNDC approach achieves great performance in detecting outliers and identifying their sources, as well as the low time consumption.

1. Introduction

In recent years, wireless sensor networks (WSNs) have been widely used in a variety of applications, such as object positioning [1], health management [2], industrial safety control [3], and social media [4–6]. WSNs usually consist of lots of low-cost sensor nodes distributed over a wide area, which leads to the data in WSNs being generated in the form of streams. That is, the data in WSNs are arriving instantly and continuously, and the generating speed of the data is very quick. The use of a large number of sensors makes the real-time processing of data face more severe challenges, while edge computing [7, 8] provides flexible and on-demand processing power to quickly process the data in WSNs. Nevertheless, the limitations in terms of memory, communication bandwidth, battery power, and computational capacity existing in sensor nodes cause the unreliability of collected data, which further results in the

inaccurate results of edge computing-based processing. Therefore, it is necessary to perform some measures to guarantee the quality of collected data.

As an efficient technology to ensure accurate and reliable data for edge computing, outlier detection [9–11] (also known as anomaly detection [12, 13]) plays a critical role in data security; it detects those data in each edge of WSNs that deviate from the rest based on a certain measured approach of data pattern [14, 15], thereby providing high-quality data for edge computing. In addition, to detect abnormal data, outlier detection can also be used to discover some abnormal events [16]. Because of the importance of data security in all walks of life, massive outlier detection approaches have been proposed and used in many applications, such as intrusion detection [17, 18], health diagnosis [19], and social network detection [20, 21]. However, most outlier detection approaches require the collected data to be scanned several times, but the characteristics of streaming

sensor data do not allow the multiple scans of data since the time, cost, and computational complexity are very high [22], which leads to these approaches not being effectively used in WSNs.

Compared with distance-based [10, 23], density-based [24], clustering-based [25, 26] outlier detection approaches, clustering-based approaches are often used in WSNs for the advantages of simple, efficient, low time complexity, and low space complexity. However, the clustering-based outlier detection approaches have some issues as follows: (1) easy to determine normal data as an outlier when the size of the sliding window is not suitable; (2) cannot effectively process high-dimensional data for the calculation of distance or density, as it is very time-consuming; (3) the detection accuracy is easily affected by noise or outliers. In addition to the above problems, clustering-based approaches usually consider the spatial correlation between data to determine the outliers but rarely consider the time correlation of the data, while the data in WSNs have a typical time correlation. Furthermore, the sources of outliers that occurred in WSNs could be caused by errors, events, or malicious attacks [27], and different types of outliers can result in different influences; therefore, accurately analyzing the sources of outliers can help to provide some hints about how to handle the detected outliers. For example, if the detected outlier is not an event, it should be removed from the streaming sensor data to ensure the high quality; otherwise, it could be caused by some unexpected reasons and must be paid attention to, such as fire [28]. Unfortunately, most existing clustering-based outlier detection approaches cannot effectively distinguish the sources of outliers [29, 30].

To solve the problems existing in clustering-based outlier detection approaches, with the use of w-k-means algorithm (which assigns different weights to each attribute according to the correlation between attributes to reduce the impact of irrelevant attributes on the clustering results and thus solving the problem of other k-means algorithms for their high false alarms caused by the irrelevant data), this paper proposes an efficient outlier detection approach based on neighbor difference and clustering, namely, ODNDC, to accurately detect the outliers and identify the source of outliers in the streaming sensor data collected from WSNs. The major contributions of this work are concluded as follows:

- (1) With the use of neighbor difference, we propose an efficient outlier detection algorithm to accurately detect potential outliers with the consideration of time correlation of the data, and then use the w-k-means algorithm with the consideration of correlation between each attribute to reduce the impact of irrelevant attributes on the clustering results, thereby improving the detecting performance.
- (2) With the consideration of spatial correlation, we propose an efficient outlier source identification algorithm to improve the identifying accuracy.
- (3) Based on a synthetic dataset and a real dataset, we conduct extensive experiments to evaluate the efficiency of the ODNDC approach, and the experimental result verifies that the proposed ODNDC approach can accurately detect potential outliers

from streaming sensor data and identify the sources of outliers as well as cost in a short time.

The remaining of this paper can be organized as follows. Section 2 presents the related works on outlier detection approaches for WSNs and clustering-based outlier detection approaches. Section 3 gives some definitions and presents the classification of outliers. Section 4 first describes the framework of the proposed approach, and then presents the details of the proposed approach. Section 5 demonstrates our experimental results. Finally, we conclude our paper with an outline of future work in Section 6.

2. Related Works

In this section, we first review the related outlier detection approaches for WSNs, and then review the clustering-based outlier detection approaches.

2.1. Outlier Detection Approaches for WSNs. The outlier detection is an essential and challenging task, which aims to identify the data instances that significantly differ from other observations, while outlier detection for WSNs is more difficult because data distribution may change accompanied with the environment. To accurately detect the outliers existing in WSNs, researchers proposed many efficient outlier detection approaches in recent years. Exploiting the time-series analysis, Zhang et al. [31] defined the normal behaviors based on the spatial and temporal correlations of the data, and then proposed an efficient temporal and spatial real data-based outlier detection technique (called TSOD) to identify the outliers in WSNs. However, the communication overhead and spatial consumption of TSOD are very high. Around the concept of edge computing, Bharti et al. [32] proposed an in-network contextual outlier detection on edge (called INCODE) to detect the contextual outliers and estimate the abnormal degree; the experimental result verified by the INCODE can accurately detect the outliers as well as minimize the WSNs resource consumption. Poornima and Paramasivan [33] first reduced the dimensionalities online with the principal component analysis to handle the irrelevant and redundant data, and then proposed an online locally weighted projection regression (called OLWPR) to identify the outliers in WSNs; however, the AUC of OLWPR is not ideal. By using the probabilistic inference to adapt the Bayesian networks distributed over wireless sensor nodes, De Paola et al. [34] proposed an adaptive distributed outlier detection approach (called ADOD) to detect outliers in the data collected in WSNs with a high detection accuracy. In addition to ADOD, the local outlier detection algorithm (called LODA) [35] was also based on the adaptive Bayesian network; it detected the potential outliers through time-series modeling on each sensor node locally without collaboration with neighbors. Although the detection efficiency of ADOD and LODA is obviously improved, their time complexity and communication complexity are still very high.

