

Received May 13, 2021, accepted July 15, 2021, date of publication July 20, 2021, date of current version July 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3098849

A Balance Regulation Algorithm for Reliable Data in Hierarchical Private Cloud Architecture

CHAO-HSIEN HSIEH AND ZIYI WANG 

School of Cyber Science and Engineering, Qufu Normal University, Shandong, Qufu 273165, China

Corresponding author: Chao-Hsien Hsieh (george_hsieh@qq.com)

This work was supported by the Shandong Provincial Natural Science Foundation of China under Grant ZR2020MF048.

Chao-Hsien Hsieh and Ziyi Wang have contributed equally and are considered to be co-first authors.


ABSTRACT At present, enterprises need data for data analysis. They mostly use a kind of point-to-point data transmission form, which does not have regulation mechanism in the process. However, it has the problem of low data reliability, including two parts as follows: 1) Faulty sensor affects the collected data amount of the terminal server; 2) offensive data invade the data in transmission. In view of this, we propose the hierarchical private cloud architecture, including three aspects as follows. Firstly, we use distributed computing and virtualization capabilities of cloud computing to realize the hierarchical transmission of data. Secondly, through this mechanism of hierarchical transmission and classification algorithm of machine learning, we realize hierarchical filtering of offensive data. Finally, by combining hierarchical transmission mechanism with threshold value, classification algorithm, and limit tolerance mechanism, we regulate the data amount to monitor fault sensor in real time. Experiments are conducted to assess the proposed architecture's performance. The results show that each layer acts as a protective screen to counterattack the offensive data, which shows good robustness, real-time, and adaptive ability. Moreover, to compare with OM mode, the identification efficiency of fault sensor of TM mode is improved by 2 times. Also, TM mode improves 33.33% identification acuity, which is suitable for the enterprises that are mainly based on streaming computing. In summary, the hierarchical private cloud architecture achieves the filtering of offensive data and the real-time identification of faulty sensor, which guarantees the security, accuracy, and integrity of the data transmission process.

INDEX TERMS Balance regulation, cloud computing, machine learning, threshold value.

I. INTRODUCTION

The development of big data has facilitated various enterprises such as semiconductor fabrication plant, colored filter manufacture [1], casting production [2], and commodity sales because of the potential value of data. To obtain knowledge, these enterprises collect data continuously for data analysis. By using the wisdom extracted from the knowledge, they can make the quality of products even better and swell the profits. Thus, it is important for enterprises to obtain reliable data which are helpful for effective data analysis. So, data reliability is a challenging problem in data collection. Regarding this problem, it is found that it relies mainly on the data transfer process. Traditionally, enterprises mostly use a kind of point-to-point transmission method (PTPTM), which transmits data from collection points to terminal servers without regulation mechanisms in the process. However, the

disadvantages in this method have attracted researches' interests. Many efforts have been conducted to overcome these shortcomings. For example, in order to utilize power properly and reduce the overall power consumption of network significantly, a grid-based data gathering algorithm called energy-efficient structured clustering algorithm with relay was proposed in [3]. In [4], the authors proposed a mobile and hierarchical data transmission architecture which aims to improve data transmission efficiency and achieve more efficient usage of energy. In [5], the authors mentioned an online and credible data integrity monitoring method for digital sensors. And they set up a motion tracking platform based on accelerometers, gyroscopes, and magnetometers for the integrity monitor of online trusted data to improve data quality. In [6], the authors described that we can use attack signature to identify attackers who try to utilize known network, operating system, or application vulnerability to attack the data in transmission. When the detection system finds a sequence of events that match any signature, it will trigger

The associate editor coordinating the review of this manuscript and approving it for publication was Claudio Agostino Ardagna .

an alarm. Also, this detection method has low false alarm rate.

In recent years, the above researches have proposed methods mainly on problems of high-power consumption, low data quality, and slow data transmission rate of PTPTM. They all have their own characteristics and the scope of applications.

However, in terms of convenient access to reliable data, the difficulties we still face are divided into the following two points. Firstly, fault sensors affect the data amount collected by the terminal servers. In the case of the insufficient data, when we carry out factor analysis and cluster analysis, we can get results by SPSS. But the analysis results will be biased, even completely wrong. Instead, if the data are too much, it may cause unnecessary resources waste of storage and computing. Secondly, offensive data invade the data that are being transmitted in complex network environment, which pose a security threat to the terminal servers.

