Hindawi
Scientific Programming
Volume 2021, Article ID 6699313, 9 pages
https://doi.org/10.1155/2021/6699313

Research Article

Sensors Anomaly Detection of Industrial Internet of Things Based on Isolated Forest Algorithm and Data Compression

Desheng Liu, Hang Zhen, Dequan Kong, Xiaowei Chen, Lei Zhang, Mingrun Yuan, and Hui Wang

College of Information and Electronic Technology, Jiamusi University, Jiamusi 154007, China

Correspondence should be addressed to Dequan Kong; 893907285@qq.com and Hui Wang; 3120205463@bit.edu.cn

Received 27 October 2020; Revised 20 December 2020; Accepted 15 January 2021; Published 31 January 2021

Academic Editor: Liang Zhao

Aiming at solving network delay caused by large chunks of data in industrial Internet of Things, a data compression algorithm based on edge computing is creatively put forward in this paper. The data collected by sensors need to be handled in advance and are then processed by different single packet quantity K and error threshold e for multiple groups of comparative experiments, which greatly reduces the amount of data transmission under the premise of ensuring the instantaneity and effectiveness of data. On the basis of compression processing, an outlier detection algorithm based on isolated forest is proposed, which can accurately identify the anomaly caused by gradual change and sudden change and control and adjust the action of equipment, in order to meet the control requirement. As is shown by experimental simulation, the isolated forest algorithm based on partition outperforms box graph and K-means clustering algorithm based on distance in anomaly detection, which verifies the feasibility and advantages of the former in data compression and detection accuracy.

1. Introduction

With the rapid development and integration of the Internet of Things (IoT) and cloud computing technology, we have gradually entered the era of "Internet of Things, comprehensive perception" [1]. At the same time, a large number of sensor devices are widely used in various fields including biomedicine, petrochemical, public transportation, environmental protection, electric power, and industrial manufacturing. In spite of the excitement, IoT sensor-based technology still faces great challenges and uncertainties in its authenticity, timeliness, reliability, and security. With the extensive use of sensor devices, people's lifestyle changes a lot; meanwhile, massive time series data are generated during the process of application. According to the Internet Data Center (IDC) [2], by 2020, the global .data are expected to exceed 40 zb. Boeing 787 generates more than 5 GB of data per second, and the bandwidth between the aircraft and the satellite is not enough to support real-time transmission [3]. In order to capture road information in real time, sensors and cameras mounted on unmanned vehicles will generate about 1 GB of data per second. According to IHS, by 2035, there will be 54 million driverless vehicles in the world [4].

Usually, sensors collect data at a certain frequency and send the data to the cloud. The cloud then receives the observed data in strict-time sequence. These data known as "time series data" accurately record the real-time changes of certain parameters at some point, such as speed, power, and temperature. They can reflect the regulation of data changes under certain parameters, which is the premise of subsequent data analysis and mining. In practical scenarios, there are always some abnormal data that deviate from the normal perception in the process of data acquisition and transmission; thus, it is very difficult to obtain high-quality data through sensors. Furthermore, the occurrence of faults is always unpredictable. Nowadays, most of the anomaly detection algorithms are based on statistics, clustering, similarity measurement, constraint rules, and neural network

[5–9]. Statistical methods usually know the distribution of the sequence. By maintaining the sliding window and calculating the statistical characteristic indexes, abnormal parts can be detected accurately. This method is suitable for detecting discrete and abrupt value anomalies in the sequence, but it is difficult to effectively identify the continuous abnormal sequence interval. The clustering method quantifies the distance between outlier and normal cluster to judge outliers. Computational complexities of different clustering models vary tremendously, and the detection results depend on the quality of clustering. The method based on similarity measure can judge whether there is abnormal data by calculating the similarity between the standardized sequences. However, this method can take a long time. In the rule-based method, researchers have proposed sequence dependence and speed constraint, which can effectively use the characteristics in time series to repair highly abnormal data. However, this method can hardly meet the needs of sequence anomaly detection with variable patterns [10]. Yu et al. [11] proposed the framework of IoT monitoring system based on edge computing, and an anomaly detection approach using self-encoding neural network. According to the particularity of time series data and the difference of data composition, literature [12] proposes that, in the field of time series data, most anomaly detection methods are based on pattern recognition and clustering. In [13], a new anomaly detection algorithm for time series data is put forward, constructing a distributed recursive computing strategy and k-nearest neighbor fast selection strategy. Qi et al. [14] proposed a real-time anomaly detection algorithm for sensing data based on edge computing. By analyzing the continuity and correlation between sensing data in the form of time series, the algorithm establishes a distributed anomaly detection model of sensing data based on edge computing so as to effectively detect anomalies in real-time sensing data. At present, most of the existing time series anomaly detection methods focus on the abnormal recognition of single dimension periodic or simple pattern time series. A lot of misjudgments and omissions may occur in the process, which leads to the performance degradation of anomaly detection methods. Although various anomaly detection methods have been proposed in the literature, it is still difficult to accurately detect the abnormal data and patterns for the one-dimensional time series with variable patterns.