2.2. Clustering-Based Outlier Detection Approaches. In the clustering-based outlier detection approaches [36, 37], it is necessary to define and calculate the distance or similarity metric between two data instances; then, based on the metric, the data instances that are far away from their closest cluster centroid or where their density is below a threshold are declared as outliers. The k-means is one of the most well-known clustering-based algorithms; it has been widely used in outlier detection since its simplicity and efficiency. Elahi et al. [38] presented a strategy of outlier detection in streaming data: as a data instance is detected as abnormal at the first time, it would be declared as a candidate rather than an outlier; and once the times that a candidate outlier is declared abnormal exceeds a threshold in the fixed time, it would be declared as a real outlier; otherwise, it would be regarded as normal. Furthermore, as the dimension of data becomes higher, the space of data becomes sparser, which results in every point in high dimensional space becoming an almost equally good outlier. As a result, the definition of distance becomes meaningless and the clustering-based approaches will fail, known as “curse of dimensionality.” To reduce the number of attributes, many approaches have been proposed. One common way is to execute the feature extraction before clustering [39]; another way is to use weighted k-means algorithm to reduce the effect of irrelevant attributes and noisy attributes by weighting attributes based on their relevance [40]. Common clustering-based approaches use spatial relationship among data to detect outliers regardless of the temporal relationship among data. But, in reality, there are several typical temporal relationships among streaming sensor data. Therefore, it is necessary to detect outliers with the consideration of temporal relationships between the data collected from WSNs, thereby providing accurate and reliable data for edge computing.

Compared with these existing outlier detection approaches for WSNs and clustering-based outlier detection approaches, the proposed approach innovatively uses neighbor difference to detect candidate outliers before performing clustering operations to eliminate the influence of outliers on clustering efficiency, as well as solves the problem of clustering-based approaches not considering time correlation of data. In addition, it also uses spatial correlation of data generated by distributed adjacent sensor to effectively identify the source of outliers, thereby improving the identifying efficiency.

3. Preliminaries

In this section, we provide the necessary background and definitions.

3.1. Sources of Outliers. Outliers may be caused by measurement errors, system errors, or inherent characteristics of the data. Therefore, the handling measures for outliers from different sources are different. The sources of outliers in WSNs can be classified into errors, events, and malicious attacks [27]. Considering that there are fewer outliers caused by malicious attacks in WSNs, and this category of outliers is

not concerned with data, our work mainly focuses on errors and events.

An error normally represents a large change in the reading of a sensor or a large deviation between the data and other surrounding data samples [41]. In WSNs, errors usually originate from unstable or inaccurate sensor readings caused by sensor noise, time drift, human error, etc., or originate from the frame loss or transient network transmission errors caused by weak communication signals. Because the handling of outliers caused by errors will not cause the loss of important information, these outliers can be discarded directly.

An event normally originates from the sudden changes in real-world conditions [41]—rainfall, fire, etc. The sources of events in WSNs can be further divided into internal events or external events. Internal events are usually caused by battery exhaustion, sensor failure, communication interruption, etc., while external events are usually caused by forest fires, air pollution, or environmental changes. The outliers caused by events should be paid sufficient attention, and simply discarding of these outliers may lead to the loss of important information.

In general, the outliers caused by errors appear more frequently than that caused by events, and the probability of outliers caused by errors is more frequent.

3.2. Classification of Outliers. The classification of outliers is very important for the identification of outliers because the sources of outliers are usually caused by different types of outliers. In general, after the outliers are detected by the detection algorithm, domain experts need to analyze and explain the cause and source of outliers, and then the user finally determines whether the suspicious data is a real outlier, while outlier detection algorithm only presents suspicious data to the user from the perspective of data distribution. According to literature [41], outliers can be classified into point outliers, contextual outliers, and collective outliers, but this classification cannot accurately distinguish the sources of outliers in WSNs. To improve the identification accuracy, we assume that the phenomena in real world represented by sensor data is continuous and with a sort of regularity. With this assumption and the characteristics of sensor data, we classify the outliers of sensor data in WSNs into following four types rather than three types.

3.2.1. Point Outliers. If an individual data instance is diverged from normal pattern of the sensor data, it is termed as a point outlier (e.g., t_1 in Figure 1). Point outliers are the simplest type of outliers, and the detection of point outliers is the base of other outlier detections. In general, point outliers are usually caused by errors.

3.2.2. Collective Outliers. If a group of linked data instances are different from entire patterns in the dataset, these data instances are termed as collective outliers (e.g., t_2 in Figure 1).

FIGURE 1: A case for demonstrating types of outliers.

3.2.3. Jump Outliers. If a data instance has much difference from its previous data at a specific moment and the subsequent data instances vary within a small range, these data instances are termed as jump outliers (e.g., t_3 in Figure 1).

