

Research Article

A Real-Time Detection Method for Abnormal Data of Internet of Things Sensors Based on Mobile Edge Computing

Xuguang Liu 

Department of Urban Rail Transit and Information Engineering Anhui Communications Vocational and Technical College, Hefei 230051, Anhui, China

Correspondence should be addressed to Xuguang Liu; liuxg1973@ahszu.edu.cn

Received 18 December 2020; Revised 7 February 2021; Accepted 22 February 2021; Published 28 February 2021

Academic Editor: Yi-Zhang Jiang

Copyright © 2021 Xuguang Liu. 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.

Aiming at the anomaly detection problem in sensor data, traditional algorithms usually only focus on the continuity of single-source data and ignore the spatiotemporal correlation between multisource data, which reduces detection accuracy to a certain extent. Besides, due to the rapid growth of sensor data, centralized cloud computing platforms cannot meet the real-time detection needs of large-scale abnormal data. In order to solve this problem, a real-time detection method for abnormal data of IoT sensors based on edge computing is proposed. Firstly, sensor data is represented as time series; K-nearest neighbor (KNN) algorithm is further used to detect outliers and isolated groups of the data stream in time series. Secondly, an improved DBSCAN (Density Based Spatial Clustering of Applications with Noise) algorithm is proposed by considering spatiotemporal correlation between multisource data. It can be set according to sample characteristics in the window and overcomes the slow convergence problem using global parameters and large samples, then makes full use of data correlation to complete anomaly detection. Moreover, this paper proposes a distributed anomaly detection model for sensor data based on edge computing. It performs data processing on computing resources close to the data source as much as possible, which improves the overall efficiency of data processing. Finally, simulation results show that the proposed method has higher computational efficiency and detection accuracy than traditional methods and has certain feasibility.

1. Introduction

In recent years, with the continuous development and integration of technologies such as the Internet, IoT, and cloud computing, a large number of sensor devices have been widely used in different fields such as power systems and thermal systems [1, 2]. Usually, sensors collect data at a certain frequency and send data to corresponding data receivers. The data receiver receives one or more sets of observations in a strict order; these observation data are basically time series data [3]. Time series data accurately records the real-time changes of a certain parameter and reflects trends and change law within a certain time range. Therefore, the time series data collected by sensor devices are not only an important data source for data visualization but also the basis of data mining (such as classification, prediction, clustering, and association) [4–6]. However, there

will always be some abnormalities in the data collection and transmission processes of sensor equipment, such as error codes and sensor failures in actual data collection scenarios [7, 8]. Thus, in order to provide high-quality source data for subsequent data mining research, it is very necessary to effectively identify outliers in sensor data from the perspective of time series data analysis.

According to the wireless sensor network (WSN) characteristics, anomaly detection methods are divided into statistics-based, classification-based, clustering-based, and neighbor-based methods. Single-sensor data streams usually use the time correlation of data for anomaly detection, and many applications are based on statistical analysis and nearest neighbor distance for anomaly detection. Multi-sensor data streams have both time and space correlation, and cluster-based methods are usually used for detection. The time correlation of sensors is often ignored in the

clustering process. For example, reference [9] proposed a new anomaly detection algorithm for time series data by constructing a distributed recursive computing strategy and KNN quick selection strategy. Reference [10] proposed a clustering algorithm that used local parameters for unbalanced data to detect abnormal data. Reference [11] applied the K-means algorithm to cluster analysis of iris data with 5 attributes. Compared with traditional methods, outlier removal clustering (ORC) technology achieved better results. Reference [12] was based on Spatiotemporal (ST) correlation and detected outliers by calculating the cross-correlation between sensor data streams. However, traditional detection algorithms mainly focus on the sequence continuity of single-source sensor data and ignore the correlation between multisource sensor data. In addition, it needs to be particularly emphasized that the current common sensing data anomaly detection and processing method use relatively mature cloud computing models and common big data processing products to directly transmit the data obtained by various data collection devices to cloud computing center for processing and storage. And the powerful computing power of the cloud computing center was used to complete corresponding anomaly detection and data cleaning work [13]. Although there may not be a clear correlation between time series data from different sensors, its inherent characteristics may have a high correlation. If these data are uploaded to the data center for feature extraction, it will cause a lot of computational pressure on the data center. Since underlying devices, including sensors, have certain computing capabilities, hidden features can be extracted.