In addition, nowadays the IoT data is processed in the cloud, and cloud computing can provide an efficient computing platform for big data processing. However, with the growth rate of network bandwidth lagging well behind that of data, data transmission delay and energy consumption of cloud data center have increased significantly, which lead to the bottleneck of cloud computing. As a new computing mode, the core of edge computing is to migrate the decomposed computing tasks to the edge nodes for processing, so as to realize the preprocessing of data before entering the cloud server, and to reduce the computational load of cloud computing data center. It has been applied in many fields, such as online shopping, smart home, smart city, intelligent transportation, security monitoring, etc. [1, 15]. In order to provide a better computing platform for the Internet of Things, a cloud computing center with strong computing power and mass storage, this paper proposes an edge collaborative cloud architecture with the help of edge devices processing massive data and private data in edge computing. On this basis, an algorithm of data compression and anomaly detection based on edge computing comes into being. The data collected by sensors are preprocessed to reduce the amount of data transmission, so as to greatly reduce the cloud computing load. Analyzing from the perspective of time series data, anomalies in the sensor data can be effectively identified, and the normal data fluctuation in the sensor data is entirely retained.

The main content of this paper is as follows: Section 2 outlines the application of edge computing in the IoT and the advanced algorithms in sensor outlier detection. Sections 3 and 4 describe the basic principle and structure of isolated forest algorithm. In Section 5, the evaluation indicators of compression algorithm and anomaly detection algorithm are discussed, and the performance of the algorithm is evaluated experimentally using actual data. In Section 6, the whole research idea is summarized.

2. Related Work

2.1. Internet of Things. The Internet of Things, also known as the "Internet connecting goods," is an outstanding practical result of information network development during the third revolution of science and technology. IoT has now penetrated into various fields, including transportation, public safety, environmental protection, electric power, smart home, and medical health, and has received widespread attention from all walks of life. The Internet of Things refers to the connection of any object with the network through the information sensing equipment according to the agreed protocol. The objects exchange information through the media, so as to realize intelligent identification, positioning, tracking, supervision, and other functions. The Internet of Things, as the name suggests, is developed on the basis of the Internet. Put simply, it is an extension of the Internet. The information exchange and sharing of client extend the communication between things.. The Internet of Things is formed when everything is connected at any time, in any place, and between anyone.

Compared with the Internet, the Internet of Things covers a wider range. It does not necessarily require direct participation of people. Problems of objects are analyzed and managed by artificial intelligence. It contains a large number of sensor applications. Sensor is the source of massive data in the Internet of Things, which is more abundant in data types and processing diversification. It mainly uses wireless technology to connect. It can carry out real-time information interaction and data transmission, as well as information processing. It can integrate the storage, processing and analysis capabilities of things at one end of things, real-

time data processing, and feedback to improve user response efficiency and user experience [16].

With the development of information and communication technology, many items and devices can be connected to the network, for example, articles carrying radio frequency identification code, and most devices in industrial control, environmental control, and traffic control. Therefore, the IoT technology can make things more intelligent. The application of Internet of Things has covered the whole Internet field. The IoT architecture can be divided into perception layer, network layer, and application layer. The perception layer is the source of data and the source of identifying objects and collecting information in the Internet on Things. Mainly composed of a large number of sensors, RFID tags, cameras, and other sensing sensors, it is the basic layer supporting the whole IoT system. The network layer is the center of the Internet of Things, which is responsible for data transmission. It connects the application layer and the perception layer and realizes the relationship between things by wireless communication through the exchange equipment and transmission equipment. In this way, the user terminals distributed in different locations are connected to form a complete information transmission path. The application layer is a direct user-oriented interface, through which users interact with objects [17].

With the rapid development of IoT technology, a series of national strategies, including Made in China 2025, Advanced Manufacturing Partner program of the United States, and German Industry 4.0, are put forward and implemented. The Industrial Internet of Things (IIoT) emerges as the times require and has become an important driver of the intelligent transformation of global industrial system (originated from China Institute of Electronic Technology Standardization). IIoT, a cutting-edge industry of huge commercial value, is widely used in design, production, management, and service [16]. IIoT realizes flexible allocation of raw materials, execution of manufacturing process on demand, reasonable optimization of production process through network interconnection and rapid adaptation to the manufacturing environment, and data exchange and system interoperability of industrial resources to achieve efficient utilization of resources, in order to build a new service-driven industrial ecosystem [18, 19]. The Internet of Things (IoT) is equivalent to information about physical objects (sensors, machines, cars, buildings, and other objects), which makes possible the interaction and cooperation between these objects to achieve common goals. It helps realize remote monitoring and intelligent maintenance application scenarios of industrial equipment, and remote monitoring, preventive maintenance, and performance optimization analysis of equipment [20]. The so-called IIoT is an advanced production mode that uses cloud platform to upgrade traditional industry to intelligent industry.