Consequently, in this article, we proposed a hierarchical private cloud architecture which relies on the security characteristic of private cloud [7]. The construction of this architecture includes three steps as follows. Firstly, we turn the ideal of hierarchical transmission of data into reality with the help of distributed computing and virtualization capabilities of cloud computing. Secondly, we use classification algorithm of machine learning and the hierarchical transmission mechanism to achieve hierarchical filtering of offensive data. Thirdly, by the combination of hierarchical transmission mechanism, threshold value, classification algorithm, and limit tolerance mechanism, the hierarchical private cloud architecture allows to monitor fault sensors in real time. For better understanding, we will take semiconductor fabrication plant as an example for further explanation.

In fact, a semiconductor fabrication plant has many factories. Every factory has many production units. To collect status messages of production environment, there are multiple sensor devices in each production unit. Thus, the hierarchical private cloud architecture of semiconductor fabrication plant is divided into three layers: production unit layer, factory layer, and enterprise layer. Due to the strict requirements for production environment, monitoring workers need to analyze whether production environment is qualified by collecting status data of production environment in real time. So, Fig. 1 shows the hierarchical private cloud architecture which describes the real-time collection and transmission process of a large number of monitoring data. In this architecture, the sensors collect status data of each corresponding production unit. Then, these collected data are continuously transmitted to the corresponding clouds in production unit layer. After processing, each server of the production unit layer transmits these data to the corresponding clouds in factory layer. Finally, the servers of factory layer send these data to the final cloud of enterprise layer. At this time, environment monitoring workers begin to use data. During this process, the servers of different layers collect, process, and transmit data to the upper layer. But, the data in transmission can be easily invaded by the stubborn aggressive data. For

this, we make each layer generate its own filtering model protection screen after data training, which aims to block the offensive data according to their own filtering abilities. Besides, this architecture follows multiple calculation rules and regulation modes for fault sensors monitoring. And we set limit tolerance, which aims to help the classification algorithm of machine learning realize real-time recognition of abnormal sensors through adjusting the amount of data collected by sensor devices in different length of regulation cycle. Thus, to let cloud service providers provide enterprises with continuous and reliable data services [8], there is a need of achieving hierarchical filtering of offensive data and real-time detection of fault sensors, which helps environment monitoring workers conduct data analysis efficiently.

The **contributions** of this paper are summarized as the following three parts.

- 1) This paper makes an attempt to add multiple regulation mechanisms in the process of data transmission. Distributed computing and virtualization capabilities allow hierarchical private cloud architecture to transfer data from lower layer to upper layer step by step.
- 2) This paper gives the idea of multivariate filtering for offensive data. The hierarchical private cloud architecture achieves hierarchical filtering of offensive data through hierarchical transmission mechanism and classification algorithm of machine learning. Therefore, the security of data transmission process has strengthened. Meanwhile, the security of final collected data has also boosted.
- 3) This paper applies machine learning technology to the intelligent recognition of abnormal sensors under the real-time regulation mechanism. The hierarchical private cloud architecture combines hierarchical transmission mechanism with threshold value, classification algorithm, and extreme tolerance mechanism to achieve the balance regulation of data amount. Based on this, when the number of rounds of balance regulation (RD) is equal to limit tolerable adjustment times (MRT), the real-time monitoring of fault sensors can be realized.

The rest of this article is organized as follows. Section II represents the related works. Then, section III defines and explains the problem of this article. The system architecture and method are introduced in section IV. And, section V mentions experiments and analyses. Finally, the conclusion is in section VI.

II. RELATED WORKS

Many fields need data for data analysis, such as animal monitoring [9], smart factory [10], fault identification of intelligent manufacturing system [11], medical monitoring [12], intelligent manufacturing automation [13], metal forming process monitoring [14], monitoring of robot motion and vehicle collision [15], and monitoring of die-casting process [16]. In fact, sensors should firstly collect the data that are needed by these fields, and then transmit them to terminal servers