The way of distinguishing collective outliers and jump outliers can be performed by determining whether the number of data samples after the jump exceeds a preset threshold. In other words, the collective outliers mean that the sensor data will revert to the previous mode after a short period when the outliers appear, while the jump outliers mean that the data will reconstruct a new distribution model after the appearance of the outlier and no longer return to the previous distribution mode. The causes of collective outliers and jump outliers are different. In general, the collective outliers may come from multiple continuous measurement errors of the sensor or an occasional short-term event, and the jump outliers usually come from an event rather than an error.

3.2.4. Contextual Outliers. Unlike the above three types of outliers, there is no obvious difference between contextual outliers and normal data. In general, contextual outliers are abnormal data in a specific context, which are also termed as conditional outliers.

4. Proposed Approach

Based on the above definitions and the sources of outliers, we proposed an efficient outlier detection approach based on neighbor difference and clustering, namely ODNDC, which can accurately detect the outliers and identify the sources of outliers. Overall, the primary features of ODNDC can be concluded as follows: (1) can detect outliers in an unsupervised way, (2) can detect outliers through one scan of the collected streaming sensor data, (3) can handle concept evolution, (4) has low time complexity, and (5) can identify the source of outliers. And then, the proposed ODNDC approach is described in detail.

4.1. The Framework of ODNDC Approach. In order to realize the detection and identification of outliers, drawing on the idea of CluStream algorithm [42], we divide the framework of ODNDC approach into online detection phase and offline identification phase, and the specific framework of ODNDC

approach is shown in Figure 2. The online detection phase aims to realize the real-time detection of outliers in the streaming sensor data, which requires the algorithm to have a faster running speed to catch up with the generating of streaming sensor data. The offline phase aims to identify the source of outliers detected at the online detection phase. Since the amounts of outliers are much smaller than that of normal data, it is not necessary to consider more the time consumption in offline identification phase.

The ODNDC approach contains three algorithms, including outlier detection based on neighbor difference (ODND), outlier detection based on clustering (ODC), and outlier sources identification based on correlation (OSIC). The ODND algorithm and ODC algorithm belong to online detection phase and they are used to detect the potential outliers from the streaming sensor data; these two algorithms use neighbor difference to each streaming sensor data on the base of the clustering algorithm to solve the problem of traditional clustering-based outlier detection approach that does not consider the time correlation of the data. The OSIC algorithm belongs to offline identification phase, and it is used to identify the sources of outliers.

4.2. ODND Algorithm. The clustering-based outlier detection approaches are very sensitive to noise and outliers, which results in their detection accuracy not being stable when processing streaming sensor data. In order to eliminate the influence caused by noise and outliers, with the consideration of time correlation in the streaming sensor data, we propose an outlier detection algorithm based on the neighbor difference, namely ODND; it detects the outliers before performing clustering-based outlier detection algorithm. The proposed ODND algorithm not only solves the problem of the clustering-based outlier detection algorithm not considering the time correlation but also eliminates the point outliers, thereby improving the performance of subsequent ODC algorithm.

In the ODND algorithm, the neighbor difference between two adjacent data instances in the streaming sensor data is calculated, and the neighbor difference of sensor data (shown in in Figure 1) is the lower curve in Figure 3. It can be seen from Figure 3 that once outliers appear, the neighbor difference values will have significant changes, and different types of outliers have their own specific features, that is, with

FIGURE 2: The framework of ODND approach.

FIGURE 3: Contrast between sensor data and its neighbor difference, where the sensor data source is from temperature values in Figure 1.

a different neighbor difference value. When a point outlier is appearing (t_1 in Figure 3), the neighbor difference value has consecutive significant changes with opposite signs. When a collective outlier appears (the interval of t_2 in Figure 3), there are several small differences in the adjacent difference value with obvious opposite signs, and the number of these small differences is equal to the number of collective outliers according to its definition. Different from collective outliers, when the number of small differences exceeds a given threshold, the outliers are jump outliers (t_3 in Figure 3). Note that the set of threshold can directly influence the final result of the ODND algorithm. In the WSNs, the sensor data meet the normal distribution in general, so the outliers are mostly distributed in the areas outside 3-standard deviation (marked as 3δ) of average value (marked as μ); therefore, we adopt 3δ -principle [43] in the setting of the threshold, where the threshold is set to $(\mu - 3\delta \leq \text{threshold} \leq \mu + 3\delta)$ of the collected streaming data.

According to the above analysis, the ODND algorithm can use the change of neighbor difference to distinguish point outliers, collective outliers, and jump outliers, but it cannot distinguish contextual points (the contextual points will be detected by ODC algorithm). To reduce the false alarms, the detected collective outliers and jump outliers will be regarded as candidate outliers rather than true outliers, and these two types of outliers will be determined as true outliers when they are also declared as outliers in the ODC algorithm.

Before presenting the specifics of the ODND algorithm, we give some definitions of variables:

- (i) curDiff: the difference between current sensor data and previous one
- (ii) abDiff: the abnormal difference detected for the first time
- (iii) count: a counter used to distinguish collective outliers and jump outliers
- (iv) isAbnormal: a flag representing whether the data are being marked as outliers

The specific operation of the proposed ODND algorithm is shown in Algorithm 1.

Furthermore, we use different processing ways for different types of outliers to keep the original distribution of data as much as possible. Normally, the point outliers will be regarded as true outliers and each point's value will be replaced by mean value of its adjacent points. In contrast, the values of other types of outliers will not be modified in the ODND algorithm.