2.2. Edge Calculation. As a key technology to realize the Internet of Things, edge computing is widely used in many fields, such as smart city, intelligent manufacturing, intelligent transportation, smart home, privacy protection [21], disaster relief [22–25], etc. In the aspect of smart city, edge computing can meet three requirements of large data volume, low latency, and real-time location identification in the construction of smart city. It can efficiently process the massive data in various fields including public safety, health data, public facilities, and transportation information. It can reduce the time for data transmission and process the private data of users and relevant institutions more safely. In the aspect of intelligent manufacturing, edge computing can effectively realize the interaction and cooperation of information in each part of the intelligent manufacturing system and ensure the real-time data processing in the intelligent process. It can upload the processing results to the cloud for compensation calculation and then download them to the controller for operation, so as to reduce the communication cost and improve the processing efficiency. In the aspect of intelligent transportation, the system analyzes the data collected by cameras and sensors in real time through edge calculation and makes corresponding decisions, which can solve bandwidth waste and delay, improve security of intelligent transportation, extend the applicability of it, and provide a better user experience. In the aspect of smart home, the edge computing system runs on the edge gateway inside the home, integrating smart home devices into the system. And the data generated by the devices can be processed and desensitized locally, which can effectively reduce the data transmission delay and better protect the privacy of users. In the aspect of disaster rescue, the key of intelligent fire protection is to process, analyze, and predict the data obtained from multiple data sources, and effectively transmit the results to rescuers, which require high computing power and timely response. Through edge computing, the data can be transmitted to the base station through the edge equipment and then to the cloud without infrastructure. In transmission, the edge computing and storage resources will be used nearby to realize the partial processing, analysis, and prediction of the data, reduce the number of data transmissions, and shorten the bandwidth and response time.

Cloud computing and edge computing are key technologies to realize the Internet of Things. As a computing model, cloud computing accesses computing resources, network resources, and storage resources of the data center through the network and provides scalable distributed computing capability for applications [26]. With the characteristics of large-scale servers, high reliability, strong extensibility, and virtualization, IT cloud computing is used by more and more enterprises and organizations to deploy their applications. But in cloud computing mode, computing tasks are handled by the cloud center. The service provider provides the data to be uploaded to the cloud center, and the client of the terminal sends the request to the cloud center. The cloud center responds to the relevant request and sends the relevant data to the terminal customer. The terminal customer always plays the role of consumer. Edge computing is a new computing mode to perform computing at the edge of the network, which places the data that should be processed in the cloud center near the data source. The comparison between edge computing and cloud computing is shown in Table 1.

Table 1: Comparison of edge computing and cloud computing.

Content	Edge computing	Cloud computing
Target application	Internet of Things or mobile application	General Internet
Service node location	Edge network	Data center
Communication network	WLAN 4 g/5 g	Wan
Number of devices available for service	Billions	Millions
Types of services provided	Local information	n global information

As can be seen from Table 1, compared with cloud computing, edge computing has the following obvious advantages: first, it can improve the security of data center; third, it can enhance the security of data. But edge computing cannot replace cloud computing. It is the extension of cloud computing, providing a better computing platform for the Internet of Things. Edge computing model requires the strong computing ability and mass storage support of cloud computing center. Cloud computing also needs the processing of massive data and private data by edge devices in edge computing to meet the real-time requirement and satisfy the needs of privacy protection. Therefore, the device edge cloud architecture model can provide a better configuration scheme.

3. Data Compression Preprocessing Based on Edge Computing

Aiming at the problem of cloud computing transmission and feedback delay caused by massive IoT data, an effective method is designed to better process a large amount of sensor time series data. Generally, increasing data redundancy can improve the stability of the system. In a sense, low data redundancy and high data reliability are contradictory, which means it is very difficult to find the optimal solution of minimum data redundancy and maximum data reliability. The shorter the processing time is, the better the compression processing is carried out on the premise that original data characteristics of the sensor and the true reflection of the data are not changed.

The method used in this paper needs to set the number k and error threshold e of each group of data packets in advance. When the time sequence data t is uploaded to the edge end, all the first k temperature data are uploaded. When the average value of the time sequence data $T[i+k]$ and its first k time series data is less than the error threshold e, the output will not be carried out, so as to cycle when $T[i+2k-1]$ and $T\ [i+2k-1]$ still meet the above conditions. We take the average value of $T\ [i+2k-1]$ and the first $k-1$ data as the uploaded data and store them in out2.txt, and $I+k$ in out1.txt. If the time series data $T[i+k]$ appears and the average value of the first k time series data in the group is no less than the error threshold e, then $T[i+k]$ is directly uploaded and stored in out2.txt, and $I+k$ is stored in out1.txt to reduce the amount of data transmission and subsequent data processing. Among them, $T[i]$ is the ith time series data collected, and out1.txt and out2.txt are edge storage files. The implementation of sensing data compression algorithm is shown in Algorithm 1.