In order to improve the detection accuracy of sensor abnormal data, a real-time detection method of abnormal data based on edge computing is proposed to better meet the real-time detection requirements of large-scale abnormal data. The parameters use nonmutation characteristics of time series data, and the improved DBSCAN algorithm uses the spatial correlation characteristics of multidimensional data in the KNN algorithm. The proposed algorithm fully considers data relevance and can effectively mine their potential relationship. Furthermore, a distributed anomaly detection model for sensor data based on edge computing is proposed to process data on computing resources close to the data source as much as possible. Experiments have proved that the proposed algorithm has positive significance for improving algorithm detection accuracy and the overall data processing efficiency.

2. Sensor Data Modeling Based on Time Series

Definition 1. The multisource data collected and transmitted by sensors can be expressed as the following time series data:

$$\begin{aligned} TD_M &= \{D_1, D_2, \dots, D_i, \dots, D_M\}, \\ D_i &= \{d_1, d_2, \dots, d_j, \dots, d_N\}, \end{aligned} \quad (1)$$

where $1 \leq i \leq M$, $1 \leq j \leq N$, and TD_M are the time series data sets of multisource data, and M is multisource number. D_i in equation (2) is the data set of the data source i . D_i in equation (3) is the length of N , and d_j is the sensed value at time t_j .

Based on the single data source representing D_i , a sliding window (SW) is introduced to store part of the data D_i . Therefore, the length of SW is expressed as $\text{Length}_{\text{SW}}$.

Definition 2. According to some known correlations in the multisource data set $TD_M = \{D_1, D_2, \dots, D_i, \dots, D_M\}$, perform the necessary combination and transformation of TD_M to obtain a new time series (denoted as $T D'_K$). And enter it into the so-called related parameter set $\Omega_K = \{D'_1, D'_2, \dots, D'_K\}$. Then, abnormal data can be found by detecting the linear correlation of Ω_K in the data correlation detection (DCD) process.

According to Definition 1, from the linear correlation of TD_M in same SW, the correlation of TD_M can be realized. Considering that there may not be linear correlation or nonlinear correlation in TD_M , it is necessary to convert TD_M into a multisource signal TD'_K with linear correlation characteristics for subsequent DCD processing. Here are the three correlations that exist:

- (1) *Basic Correlation.* It can also be called linear correlation. Taking the power system as an example, a binary time series $TD_2 = \{P, f\}$ composed of generator active power output P and grid frequency f is defined, where $P = \{p_1, p_2, \dots, p_n\}$, $f = \{f_1, f_2, \dots, f_n\}$ ($1 \leq i \leq n$). Due to the droop characteristic of the power system, the active output power P and grid frequency f satisfy a binary linear correlation, namely $p_i = k f_i + b$. Thus, let $\Omega_2 = \{P, f\}$ be the binary related parameter set. Similarly, if ternary time series data $TD_3 = \{D_1, D_2, D_3\}$ satisfies ternary linear correlation, the corresponding set of related parameters can be given by $\Omega_3 = \{D_1, D_2, D_3\}$.
- (2) *Combination Correlation.* This shows that there is no linear correlation of a single time series D_i in a given time series TD_M , but there is a linear correlation after combining D_i . Taking the thermal system as an example, a ternary time series $TD_3 = \{Q, W, t\}$ composed of thermal power Q , instantaneous temperature observation value W , and time t is defined. However, according to the basic heat power theorem in thermodynamics, there is a positive linear relationship between heat Q and temperature change rate W (i.e., $\Delta W / \Delta t$). In other words, $p_i = k_1 \Delta W_i / \Delta t_i + k_2$, where $\Delta W_i = W_i - W_{i-1}$, $\Delta t_i = t_i - t_{i-1}$, k_1 and k_2 are coefficients. Through a certain combination, the time series in the original $TD_3 = \{Q, W, t\}$ can be converted into a binary linear model. The corresponding binary time series $TD'_2 = \{P, \Delta W / \Delta t\}$ is put into a parameter set Ω_2 , denoted as $\Omega_2 = \{P, \Delta W / \Delta t\}$.
- (3) *Conversion Correlation.* That is, in a given TD_M , there is no basic correlation or combination correlation in time series, but there is a nonlinear correlation, such as exponential model, hyperbolic model, polynomial model, etc. For example, flow optimization coefficients m and b of radiator satisfy