4.3. ODC Algorithm. The proposed ODND algorithm has the advantages of being simple, effective, and fast, and it can improve the performance of the ODC algorithm. However, it only can be used for detecting potential outliers from single streaming sensor data and cannot

Input: streaming sensor data, threshold
Output: outliers and their types

```

(01) calculate the curDiff
(02) if isAbnormal = true then
(03)   count = count + 1
(04) end if
(05) if curDiff outside threshold then
(06)   if isAbnormal = true the
(07)     if curAbnormal and abDiff has opposite sign then
(08)       if count = 1 then
(09)         current data are labeled as true point outlier
(10)       else
(11)         if count in threshold then
(12)           all data between current data and the data corresponding to abDiff are labeled as candidate collective outliers
(13)         else if count outside threshold then
(14)           the data corresponding to abDiff is labeled as a candidate jump outlier
(15)         end if
(16)       end if
(17)       reset the temporary variables
(18)     end if
(19)   else
(20)     if count in threshold then
(21)       label the data corresponding to abDiff as a true jump outlier
(22)       reset the temporary variables
(23)     end if
(24)   end if
(25) end if

```

ALGORITHM 1: Outlier detection based on neighbor difference (ODND).

Input: the data labeled by ODND
Output: outliers and their types

```

(01) use a sliding window model to receive data from all sensors
(02) for each point outlier detected by ODND do
(03)   replace them by their mean values calculated with adjacent data instances
(04) end for
(05) detect outliers using w-k-means algorithm
(06) Reassign the labels of all outliers detected by ODND and w-k-means algorithm

```

ALGORITHM 2: Outlier Detection based on Clustering (ODC).

detect contextual outliers. To effectively detect the outliers in multiple streaming sensor data and detect contextual outliers, an outlier detection algorithm based on clustering, namely ODC, is added after ODND algorithm. In the WSNs, there is no obvious correlation between most streaming sensor data. However, the existence of irrelevant data interferes and conceals the true clustering structure of k-means algorithm, which reduces the efficiency of outlier detection and leads to high false alarms. Different from k-means clustering algorithm, w-k-means algorithm [40] assigns different weights to each attribute according to the correlation between attributes, which can reduce the impact of irrelevant attributes on the clustering results and thus improving the detection performance. Therefore, the w-k-means algorithm is used in ODC algorithm to improve the detection efficiency to a greater extent.

The proposed ODC algorithm contains three blocks, including updating block, clustering block, and outlier detection block. (1) In the update block, k samples are randomly selected as initial cluster center firstly and each object in the data block is assigned to the nearest cluster. And then, the objects in small clusters are regarded as candidate outliers. In order to avoid the influence of outliers on the clustering effect, candidate outliers are not used but the remaining objects are used to calculate the cluster centers and weights. This processing can effectively solve the evolution problem of streaming sensor data. (2) In the clustering block, the w-k-means algorithm [40] is adopted, which reduces the influence of noise attributes on the clustering performance by calculating and assigning a weight to each attribute. (3) In the outlier detection block, the outlier score of objects that are identified as candidate outliers for the first time is set to 1, and then enter to the next

cycle to determine whether this candidate outlier would be detected as an outlier. If the candidate outlier is detected as an outlier again, the outlier score is updated. Within the set period, when the outlier score of a candidate outlier is equal to L , it will be regarded as a true outlier; otherwise, it will be regarded as a normal sample.

The specific operation of the proposed ODC algorithm is shown in Algorithm 2.

The difference between ODC and ODND algorithms is that ODND uses time correlation between the historical data to realize outlier detection of single streaming sensor data, which runs fast but does not consider the spatial correlation between multiple sensors; The ODC algorithm is based on a w-k-means clustering algorithm and uses the spatial correlation between multiple sensors to realize outlier detection. The difference between ODC algorithm and w-k-means clustering algorithm is that ODC algorithm not only needs to detect outliers based on w-k-means clustering algorithm but also needs to jointly determine the outliers detected in the ODND algorithm.

The rules for re-identification of outliers are as follows:

- (1) The point outlier detected in the ODND algorithm usually means that the point is different from isolated data points around it. It is usually caused by operating errors or instant errors of sensor, thus, the point outlier marked by the ODND algorithm is still maintained in the ODC algorithm, and the point outlier is processed before the ODC algorithm, while the processing method is usually to take the average or median value of adjacent data.
- (2) The collective outlier detected in the ODND algorithm may be caused by the sensor error or physical quantities in the objective world in which appear short-term temporary changes, and it is unable to determine the cause of this type of outliers on a single sensor. In order to reduce the false alarms, the collective outliers need to be cleverly detected by the ODND algorithm combined with other sensor variables. Therefore, if most of the data in the ODND algorithm marked as continuous collective outliers are still marked as outliers in the ODC algorithm, then, these collective outliers are determined as true outlier; otherwise, they are considered as normal data.
- (3) The data that are detected as outliers by the ODC algorithm but detected as normal data by the ODND algorithm usually mean that the data on a single sensor is not significantly different, but in the data appear changes when the data of other sensors is considered comprehensively. Therefore, this type of outliers is determined as contextual outliers in the ODC algorithm.
- (4) The jump outlier detected in the ODND algorithm usually means the data generated by a certain sensor have a sudden change as well as other sensors. If the candidate jump outliers detected in the ODND algorithm are not regarded as outliers in the ODC algorithm, the candidate outliers are marked as jump outliers; otherwise, they will be marked as collective outliers.

4.4. OSIC Algorithm. In the ODND approach, the sources of detected outliers will be identified offline using an outlier sources identification algorithm based on correlation, namely OSIC.