4. Anomaly Detection Based on Isolated Forest Algorithm

Isolation forest algorithm is an unsupervised anomaly detection method based on random binary tree and suitable for continuous data [28]. In isolated forests, anomalies are defined as "outliers that are easily isolated," that is, points with sparse distribution and far away from high-density population. In the feature space, the sparsely distributed region indicates that the probability of events occurring in the region is very low, so it is judged that the data distributed in the sparse area is abnormal. It is suitable for anomaly detection of time series data.

The forest isolation algorithm is described in detail:

(i) Define 1 so that t is a binary tree and N is the node of T. if N is a leaf node, it is called an external node; if N is a node with two children, it is called an internal node.

Definition 2 in an iTree; the data of the edge from the root node to the outer node is called the path length, which is denoted as $H\ (s)$.

The construction process of a single iTree is as follows: select a point randomly from the data set $S = \{S1, S2, S3, \cdot\cdot, Sn\}$ to generate the cut point P randomly. The cutting point P is generated between the maximum value and the minimum value of the specified dimension in the current node data, and then each data is divided. The selection of the cutting point generates a hyperplane, which places the points smaller than P in the left branch of the current node and points greater than or equal to P in the right branch of the current node. The left and right branches are constructed recursively until only one data set or tree on the leaf node has grown to the set height. Traverse each iTree to find the final path length of S. Since the cutting process is completely random, we need to use the method of ensemble to make the result converge; that is, repeatedly start cutting from the beginning, and then calculate the average value of each segmentation result, namely, $H\ (s)$. The schematic diagram of data traversal iTree is shown in Figure 1.

```
Input: data.txt sensor data T number of packets processed in a single group K, error threshold E.
Output: out1.txt, out2.txt.
(1)  for i = 1 to N
(2)    read the data from "test.txt", and write them to "data.txt"
(3)  if e of the "test.txt"
(4)    break
(5)  end if
(6)  for i = 1 to N
(7)    read the data from "data.txt" to T[i + 1]
(8)  aver = sum(T)/i + 1;
(9)  end
(10) if (aver < 0)
(11)   for i = 1 to k
(12)     aver < aver + T[i]
(13)     aver < aver/k
(14)   end
(15) else
(16)   for i = 2 to n
(17)     temp < aver
(18)       for j = 0 to k − 1
(19)         if i + j ≥= n
(20)           temp < −1
(21)           aver < aver + T[i + j]
(22)         end if
(23)       end
(24)     end if
(25)   end
(26) end if
(27) aver < aver/k
(28) if |aver-temp| >= e
(29)   put i + j −1 to "out1.txt"
(30)   put T[i + j −1] to "out2.txt"
(31) end if
(32) return "out1.txt","out2.txt"
```

Algorithm 1: Sensor data compression algorithm.

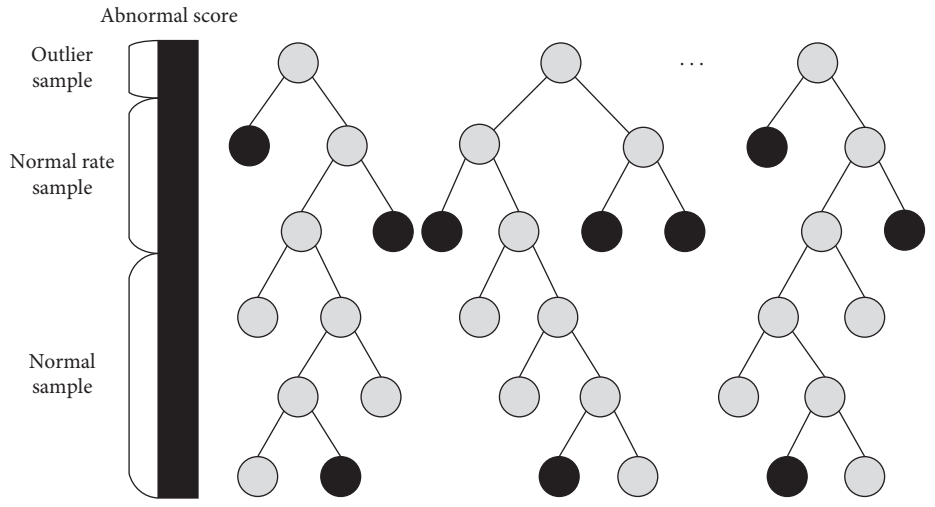

Figure 1: Orest anomaly detection process.