the hyperbolic relationship: $m = 1/(1 + b)$ in a thermal system. Another example is that kinetic energy E and the angular velocity ω of the generator rotor satisfy a polynomial relationship: $E = J\omega^2/2$ in the power system. Through some data conversion methods, nonlinear correlation models can be transformed into linear correlation models.

3. Proposed Real-Time Detection Algorithm for Abnormal Sensor Data

Due to the huge amount of sensor data in IoT, traditional centralized cloud computing framework may have low efficiency in solving detection algorithms when computing resources are limited. Therefore, a detection framework based on edge computing is proposed to detect abnormal sensor data.

3.1. Real-Time Detection Framework of Sensor Data Based on Edge Computing. The linear growth of centralized cloud computing power has been unable to meet the rapid growth of data processing needs for edge devices [14]. Besides, from a technical or economic point of view, it is unlikely that the ever-growing edge data will be concentrated in one or more data computing centers to complete corresponding computing tasks.

In the edge computing framework, computing tasks are assigned to many distributed devices with certain computing capabilities [15–18]. Therefore, computational efficiency can be improved while reducing the performance requirements of computing equipment. So, an edge computing architecture is built to detect abnormal sensor data in real time. As shown in Figure 1, the corresponding edge layer data node is established near the sensor data collection terminals to complete the detection task of related data while receiving sensor data.

Edge computing function is the core function of this system. In this paper, the main content of edge computing is abnormal data detection, estimation, and correction, and other edge computing tasks can also be added to this functional module according to actual needs. The realization of the edge computing module mainly realizes the functions of sequence generation, anomaly detection, and correction: retrieve the configuration information in the database and record the parameters that need to be edged. And the data to be processed is divided into different sequences according to different parameters to facilitate subsequent data processing. Then, perform anomaly detection and estimation correction algorithms on the corresponding sensor data sequence, mark abnormal data, and add estimated correction values.

3.2. Outlier and Isolated Group Detection Based on KNN Algorithm. The basic rule of the KNN algorithm is to find the c nearest neighbors of N samples (where $1 < k < N$). When $c = 1$, KNN problem is equivalent to the nearest neighbor problem. $c_i \in N$ represents the number of sensor data belonging to the i category. In general, the judgment rule for judging which type of sensor data belongs to is the

voting principle. In addition, set c to an odd number to avoid divergence caused by equal votes. The voting principle can be expressed mathematically as follows:

$$j = \arg \max_i g_i(\mathbf{x}), \quad (2)$$

where x is the sampled data, and $g_i(x)$ is the number of sensors belonging to the i category.

The steps of the KNN algorithm are as follows:

(1) *Distance Calculation.* For a given sensing data set, calculate the distance between each object in the training set. In this paper, Euclidean distance is used as follows.

For two-dimensional vectors $a(x_{11}, x_{12}, \dots, x_{1n})$ and $b(x_{21}, x_{22}, \dots, x_{2n})$, Euclidean distance is given as follows:

$$d_{12} = \sqrt{\sum_{k=1}^n (x_{1k} - x_{2k})^2}, \quad (3)$$

where d_{12} is Euclidean distance between sensor data a and b .