In the WSNs, multiple types of sensor nodes are usually deployed at a high density in a certain space to achieve the coverage. Due to high-density distribution of sensor nodes, the environment perceived by sensor nodes deployed in adjacent locations of the space is also similar, which results in the data having a certain spatial correlation. Therefore, the recognition of events can be effectively realized with the use of mutual cooperation of spatially correlated sensor nodes, where the rules for identifying the sources of outliers are defined as follows:

- (1) The source of point outliers is usually marked as an error. Because the point outlier is a data point isolated from other surrounding samples, and it is deviating much from data samples surrounding it, the point outlier is usually caused by operating errors or sensor noise.
- (2) The source of jump outliers is usually marked as an event. The occurrence of a jump outlier usually means that the physical quantity of the objective world appears as a real change, and it also may be caused by a long-term failure of sensors. These two situations require user's attention. The former requires the user to use other domain knowledge to further determine what is happening at the monitoring site, and the latter requires the user to further determine whether the sensor needs maintenance. The long-term sensor failure is regarded as an event in the OSIC algorithm. On the contrary, the short-term sensor failure can be quickly recovered by itself and usually without special treatment, as it is regarded as an error in OSIC algorithm.
- (3) The source of collective outlier may be an error or an event, which requires further judgment. In the WSNs, sensors usually adopt a redundant design to adapt complex environment and improve its stability and reliability, that is, the measurement of the same physical quantity in the same area usually uses multiple sensors with the same type. Because the occurrences of collective outliers can be caused by either errors or events, it is necessary to identify them by significant correlations among sensors variables. There are two categories of significant correlations [44]: (1) One is the spatial neighbor sensors with the same type. For example, the temperature variable of adjacent sensors is usually very similar. If the streaming data of one sensor changes and is quite different from other sensors, this sensor may have functional failures, and it should be marked as an error. If the error remains longer than a given time, then it should be marked as an event. In this manner, the sensor with long-time error should be maintained, while the sensor with short-time error will recover soon automatically. If all streaming data of

temperature variables of adjacent sensors change a lot, an event may occur. (2) The other category of significant correlate sensor variables is several spatial neighbor sensors with different types. For example, the readings of several temperature sensors and several humidity sensors within a certain geographical space have a good correlation. When the temperature in the real world rises, the humidity usually falls correspondingly, and vice versa. Compared with the second category of correlation, the first correlation is easier to judge the cause of outliers. Therefore, in the OSIC algorithm, the first selection for identifying the source of outliers is the first category of correlation.

The criterion for distinguishing the source of outliers is the correlation coefficient of sensor variables. In the ODNDC approach, if the value of correlation coefficient exceeds a threshold of th_1 , it means that all sensors belong to neighbor nodes with the same type; then, a majority voting method is adopted to identify the sources of outliers. If the absolute value of correlation coefficient exceeds a threshold of th_2 , we use a predictive algorithm to predict the mean and the variance of normal data. If a data instance exceeds the range, it will be declared as an error; otherwise, it will be declared as a suspected event, which means that an event may have occurred. In the OSIC algorithm, a support vector regression algorithm with 10-folder cross validation is used to predict the mean and variance.

Note that the choice of thresholds th_1 and th_2 can be done in two ways. One way is to manually set the fix values of th_1 and th_2 , which are based on experts' advice and the implementation of sensors in real life. The advantage of this method is very simple and effective, while the disadvantage is that the accuracy is quite sensitive to preset values. For example, if the thresholds of th_1 and th_2 have been set improperly, the result of this algorithm is inaccurate. Another way is to choose the correlation value of sensors that are in top rank as th_1 and th_2 to automatically determine them. However, the second method is inapplicable if the number of correlation sensors is very less. Therefore, the thresholds of th_1 and th_2 should be chosen based on the actual situation of the sensor deployment.

The design of the OSIC algorithm is shown in Algorithm 3.

5. Experimental Results

In the experiments, we use two datasets to evaluate the efficiency of the proposed ODNDC approach.

5.1. The Description of Datasets. The first dataset is a synthetic dataset, as shown in Figure 1. The second dataset is a real dataset that is collected at intervals of 1 minute during the period from November 1 to November 30 in 2020 at an experimental jujube yard; we only selected 43154 data instances (shown in Figure 4) in the collected data. The sensor variables in the second dataset include temperature,

humidity, and atmospheric pressure, and the specific information of outliers marked by the experts is shown in Table 1.

5.2. Experimental Results. Firstly, we use the synthetic dataset to test the proposed ODNDC approach. The results in Figure 5 show that the point outlier, collective outlier, and jump outlier are correctly detected, where the source of point outlier is labeled as an error, and the source of jump outlier is labeled as an event. Because the dataset only has temperature variable, there are no relative variables that can be used to further distinguish the source of collective variables; thus, the source of collective outlier is labeled as unknown.

And then, we use the real dataset to test the ODNDC approach, and the experimental results are shown in Figure 6, where the specific detection accuracy of the ODNDC approach is listed in Table 2.

It can be seen from Figure 6 that through the use of our proposed ODNDC approach, there are 5 point outliers, 284 collective outliers, 7 jump outliers, and 27 contextual outliers that can be detected in the temperature variable of the sensor data; 15 point outliers, 240 collective outliers, 14 jump outliers, and 27 contextual outliers can be detected in the humidity variable of the sensor data; and 146 collective outliers, 2 jump outliers, and 27 contextual outliers can be detected in the atmospheric pressure variable of the sensor data. We can see from Table 2 that the detection accuracy of the proposed ODNDC approach is relatively high, and it also can accurately distinguish different types of outliers with the use of neighbor difference, w-k-means algorithm, and re-identification rules.

To further evaluate the proposed ODNDC approach, we use four outlier detection approaches, including LODA [35], ROCF [37], CORM [38], and Heuristic algorithm [44], to test the detection efficiency, where the real dataset collected at the experimental jujube yard is used in the experiment, and the experimental result is shown in Table 3.