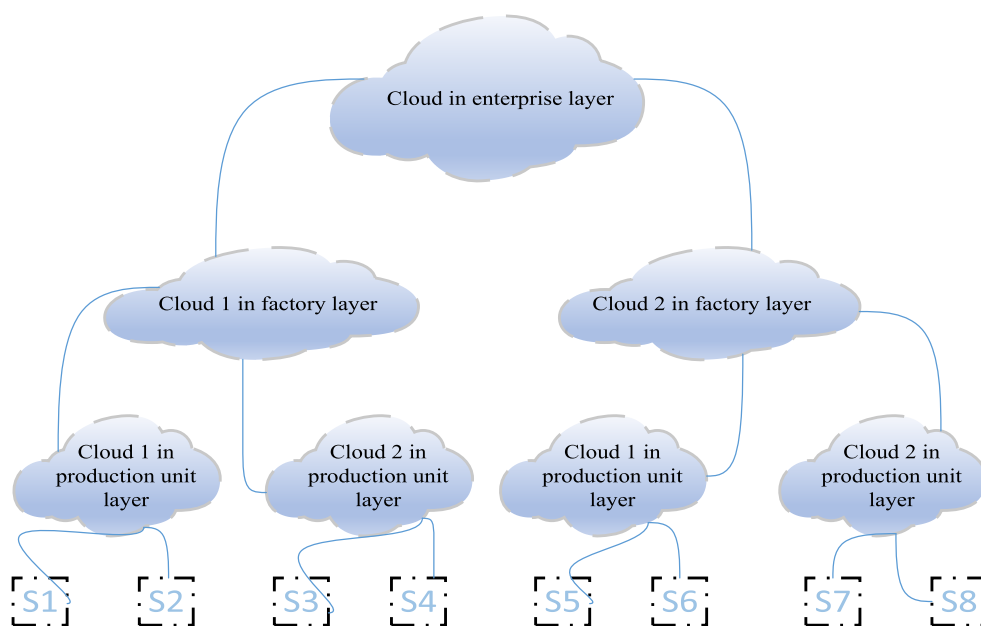


FIGURE 1. Hierarchical private cloud architecture for data collection and transmission of semiconductor fabrication plant.

through network. For this, most enterprises currently take a kind of data transmission form of PTPDTM, which does not have regulation mechanism in the process of transmission. Thus, this transmission form has many shortcomings, one of which is data security.

In this case, in [17], the authors mentioned some abnormal detection technologies of data transmission process, which includes statistics, clustering, etc. These technologies are divided into three categories according to the type of sample data: supervised, semi-supervised, and unsupervised. In [18], the authors compared several unsupervised approaches of intrusion detection in conventional computer networks, which are based on local outlier factor, near neighbors, Mahalanobis distance, and support vector machine (SVM). Their experiments show that the local outlier factor is the most adequate approach. In [19], regarding anomaly detection in intrusion detection systems (IDS) of wireless sensor networks (WSN), the authors surveyed the most popular technologies. Generally, the nodes which contain IDS components gather and analyze network status data concerning anomalous operation activities of their neighbors. When anomaly occurs, the nodes trigger an alarm at the base station. Moreover, some researchers proposed that the blockchain technology can be well used to overcome the security problem of data transmission of cloud computing [7]. In [20], the authors used geostatistics and time series analysis to detect abnormal values of the reading from meteorological sensor. And, they claimed that the temporal and spatial real-data-based outlier detection has high performance, which can identify all outliers. In [21], the authors proposed a two-stage algorithm. In the first stage, the algorithm uses a type of

SVM to look for the time anomalies. After this, in the second stage, the algorithm reduces the false positives, and then uses the K-nearest neighbor method to classify anomalies. In [22], the authors pointed out that Mahalanobis distance has high detection accuracy and robustness when it is used to detect internal attacks. In [23], the authors used one-class quarter-sphere SVM in two new anomaly detection algorithms: lightweight anomaly detection algorithms using sort and quick select. The algorithms are suitable to run in constrained nodes due to their low computational complexity. Moreover, their experiments show a high performance, e.g., 95% true positive rate and a false positive rate below 10%. In [24], the authors proposed a traffic anomaly detection algorithm for WSN. Through the network traffic analysis in the nodes, we can make traffic prediction and judge the anomaly in a WSN. The experiments in the literature show an accuracy higher than 96% and a false positive rate lower than 3%. In [25], the authors mentioned that the machine learning algorithms based on random forests have been successfully applied to anomaly detection of different fields in many scenarios. But their popularity has not reached the level of SVM yet. Furthermore, some algorithms such as deep belief networks [26], convolutional neural networks [27], and recursive neural networks [28] have been effectively used in several scenarios to improve the performance of previous technologies. In [29], in the area of anomaly detection, the authors used deep learning in combination with other technologies to identify outliers. It can be applied to high-dimensional and large-scale domains.