(2) *Neighborhood Discovery.* The nearest c training objects are identified as the nearest neighbors of test objects.

(3) *Classification.* The test objects are classified according to the main categories of the above c neighbors.

3.3. Abnormal Sensor Data Detection Based on DBSCAN Algorithm. The basic DBSCAN algorithm uses globally unique parameters Eps and MinPst to achieve clustering. Inspired by reference [10] using local parameters for clustering, this paper proposes a method based on SW data partition, uses local parameters to achieve density clustering of small sample data. The algorithm flow is shown in Figure 2.

The clustering process consists of three parts, namely parameter update, clustering, and anomaly detection. During the parameter update process, set the size of the clustering window and calculate the average distance difference between attributes in the window, take K as the number of points in the neighborhood MinPst, and Euclidean distance between attributes as radius Eps to ensure that the data clustering is correct in a single case. The formula is as follows:

$$\text{Eps} = \sqrt{\sum_{i=1}^M y_i^2}, \quad (4)$$

where y_i is the average distance difference of attributes; M is the size of SW.

DBSCAN algorithm is used for clustering in the clustering process. In view of the inconsistency between attributes, weights are assigned to each attribute to reduce the

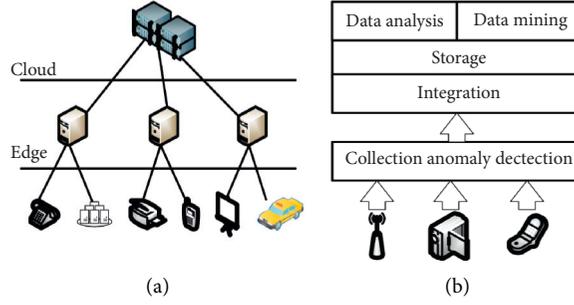


FIGURE 1: System architecture diagram based on edge computing.

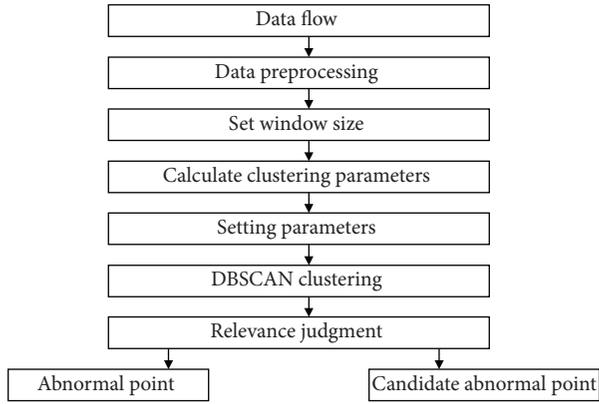


FIGURE 2: Algorithm flow diagram.

impact on the clustering effect. The weight W_{XY} is calculated by the correlation coefficient, and the formula is as follows:

$$r(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}$$

$$W_{XY} = \begin{cases} 0, & 0 \leq r(X, Y) \leq 0.1, \\ \frac{1}{r(X, Y)}, & 0.1 < r(X, Y) \leq 1, \end{cases} \quad (5)$$

where $\text{Cov}(X, Y)$ is the covariance of attribute X and attribute Y ; $\text{Var}(X)$ is the variance of attribute X ; $\text{Var}(Y)$ is the variance of attribute Y .

The anomaly detection process will analyze clustering results. In the clustering process, the object marked as an abnormal point for the first time is recorded as a candidate abnormal point, and the abnormal score is set plus 1 (the initial value is 0). The candidate abnormal points enter the next cycle, continue clustering, and update the abnormal score. If the abnormal score S is equal to the number of clustering C (the number of clustering is the inverse of SW overlap rate), it is marked as an abnormal point; otherwise, it is a normal point.