It can be seen from Table 3 that the detection accuracy of the proposed ODNDC approach is the highest, while the detection accuracy of the CORM is the lowest. In these five compared approaches, the proposed ODNDC and compared Heuristic can identify the sources of outliers, and the identifying accuracy of the ODNDC is higher than that of the Heuristic. The reason for the high detection accuracy of the ODNDC approach is that it first uses neighbor difference to detect candidate outliers to reduce the influence of outliers on clustering results, and then uses spatial correlation and re-identification rules to detect different types of outliers. In contrast, the LODA, ROCF, and CORM approaches can only determine whether the data instances are outlier but cannot effectively identify the sources of outliers.

And then, the identification of the sources of outliers by the ODNDC approach on the temperature of the collected real dataset in two time windows (on November 23 and November 26) is conducted in the experiment, and the experimental result is shown in Figure 7, where the blue line represents the data collected by sensors.

Input: the streaming sensors data after outlier detection
Output: outliers and their sources

```

(01) if there is no outlier in current window then
(02)   exit
(03) else
(04)   if the data instance is detected as point outlier then
(05)     label its source as error
(06)   end if
(07)   if the data instance is detected as jump outlier then
(08)     label its source as event
(09)   end if
(10)   if the data instance is detected as collective outlier or contextual outlier then
(11)     calculate the correlation coefficient
(12)     if correlation coefficient >  $th_1$  then
(13)       if more than half of attributes are labeled as outliers then
(14)         label its source as event
(15)       else
(16)         label its source as error
(17)       end if
(18)     else if |correlation coefficient| >  $th_2$  then
(19)       read these correlative variables of data instances
(20)       predict the mean and variance of normal values with 10-folder cross validation
(21)       if the outlier is out of the predicted range then
(22)         label its source as error
(23)       else
(24)         label its source as suspected event
(25)       end if
(26)     end if
(27)   else
(28)     label the source of collective outliers as unknown
(29)     label the source of contextual outliers as normal
(30)   end if
(31) end if

```

ALGORITHM 3: Outlier sources identification based on correlation (OSIC).

FIGURE 4: Continued.

FIGURE 4: The data collected at intervals of 1 minute during November 2020 at an experimental jujube yard. (a) Temperature, (b) humidity, (c) atmospheric pressure.

TABLE 1: The specific information of contained outliers.

Number	Variables		
	Temperature	Humidity	Atmospheric pressure
Point outliers	6	16	0
Collective outliers	303	253	154
Jump outliers	7	15	2
Contextual outliers	28	29	29

FIGURE 5: The result of outlier detection for the synthetic dataset.

It can be seen from Figure 7 that with the use of our proposed ODND approach, the sources of outliers, including error and event, are effectively identified. In addition, the contexture outliers can be identified as “suspected event” by our proposed ODND approach, which can prevent the identification error. The experimental result verifies that the use of correlation coefficient of sensor variables is beneficial for identifying the sources of outliers, and the designed identifying rule also improves the identifying efficiency.

Because the OSIC algorithm runs in the off-line mode, and the number of outliers is far less than that of normal instances, it has little effect on the time complexity of the ODND approach. To evaluate this aspect, we test the running time of the

ODND with the familiar w-k-means algorithm, the LODA, ROCF, CPRM, and Heuristic approaches on the real dataset, and the experimental results are shown in Figure 8.

It can be seen from Figure 8 that in the six approaches, the time cost of the LODA is the longest, and the time cost of the ROCF is the second longest, while the time cost of the w-k-means algorithm is the shortest. For the proposed ODND approach, its time cost is slightly longer than that of the w-k-means algorithm. This is because the ODND approach needs to calculate the neighbor difference before performing clustering operation to reduce the influence of outliers on clustering results. Compared with other four outlier detection approaches, the time cost of the ODND is much shorter; it indicates that

FIGURE 6: The results of outlier detection by ODNDC approach. (a) Temperature, (b) humidity, (c) atmospheric pressure.

TABLE 2: The detection accuracy of the ODNDC approach.

Types	Variables		
	Temperature (%)	Humidity (%)	Atmospheric pressure
Point outliers	83.3	93.8	—
Collective outliers	93.7	94.9	94.8%
Jump outliers	100	93.3	100%
Contextual outliers	96.4	93.1	93.1%

TABLE 3: The comparisons of detection accuracy

Types	Variables				
	ODNDC (%)	LODA	ROCF	CORM	Heuristic (%)
Point outliers	90.9	—	—	—	86.4
Collective outliers	94.4	—	—	—	92.5
Jump outliers	95.8	—	—	—	87.5
Contextual outliers	94.2	—	—	—	90.7
Overall	94.3	93.6%	92.8%	89.5%	92

FIGURE 7: Two cases of outlier source identification in two time windows. (a) Nov-23, and (b) Nov-26.

FIGURE 8: Runtime of the compared outlier detection algorithm and the ODND approach with different sizes of the time window.

the time efficiency of the ODND approach is very competitive. Compared with the w-k-means algorithm, although the time cost is slightly longer, the proposed ODND approach can accurately detect the outliers and distinguish the sources of outliers; thus, we can ignore the small amount of extra time consumption. The experimental result verifies whether the ODND is a time-efficient approach, which can apply to outlier detection for streaming sensor data.