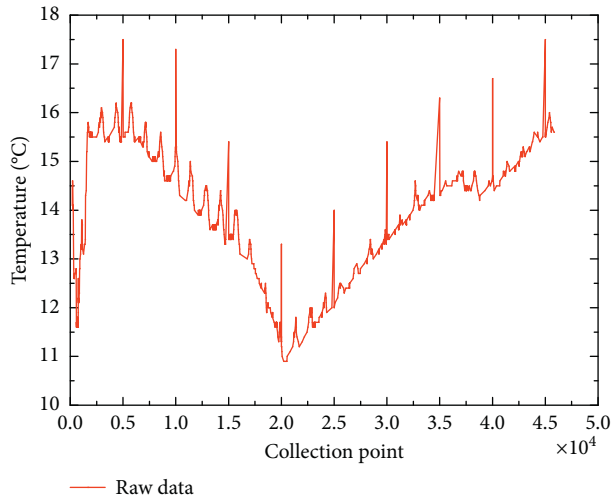

Figure 2: Scatter plot of time series variation trend of temperature data set.

$$H(k) = \ln(k) + \xi,$$

$$c(A) = \begin{cases} 2H(A-1) - \dfrac{2(A-1)}{n}, & A > 2, \\ 1, & A = 2, \\ 0, & A < 2, \end{cases} \tag{1}$$

$$E(h(S)) \longrightarrow 0, s \longrightarrow 1; E(h(S)) \longrightarrow A-1, s \longrightarrow 0; E(h(S)) \longrightarrow c(A), s \longrightarrow 0.5.$$

$H(s)$ is the node depth of S in iTree. $E[.]$ *is* the average of t iTrees. $c(A)$ is the average length of a point bisection search tree. $H(k) = \ln(k) + \xi$, ξ is Euler's constant. The closer $S(S)$ is to 1, the more likely it is to be abnormal data; and the closer it is to 0, the more likely it is to be a normal point. When the $S(S)$ of most data is 0.5, there is no abnormal value in the data.

Isolated forest algorithm is different from clustering, box graph, and other algorithms; it does not need to calculate the distance, density, and other indicators; it can greatly improve the calculation speed and reduce the system overhead. In the process of training, each iTree is randomly selected and generated independently. It accelerates the operation of the deployment of large-scale distributed systems. Based on the ensemble method, the more iTrees, the more stable the algorithm.

5. Experimental Simulation

The temperature data used in this paper is collected from the environmental data set uploaded from the experimental cloud platform of the Internet of Things. The time is intercepted from 8 : 00 on May 1, 2019, to 7 : 15, May 17, 2019. The data upload interval is 30 s, with a total of 45989 temperature sensing data, and the data accuracy is 0.1°C. Figure 2 shows a scatter diagram of time series variation trend of temperature data set, including 10 times of anomalies caused by gradual change or sudden change.

Hardware environment: all experiments are carried out with Windows 7 operating system, CPU is Intel Core i5 4200u, the graphics card is AMD Radeon HD 8670 m, memory is 4 GB, and python platform is used for simulation.

The isolated forest algorithm is used to detect the original temperature data set and four groups of compressed data sets to evaluate the performance of outlier detection. The parameters are as follows: the number of iTree $t = 100$; the number of test samples $a = 256$; the path length $H(s) = 15$. As shown in Figure 3, the test results of iForest algorithm in the original data set show that there are 10 abnormal data detected, all of which are detected without misjudgment. Figures 4–7, respectively, show the anomaly detection results of four groups of data based on iForest algorithm. In the first group, 10 abnormal data were detected, but one normal data was misjudged as abnormal data, and one abnormal data was not detected; nine abnormal data were detected in the second group without misjudgment, and one abnormal data was not detected; the third group detected 10 abnormal data, but there were 2 misjudgments, and 2 abnormal data were not detected; 10 abnormal data were detected in the fourth group, without misjudgment.

In order to verify the comparison and analysis of anomaly detection accuracy of the three algorithms, and to

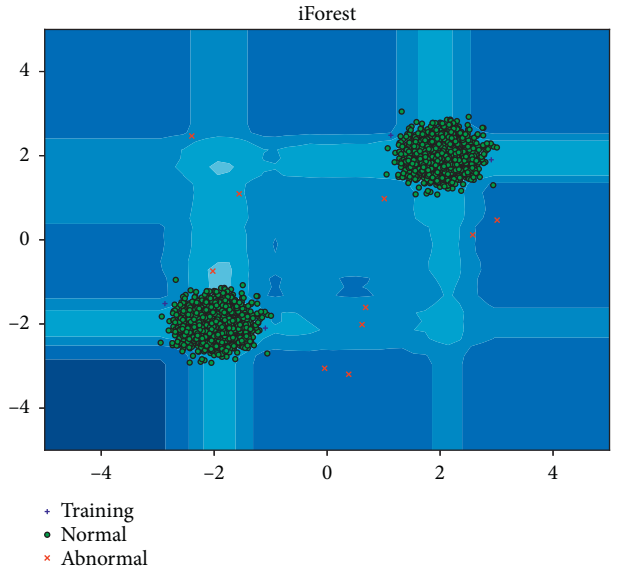

Figure 3: Anomaly detection results of iForest algorithm in the original data set.