According to Definition 1, the multisource sensor data set is represented as a time series $\text{TD}_M = \{D_1, D_2, \dots, D_M\}$. When multisource sensor data enter SW, select the area data observation value of $\text{Length}_{\text{SW}}$ whose length is equal to the length of SW to form a new multisource sensor data set

$\text{TD}'_M = \{D'_1, D'_2, \dots, D'_M\}$, where $D'_i (1 \leq i \leq M)$ contains the area data observation value of length $\text{Length}_{\text{SW}}$, namely $D'_i = \{d_{i1}, d_{i2}, \dots, d_{it}\}$, $1 \leq t \leq \text{Length}_{\text{SW}}$. According to Definition 2, part of the time series set with known correlation in TD'_M is represented as TD'_{sub} , which can be combined or transformed into a new time series TD'_K with linear correlation. Then, enter the parameters in TD'_K into the set Ω_E . Since there is usually a certain correlation between data collected by sensors, the correlation between sensor data can be used to determine whether the sensor data is abnormal in a certain time range.

3.4. Algorithm Performance Analysis. The time complexity of the DBSCAN algorithm is the time required to find a point in radius Eps neighborhood, and its time complexity is $O(n^2)$ in the worst case. Improved DBSCAN algorithm uses a SW to divide data, and its time frequency can be expressed as follows:

$$g(n) = \left(\frac{xn}{w} + 1\right)w^2, \quad (6)$$

where n is the algorithm input scale, w is SW size, that is, w^2 is a constant. x is a constant; that is, the sliding step is w/x . Thus, the time complexity $T_2(n)$ of the improved DBSCAN algorithm is as follows:

$$T_2(n) = O(g(n)) = O\left(\frac{xn}{w} + 1\right) = O(n). \quad (7)$$

The time complexity of the proposed algorithm is the sum of the time complexity of KNN and the improved DBSCAN clustering algorithm. It can be expressed as follows:

$$T(n) = O(\max(f(n), g(n))) = O(n). \quad (8)$$

It can be seen that the time complexity of the anomaly detection algorithm increases linearly. As the amount of processed data samples increases, the time efficiency is higher than the basic DBSCAN algorithm.

4. Case Study and Discussion

4.1. Experimental Environment. In order to verify the effectiveness and feasibility of this method, some case studies are carried out. The hardware environment uses Think station P910 with Intel Xeon E5V4 CPU, 32 GB memory and

1 TB SSD. The software environment is ZooKeeper V3.4.8, jdkv1.8.0 and Storm V1.0.0. All algorithms are running on CentOS 6.5.

The experimental data comes from the urban heating system in Baohe District, Hefei, with 18031 users and 400 buildings. In the data collection process, the transmitters of each building will collect the data recording heating information of each room and send them to the corresponding receivers.

4.2. Execution Efficiency of Cloud Computing and Edge Computing Platforms. The algorithm is implemented on a centralized cloud computing platform [19] and a distributed edge computing platform. Table 1 shows the average processing time for each step. As can be seen from Table 1, due to the large amount of data in cloud computing, the corresponding bandwidth pressure is also greater. Therefore, the transmission delay of cloud computing is longer than that of edge computing. Moreover, due to the relatively strong computing power of cloud computing platforms, the time required for each step of using cloud computing has increased by 211.3 ms and 101.1 ms, respectively. Therefore, the computational efficiency of edge computing is higher. This is because the paper proposes an anomaly detection model for sensory data based on edge computing and uses the big data processing idea of edge computing to process corresponding data as much as possible on computing resources close to the data source. It improves the overall efficiency of data processing while reducing the pressure on network transmission bandwidth.

4.3. Detection Results Analysis. In this subsection, the performance of the proposed algorithm is evaluated from two aspects by experimental results.