6. Conclusions

In this paper, we propose a new approach of the ODND, which can detect the outliers in streaming sensor data and identify the sources of outliers based on the analysis of the data pattern. The ODND approach is composed of ODND, ODC,

and OSIC, where the outliers are detected and labeled by the ODND algorithm and ODC algorithm, and the sources of outliers are identified by the OSIC algorithm. In addition, the ODND algorithm also focuses on the temporal relationship of each streaming sensor data, and the ODC algorithm focuses on spatial relationship of all streaming sensor data, which solves the issue of the common clustering-based outlier detection algorithms neglecting temporal relationship of streaming sensor data. The experimental results show that the proposed ODND approach can accurately detect potential outliers from the collected data from WSNs, as well as with high time efficiency. Therefore, the proposed ODND approach can be effectively used in outlier detection in the environment of WSNs, thereby providing an accurate and reliable data for edge computing.

Although the use of the w-k-means algorithm can reduce the impact of irrelevant attributes on the clustering results, its clustering results depend on the initial cluster centers and the initial weights. In the future, we would like to use the coefficient of variation or entropy to select suitable initial weights to further improve the clustering efficiency; in addition, we also would like to study the patterns of abnormal sensors in specific domain and try to use the approach in early warning for sensors.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partly supported by National Natural Science Foundation of China (Grant numbers: U1836116 and 62172194), the China Postdoctoral Science Foundation (Grant number: 2021M691310), the Future Network Scientific Research Fund Project (Grant number: FNSRFP-2021-YB-50), the Postdoctoral Science Foundation of Jiangsu Province (Grant number: 2021K636C), the Natural Science Foundation of the Jiangsu Higher Education Institutions (Grant number: 21KJB520031), the Graduate Research Innovation Project of Jiangsu Province (Grant numbers: KYCX21_3375 and SJCX21_1692), and the College Student Innovation and Entrepreneurship Training Program (Grant number: 202110299178Y).

References

- [1] H. Liu and K. Y. Ki, "Application of wireless sensor network based improved immune gene algorithm in airport floating personnel positioning," *Computer Communications*, vol. 160, pp. 494–501, 2020.
- [2] A. Farhat, A. Jaber, R. Tawil, C. Guyeux, and A. Makhoul, "On the coverage effects in wireless sensor networks based prognostic and health management," *International Journal of Sensor Networks*, vol. 28, no. 2, pp. 125–138, 2018.
- [3] B. Cao, J. Zhao, Y. Gu, S. Fan, and P. Yang, "Security-aware industrial wireless sensor network deployment optimization," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5309–5316, 2020.
- [4] L.-L. Shi, L. Liu, Y. Wu et al., "Human-centric cyber social computing model for hot-event detection and propagation," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 1042–1050, 2019.
- [5] L.-L. Shi, L. Liu, Y. Wu, L. Jiang, J. Panneerselvam, and R. Crole, "A social sensing model for event detection and user influence discovering in social media data streams," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 1, pp. 141–150, 2020.
- [6] L.-L. Shi, L. Liu, Y. Wu, L. Jiang, and A. Ayorinde, "Event detection and multi-source propagation for online social network management," *Journal of Network and Systems Management*, vol. 28, no. 1, pp. 1–20, 2020.
- [7] X. Li, T. Chen, Q. Cheng, S. Ma, and J. Ma, "Smart applications in edge computing: overview on authentication and data security," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4063–4080, 2021.
- [8] L. Ullah, M. S. Khan, M. St-Hilaire, and M. Faisal, "Task priority-based cached-data prefetching and eviction mechanisms for performance optimization of edge computing clusters," *Security and Communication Networks*, vol. 2021, Article ID 5541974, 10 pages, 2021.
- [9] S. Cai, R. Huang, J. Chen et al., "An efficient outlier detection method for data streams based on closed frequent patterns by considering anti-monotonic constraints," *Information Sciences*, vol. 555, pp. 125–146, 2021.
- [10] M. Radovanovic, A. Nanopoulos, and M. Ivanovic, "Reverse nearest neighbors in unsupervised distance-based outlier detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1369–1382, 2015.
- [11] S. Cai, J. Chen, H. Chen et al., "An efficient anomaly detection method for uncertain data based on minimal rare patterns with the consideration of anti-monotonic constraints," *Information Sciences*, vol. 580, pp. 620–642, 2021.
- [12] J. Fan, Q. Zhang, J. Zhu, M. Zhang, Z. Yang, and H. Cao, "Robust deep auto-encoding Gaussian process regression for unsupervised anomaly detection," *Neurocomputing*, vol. 376, pp. 180–190, 2020.
- [13] E. K. Boahen, B. E. Bouya-Moko, and C. Wang, "Network anomaly detection in a controlled environment based on an enhanced PSOGSARFC," *Computers & Security*, vol. 104, p. 102225, 2021.
- [14] C. Titouna, F. Naït-Abdesselam, and A. Khokhar, "DODS: a distributed outlier detection scheme for wireless sensor networks," *Computer Networks*, vol. 161, pp. 93–101, 2019.
- [15] Ł. Saganowski, T. Andrysiak, R. Kozik, and M. Choraś, "DWT-based anomaly detection method for cyber security of wireless sensor networks," *Security and Communication Networks*, vol. 9, no. 15, pp. 2911–2922, 2016.
- [16] N. Nesa, T. Ghosh, and I. Banerjee, "Non-parametric sequence-based learning approach for outlier detection in IoT," *Future Generation Computer Systems*, vol. 82, pp. 412–421, 2018.
- [17] O. Iraqi and H. El Bakkali, "Application-level unsupervised outlier-based intrusion detection and prevention," *Security and Communication Networks*, vol. 2019, pp. 1–13, 2019.
- [18] E. X. Min, J. Long, Q. Liu, J. J. Cui, and W. Chen, "Tr-I. D. S.: Anomaly-based intrusion detection through text-convolutional neural network and random forest," *Security and Communication Networks*, vol. 2018, Article ID 4943509, 9 pages, 2018.
- [19] B. Saneja and R. Rani, "An efficient approach for outlier detection in big sensor data of health care," *International Journal of Communication Systems*, vol. 30, no. 17, pp. 1–10, 2017.
- [20] K. Edward, C. D. Wang, and B. Bouya-Moko, "Detection of compromised online social network account with an enhanced knn," *Applied Artificial Intelligence*, vol. 34, no. 11, pp. 777–791, 2020.
- [21] X. Sun, "Similarity detection method of abnormal data in network based on data mining," *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 1, pp. 155–162, 2020.
- [22] K. Thangaramya, K. Kulothungan, S. Indira Gandhi, M. Selvi, S. V. N. Santhosh Kumar, and K. Arputharaj, "Intelligent fuzzy rule-based approach with outlier detection for secured routing in WSN," *Soft Computing*, vol. 24, no. 21, pp. 16483–16497, 2020.