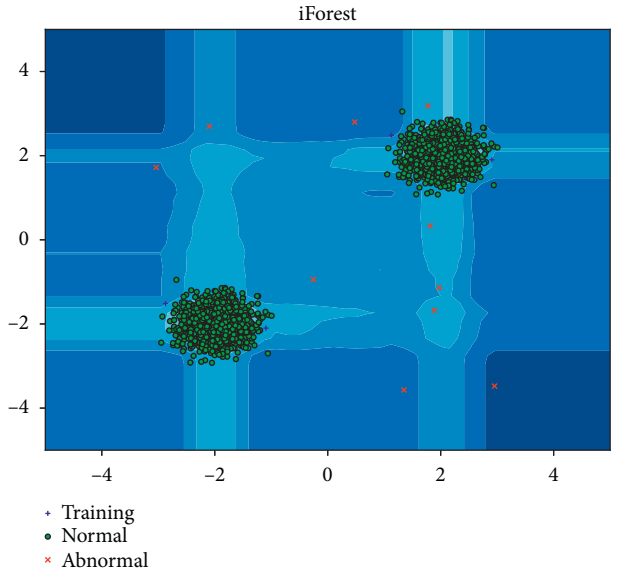

Figure 4: iForest algorithm anomaly detection results of the first group of compressed data.

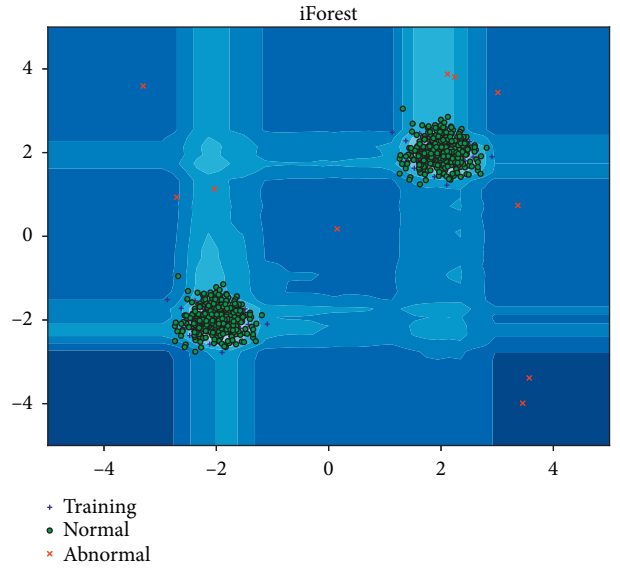

Figure 5: Anomaly detection results of iForest algorithm for the second group of compressed data.

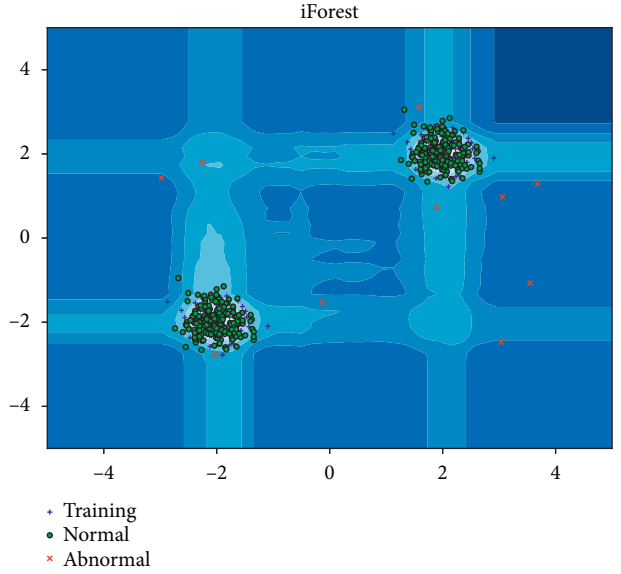

Figure 6: Anomaly detection results of iForest algorithm for the third group of compressed data.

assure the reliability and justness of results, the anomaly detection results of different algorithms for the original data and compressed data are listed in Table 2 (note: the original data is before the processing, and the data is after the processing).

In, Table 2 the accuracy is calculated by the ratio of the number of correctly classified samples to the total number of samples, the accuracy is the ratio of the correct prediction to the positive proportion of all the predicted samples, and the recall rate is the ratio of the correct prediction to the positive proportion of all the positive samples, which can be understood as the ratio between the number found and the total number to be found.