- (1) Using the anomaly detection algorithm proposed in this paper, local detection of sensor data including accumulated heat, thermal power, accumulated temperature, flow, and temperature difference is performed. The number of selected sensor data is 500000. Using the proposed algorithm to find abnormal sensor data, the test results are shown in Table 2. It can be seen from Table 2 that there are 2650 abnormal P, W or (P, W) records, and 1997 abnormal V, s or (V, s) records. AD_{sum} represents the total number of abnormal sensor data, and AD_{cor} represents the abnormal sensor data successfully detected. Thus, the detection accuracy (expressed as AD_{pre}) can be calculated as follows:

$$AD_{\text{pre}} = \frac{AD_{\text{cor}}}{AD_{\text{sum}}} \times 100\%. \quad (9)$$

It can be seen from Table 2 that the basic DBSCAN algorithm can find 2650 abnormal records ($P/W/(P, W)$) of 2,320 and 1997 abnormal records

($S/V/(S, V)$) of 1802. The detection accuracy rates were 87.5% and 90.2%, respectively. The improved DBSCAN algorithm proposed in this paper can detect 2530 cases out of 2650 abnormal records ($P/W/(P, W)$) and 1909 cases out of 1997 abnormal records ($S/V/(S, V)$). The detection accuracy rates were 95.5% and 96.0%, respectively. It can be seen that the improved DBSCAN algorithm proposed in this paper can effectively improve the detection results of abnormal data.

- (2) In order to verify the method superiority in this paper, the methods in reference [10], reference [11] reference [12] are selected as benchmarks. 3,433,756 sensor records are selected from the data of the past two years, the proposed algorithm and three benchmark methods are used to detect anomalies. Figure 3 shows the average detection accuracy.

It can be seen from Figure 3 that the detection accuracy of the proposed method is increased by 1.91%, 2.04%, and 2.7%, respectively, reaching 96.4% compared with the methods in reference [10], reference [11], and reference [12]. This is because benchmarking methods do not effectively use the correlation between multisource time series to accurately assess the change trend.

This paper intercepts the detection results of sensor data in data set for statistics. Among them, there are 15 point anomalies, 49 cluster anomalies, and 13 correlation anomalies. The detection rate is 97.8% and the false alarm rate is 2.2%. In order to describe the situation of abnormal points being marked more clearly, this paper intercepts the first 180 sample points of air temperature data for drawing; the detection results are shown in Figure 4. The abnormality of air temperature data occurs within a short period of time (each sample is collected at an interval of 10 minutes), and the reasons for abnormality are all errors.

In order to further verify the time efficiency of the proposed algorithm, the result is shown in Figure 5 by comparing and verifying the datasets.

It can be seen from the figure that the running time of the method in reference [12] increases the fastest, and the running time of the proposed method and the method in reference [10] increases slowly. When the dataset reaches 8440 hours, that is, when the number of data points reaches 337760, the running time of the proposed method and the method in reference [12] is far less than the running time of methods in reference [10] and reference [11]. Combined with the previous analysis of time complexity, it can be seen that the improved DBSCAN algorithm takes advantage of the spatial correlation characteristics of multidimensional data, fully considers the data relevance, and effectively mines the data potential relationship. Therefore, when the sample size increases to a certain extent, the time efficiency of the proposed method is lower than that of several comparison algorithms. In summary, the proposed method can be used for anomaly detection of multisensor data streams and is feasible.

TABLE 1: Time consumption of each step for the proposed algorithm.

Steps	Time consuming in edge computing/ms	Time consuming in cloud computing/ms
Receive	224.5	435.8
Save	120.2	221.3

TABLE 2: Anomaly detection results of sensor data.

Abnormal parameters	DBSCAN	Improved DBSCAN	Total abnormal parameters
W	496	536	556
P	895	965	1050
WP	929	1029	1044
$W/P/WP$	2320	2530	2650
AD_{pre}	0.875	0.955	/
V	188	195	202
S	422	452	476
VS	1192	1262	1319
$V/S/VS$	1802	1909	1997
AD_{pre}	0.902	0.960	/