- [23] F. Angiulli, S. Basta, S. Lodi, and C. Sartori, "GPU strategies for distance-based outlier detection," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 11, pp. 3256–3268, 2016.
- [24] B. Tang and H. He, "A local density-based approach for outlier detection," *Neurocomputing*, vol. 241, pp. 171–180, 2017.
- [25] E. Bigdeli, M. Mohammadi, B. Raahemi, and S. Matwin, "Incremental anomaly detection using two-layer cluster-based structure," *Information Sciences*, vol. 429, pp. 315–331, 2018.
- [26] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Hyper-spherical cluster based distributed anomaly detection in wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 74, no. 1, pp. 1833–1847, 2014.
- [27] O. Ghorbel, M. W. Jmal, W. Ayedi, H. Snoussi, and M. Abid, "An overview of outlier detection technique developed for wireless sensor networks," in *Proceedings of the International 10th Multi-Conference on Systems, Signals & Devices (SSD)*, pp. 1–6, IEEE, Hammamet, Tunisia, March 2013.
- [28] A. Alkhatib and Q. Abed-Al, "Multivariate outlier detection for forest fire data aggregation accuracy," *Intelligent Automation & Soft Computing*, vol. 31, no. 2, pp. 1071–1087, 2022.
- [29] S. A. N. Nozad, M. A. Haeri, and G. Folino, "SDCOR: scalable density-based clustering for local outlier detection in massive-scale datasets," *Knowledge-Based Systems*, vol. 228, p. 107256, 2021.
- [30] D. Krleza, B. Vrdoljak, and M. Brcic, "Statistical hierarchical clustering algorithm for outlier detection in evolving data streams," *Machine Learning*, vol. 110, no. 1, pp. 139–184, 2021.
- [31] Y. Zhang, N. A. S. Hamm, N. Meratnia, A. Stein, M. van de Voort, and P. J. M. Havinga, "Statistics-based outlier detection for wireless sensor networks," *International Journal of Geographical Information Science*, vol. 26, no. 8, pp. 1373–1392, 2012.
- [32] S. Bharti, K. K. Pattanaik, and A. Pandey, "Contextual outlier detection for wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 4, pp. 1511–1530, 2020.
- [33] I. G. A. Poornima and B. Paramasivan, "Anomaly detection in wireless sensor network using machine learning algorithm," *Computer Communications*, vol. 151, pp. 331–337, 2020.
- [34] A. De Paola, S. Gaglio, G. Lo Re, F. Milazzo, and M. Ortolani, "Adaptive distributed outlier detection for WSNs," *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 888–899, 2015.
- [35] M. Safaei, A. S. Ismail, H. Chizari et al., "Standalone noise and anomaly detection in wireless sensor networks: a novel time-series and adaptive Bayesian-network-based approach," *Software: Practice and Experience*, vol. 50, no. 4, pp. 428–446, 2020.
- [36] J. A. Lara, D. Lizcano, V. Ramperez, and J. Soriano, "A method for outlier detection based on cluster analysis and visual expert criteria," *Expert Systems*, e12473, vol. 37, no. 5, , 2019.
- [37] J. Huang, Q. Zhu, L. Yang, D. Cheng, and Q. Wu, "A novel outlier cluster detection algorithm without top-n parameter," *Knowledge-Based Systems*, vol. 121, pp. 32–40, 2017.
- [38] M. Elahi, K. Li, W. Nisar, X. Lv, and H. Wang, "Efficient clustering-based outlier detection algorithm for dynamic data stream," in *Proceedings of the International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 298–304, Spring, Xi'an, China, 19 december 2008.
- [39] N. Randive, "Sneha. Hybrid approach for outlier detection in high dimensional dataset," *International Journal of Science and Research*, vol. 3, no. 7, pp. 1743–1746, 2014.
- [40] J. Z. Huang, M. K. Ng, H. Hongqiang Rong, and Z. Zichen Li, "Automated variable weighting in k-means type clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 657–668, 2005.
- [41] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.
- [42] C. C. Aggarwal, P. S. Yu, J. Han, J. Wang, and R. Ctr, "A framework for clustering evolving data streams," in *Proceedings of the 2003 VLDB Conference. In: Proceedings of VLDB Conference*, pp. 81–92, Elsevier, 18 September 2003.
- [43] E. W. Grafarend, *Linear and Nonlinear Models: Fixed Effects, Random Effects, and Mixed Models*, Walter de Gruyter, Berlin, New York, 2006.
- [44] A. F. Hassan, H. M. O. Mokhtar, O. Hegazy, A. F. Hassan, and O. Hegazy, "A heuristic approach for sensor network outlier detection," *International Journal of Research and Reviews in Wireless Sensor Networks*, vol. 1, no. 4, pp. 66–72, 2011.