It is not difficult to find from Table 2 that the accuracy and recall rate of iForest algorithm are generally higher than those of the other two algorithms in the anomaly detection of the original data set and the compressed data set. In the comparison of different data sets of the same algorithm, due to the large amount of original data, the abnormal detection accuracy, accuracy, and recall rate of the original data are obviously higher than those of the compressed data, while the other compressed data does not deviate from changing the tracking of the original data. When the data is flat, the compressed data can replace the original data with fewer values; when the data becomes different, the original data can be replaced by the compressed data in normal time, and

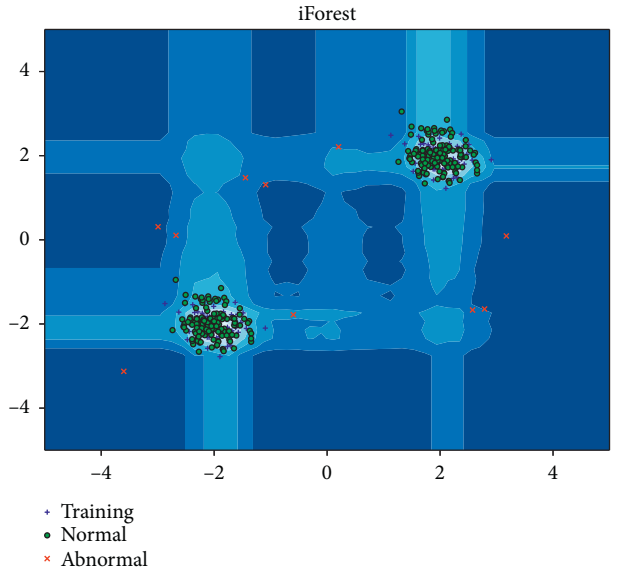

Figure 7: iForest algorithm anomaly detection results of the fourth group of compressed data.

Table 2: Comparison of detection results of three algorithms.

Algorithm category	Box diagram	K-means	iForest
Accuracy	99.98%, 96.95%	99.95%, 96.08%	100%, 99.52%
Precision	99.98%, 96.92%	99.99%, 99.3%	100%, 99.78%
Recall	100%, 99.25%	99.95%, 96.52%	100%, 99.72%
Execution time	15.33 s, 3.98 s	10.87 s, 2.85 s	14.21 s, 3.04 s

compressed data can keep the abnormal data for outlier detection, which can effectively prevent the abnormal data from being missed. In terms of algorithm execution time, the execution time of K-means clustering algorithm is always the shortest, but it is only 0.19 s shorter than iForest algorithm, which has no impact in practical application. Therefore, iForest algorithm based on partition outperforms box graph and K-means clustering algorithm based on distance in anomaly detection performance. From the aspect of the execution time of anomaly detection before and after compression processing, the box graph algorithm is shortened by 11.35 s, K-means clustering algorithm by 8.02 s, and iForest algorithm by 11.07 s. Data compression can significantly shorten the time of anomaly detection. Based on the time required for data compression, the time consumed in the whole data processing is still reduced to a certain extent. Therefore, the superiority of edge computing is finally verified.

6. Conclusions

In this paper, in order to solve the problem of cloud computing, transmission and feedback delay caused by the current massive IoT data, this paper proposes a cold chain monitoring management method based on edge computing through the research and analysis of the cold chain IoT monitoring system. The real-time sensing data is compressed to ensure that the original characteristics and true reflection of the sensing data remain unchanged, and the amount of data calculated in the cloud center can be reduced, as well as the transmission delay and response delay. Based on the data compression processing, the abnormal detection of the filtered data is carried out with high detection precision, which can timely detect anomalies and remind users of them.

In future work, we will take the lead in adjusting the compression conditions in data compression to achieve selective data compression. Secondly, in order to minimize the loss in the process of anomaly repair in the future, we try to add one or more prediction mechanisms in the follow-up work to reasonably optimize the anomaly detection method. Based on the consideration of the correlation between data files and the time interval of similar files being accessed, the cache replacement strategy will be improved.

Data Availability

According to the funding policy of this work, data cannot be shared or made publicly available during the funding contract.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was funded by Basic Research Project of Heilongjiang Province Department of Education (grant number: 2018-KYYWF-0942) and Heilongjiang Provincial Department of Education Science and Technology Innovation team construction project (2019-kyywf-1335)

References

[1] W. Shi, H. Sun, J. Cao, Q. Zhang, and W. Liu, "Edge computing: a new computing model in the Internet era," *Computer Research and Development*, vol. 54, no. 5, pp. 907–924, 2017.

[2] M. Zwolenski and L. Weatherill, "The digital universe rich data and the increasing value of the internet of things," *Australian Journal of Telecommunications and the Digital Economy*, vol. 2, no. 3, pp. 1–9, 2014.

[3] G. Pandian, M. Pecht, E. Zio, and M. Hodkiewicz, "Data-driven reliability analysis of Boeing 787 dreamliner," *Chinese Journal of Aeronautics*, vol. 33, no. 7, pp. 1969–1979, 2020.