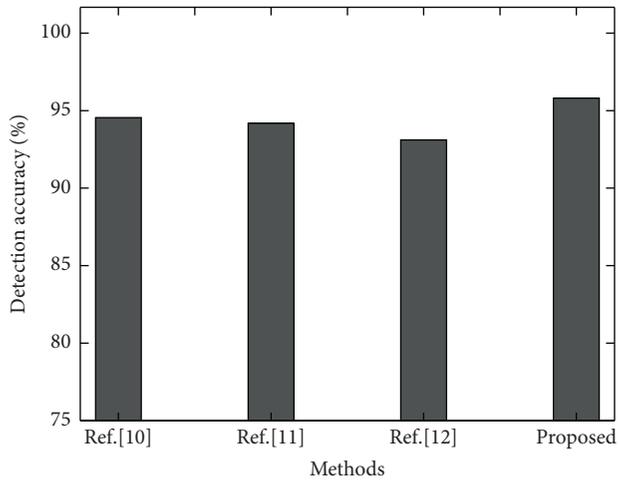
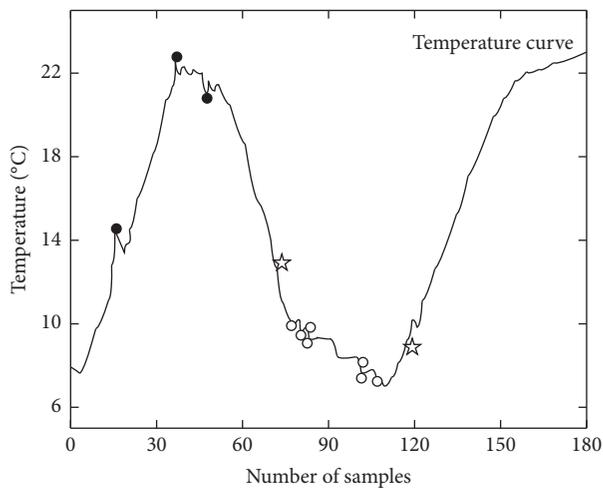


FIGURE 3: Anomaly detection accuracy of different methods.



- Abnormal point
- Abnormal gathering
- ☆ Abnormal association

FIGURE 4: Data anomaly detection results.

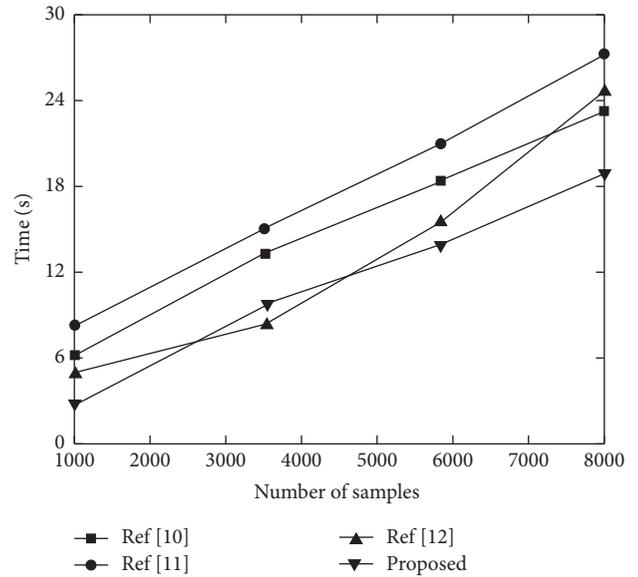


FIGURE 5: Time efficiency comparison of anomaly detection algorithms.

5. Conclusion

This paper proposes a real-time detection method for abnormal data of IoT sensors based on edge computing, which combines the ST correlation of sensor data streams and ideas of nearest neighbor algorithm and clustering algorithm. The method optimizes parameters according to the characteristics of environmental data and overcomes fixed nearest neighbor distance threshold, global clustering parameters, and slow convergence speed problems, which improves anomaly detection efficiency. For a relevant multisensor data stream, its effect can meet the use of the current environment. Simulation results show that the proposed method has higher computational efficiency and detection accuracy than traditional methods and has certain feasibility. However, limited to the author's level, the algorithm in this paper still

has room for improvement. In the future, we will focus on optimizing edge computing model and extending the detection algorithm to other real-time data application scenarios.