[4] R. K. Runting, S. Phinn, Z. Xie, O. Venter, and J. E M. Watson, "Opportunities for big data in conservation and sustainability," *Nature Communications*, vol. 11, no. 1, 2020.

[5] Z. Li and Y. Zhang, "A new hyperspectral anomaly detection method based on higher order statistics and adaptive cosine estimator," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 4, pp. 661–665, 2020.

[6] H. Liu, Y. Wang, and W. Chen, "Anomaly detection for condition monitoring data using auxiliary feature vector and density-based clustering," *IET Generation, Transmission & Distribution*, vol. 14, no. 1, pp. 108–118, 2020.

[7] P. Li, Z. Chen, L. T. Yang, J. Gao, Q. Zhang, and M. J. Deen, "An improved stacked auto-encoder for network traffic flow classification," *IEEE Network*, vol. 32, no. 6, pp. 22–27, 2018.

[8] M. Gowri and B. Paramasivan, "Rule-based anomaly detection technique using roaming honeypots for wireless sensor networks," *ETRI Journal*, vol. 38, no. 6, pp. 1145–1152, 2016.

[9] T. Nakazawa and D. V. Kulkarni, "Anomaly detection and segmentation for wafer defect patterns using deep convolutional encoder–decoder neural network architectures in semiconductor manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 32, no. 2, pp. 250–256, 2019.

[10] S. Li, "The approach of dynamic data fusion based on multi-sensor temperature data," *Chinese Science and Technology*, vol. 31, no. 1, pp. 146–149, 2015.

[11] T. Yu, Y. Zhu, and X. Wang, "Anomaly detection using self coding neural network in Internet of things monitoring system based on edge computing," *Journal of Internet of Things*, vol. 2, no. 4, pp. 14–21, 2018.

[12] P. Li, Z. Chen, L. T. Yang, Q. Zhang, and M. J. Deen, "Deep convolutional computation model for feature learning on big data in internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 790–798, 2018.

[13] J. Gao, P. Li, Z. Chen, and J. Zhang, "A survey on deep learning for multimodal data fusion," *Neural Computation*, vol. 32, no. 5, pp. 829–864, 2020.

[14] Z. Qi, H. Y. Peng, J. Cun et al., "Application of edge computing: real time detection algorithm of sensor data anomaly," *Computer Research and Development*, vol. 55, no. 3, pp. 524–536, 2018.

[15] Y. Chen, "Application of edge computing in smart home," in *Proceedings of the 2019 National Symposium on Edge Computing*, pp. 73–79, China Communications Society, Beijing, China, 2019.

[16] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[17] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (IIoT): an analysis framework," *Computers in Industry*, vol. 101, pp. 1–12, 2018.

[18] Z. Huang, J. Chen, Y. Lin, P. You, and Y. Peng, "Minimizing data redundancy for high reliable cloud storage systems," *Computer Networks*, vol. 81, pp. 164–177, 2015.

[19] J. Zhang, X. Wu, Z. Yang et al., "Research and application of industrial data acquisition technology based on industrial Internet of things," *Telecommunication Science*, vol. 34, no. 10, pp. 130–135, 2018.

[20] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for Industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18–23, 2015.

[21] A. W. Colomboaw, S. Karnouskoss, and T. Bangemannt, "Towards the next generation of industrial cyber-physical systems," *Industrial Cloud-Based Cyber-Physical Systems*, pp. 1–22, Springer-Verlag, Beilin, Germeny, 2014.

[22] Y. Xu, H. Liu, and Q. A. Zeng, "Resource management and QoS control in multiple traffic wireless and mobile Internet systems," *Wireless Communications & Mobile Computing*, vol. 5, no. 8, pp. 971–982, 2010.

[23] R. Yadav, W. Zhang, O. Kaiwartya, P. R. Singh, I. A. Elgendy, and Y.-C. Tian, "Adaptive energy-aware algorithms for minimizing energy consumption and SLA violation in cloud computing," *IEEE Access*, vol. 6, pp. 55923–55936, 2018.

[24] R. Yadav, W. Zhang, K. Li, C. Liu, M. Shafiq, and N. K. Karn, "An adaptive heuristic for managing energy consumption and overloaded hosts in a cloud data center," *Wireless Networks*, vol. 26, no. 3, pp. 1905–1919, 2020.

[25] H. Wen and P.-H. Ho, "Physical layer technique to assist authentication based on PKI for vehicular communication networks," *KSII Transactions on Internet and Information Systems*, vol. 5, no. 2, pp. 440–456, 2011.

[26] F. Huang, G. Zhou, H. Ding et al., "Electrical energy anomaly data detection based on isolated forest algorithm," *Journal of East China Normal University (Natural Science Edition)*, vol. 5, pp. 123–132, 2019.