Data Availability

The data included in this paper are available without any restriction.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] S. Pal, B. Sikdar, and J. H. Chow, "Classification and detection of PMU data manipulation attacks using transmission line parameters," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5057–5066, 2018.
- [2] Y. Yongping, L. Xiaoen, Y. Zhiping et al., "The application of cyber physical system for thermal power plants: data-driven modeling," *Energies*, vol. 11, no. 4, pp. 690–706, 2018.
- [3] G. Cavararo and R. Arghandeh, "Power distribution network topology detection with time-series signature verification method," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 3500–3509, 2018.
- [4] Y. Wei, X. Zhang, Y. Shi et al., "A review of data-driven approaches for prediction and classification of building energy consumption," *Renewable and Sustainable Energy Reviews*, vol. 82, no. 1, pp. 1027–1047, 2018.
- [5] E. Pacharawongsakda and T. Theeramunkong, "Multi-label classification using dependent and independent dual space reduction," *Computer Journal*, vol. 56, no. 9, pp. 1113–1135, 2018.
- [6] Z. Gong, H. Chen, Y. Bo et al., "Multi-objective learning in the model space for time series classification," *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 918–932, 2018.
- [7] J. Georg, Z. Sebastian, and C. António, "Generic sensor failure modeling for cooperative systems," *Sensors*, vol. 18, no. 3, pp. 925–955, 2018.
- [8] V. Pashazadeh, F. R. Salmasi, and B. N. Araabi, "Data driven sensor and actuator fault detection and isolation in wind turbine using classifier fusion," *Renewable Energy*, vol. 116, no. B, pp. 99–106, 2017.
- [9] L. Cai, N. F. Thornhill, S. Kuenzel, and B. C. Pal, "Real-time detection of power system disturbances based on k -nearest neighbor analysis," *IEEE Access*, vol. 5, no. 3, pp. 5631–5639, 2017.
- [10] K. J. Diao, Y. Q. Liang, and J. C. Fan, "An improved DBSCAN algorithm using local parameters," in *Proceedings of the 2018 First CCF International Conference on Artificial Intelligence*, pp. 3–12, Springer, Jinan, China, August 2018.
- [11] A. Sarvani, B. Venugopal, and N. Devarakonda, "Anomaly detection using K-Means approach and outliers detection technique," in *Soft Computing: Theories and Applications*, pp. 375–385, Springer, Berlin, Germany, 2018.
- [12] N. Berjab, H. H. Le, C. Yu, S. Kuo, and H. Yokota, "Abnormal-node detection based on spatio-temporal and multivariate-attribute correlation in wireless sensor networks," in *Proceedings of the 2018 IEEE 16th Intelligence Conference on Dependable, Autonomic and Secure Computing*, pp. 568–575, Athens, Greece, August 2018.
- [13] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [14] A. Aral and T. Ovatman, "A decentralized replica placement algorithm for edge computing," *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 516–529, 2018.
- [15] S. Garg, A. Singh, K. Kaur et al., "Edge computing-based security framework for big data analytics in VANETs," *IEEE Network*, vol. 33, no. 2, pp. 72–81, 2019.
- [16] Y. Cao, H. Song, O. Kaiwartya et al., "Mobile edge computing for big-data-enabled electric vehicle charging," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 150–156, 2018.
- [17] M. Gusev and S. Dustdar, "Going back to the roots—the evolution of edge computing, an IoT perspective," *IEEE Internet Computing*, vol. 22, no. 2, pp. 5–15, 2018.
- [18] A. A. A. Punitha and G. Indumathi, "Centralized cloud information accountability with bat key generation algorithm (CCIA-BKGA) framework in cloud computing environment," *Cluster Computing*, vol. 9, no. 2, pp. 25–34, 2018.
- [19] S. Sagnika, S. Bilgaiyan, and B. Mishra, "Workflow scheduling in cloud computing environment using bat algorithm," *International Journal of Computer Applications*, vol. 89, no. 2, pp. 11–18, 2018.