The above researches study the data security of data transmission process from many perspectives, including

preventing sensors from being invaded, abnormalities monitoring of data transmission, and identification of outliers. And, by using the technologies such as blockchain, clustering, and statistics, the effectiveness on offensive data monitoring of data transmission has been improved. However, none of these studies discusses the use of hierarchical transmission mechanism in the filtering of offensive data.

The accuracy of data in transmission process also attracts the scholars' attentions. Therefore, in [30], in the field of precision agriculture, the authors developed a heuristic algorithm that helps to decide which anomaly detection should be selected according to the agricultural environments. It can identify faulty sensors to discard data collected by them. In [31], the authors proposed that the accuracy of individual sensors can be easily impaired by a variety of factors. To improve the monitoring accuracy, they proposed a sensor fusion scheme based on Bayesian inference. In [32], the authors proposed a plan, which integrates knowledge reasoning and semantic data to collect and process data in real time for fault diagnosis and statistical analysis to avoid data errors. Also, in [33], the authors mentioned that unauthorized people should be warned to access sensors in restricted areas. Meanwhile, people can eliminate any potentially harmful sensors in time by setting up sensors that are able to trigger intrusion alarms. In [34], the authors proposed that the form of sensing all possible data items captured by a smart object and then sending the complete captured data to the cloud is less useful. And this approach would also lead to the waste of resources of network, storage, etc. Therefore, they proposed the fog computing that pushes everything in the core of cloud computing to the edge of the network. And we should process the data at the leaf node or edge.

In summary, for the data transmission process, the above researches analyze and solve the problem of data accuracy from many aspects, which include knowledge reasoning, semantic data integration, and spurious signal elimination. By using different methods such as machine learning, fog computing, and platform monitoring, the accuracy of the obtained data is gradually improved. However, none of these studies discuss the use of hierarchical transmission mechanism in data amount balance regulation and real-time monitoring of faulty sensors.

III. PROBLEM DESCRIPTION

In order to solve the problem mentioned in section I, this paper proposes the hierarchical private cloud architecture which aims to realize hierarchical transmission of data, hierarchical filtering of offensive data, and real-time monitoring of fault sensors. Next, this section gives a formal description of the problem and then explains its correlative concepts. For convenience, the symbols that will be used and their corresponding descriptions are shown in Table 1.

A. PROBLEM DESCRIPTIONS

There is an enterprise Δ which includes m factories ε . Every $\varepsilon_i (i = 1, \dots, m)$ has γ_i production units θ , each of which

TABLE 1. Symbol table.

Symbol	Description
Δ	The cloud of enterprise layer
ε_i, θ_i	The cloud i of factory layer and production unit layer separately
\emptyset_i	The sensor i
α_i^j	The data amount of round j collected by sensor i
$\alpha_{jgi}^{beforeFilter}, \alpha_{jfi}^{beforeFilter}$	The data amount of round j without being filtered of cloud i of production unit layer and factory layer separately
THE	Theoretical (qualified) data amount
α_{je}, α_{jf}	Actual data amount of round j collected by Δ and the cloud of factory layer separately
$\alpha_{jf}^i, \alpha_{jg}^i$	Actual data amount of round j for the cloud i of factory layer and production unit layer separately
β_{je}	The theoretical (qualified) data amount of round j of enterprise layer's cloud
$\beta_{jf}^i, \beta_{jg}^i$	The theoretical (qualified) data amount of round j for cloud i of factory layer and production unit layer separately
\cup_j^e, \cap_j^e	The left threshold value and the right threshold value of round j for enterprise layer
$\cup_{ij}^f, \cap_{ij}^f, \cup_{ij}^g, \cap_{ij}^g$	The left threshold value and the right threshold value of round j for the cloud i of factory layer and production unit layer separately
$\alpha_{status}^{fi}, \alpha_{status}^{gi}, \alpha_{status}^{si}$	The next round's data collection status of $\varepsilon_i, \theta_i, \emptyset_i$
$An APE$	A successful balance regulation
A_{sensor}, F_{sensor}	Abnormal sensor, fault sensor
T_i	Data collection time of sensor i
LOI	The under-collected/over-collected data amount of data transmission process (under-collected is negative, over-collected is positive)
ψ	Offensive data
BRT	Signal of number of rounds for data amount balance regulation
FNS	The collection status of next round for the clouds of factory layer
$GONS, GTNS$	The next round's collection status of the clouds of production unit layer under mode OM and mode TM separately
FD	The data collection amount of the clouds of factory layer
GOD, GTD	The data collection amount of the clouds of production unit layer under mode OM and mode TM separately

is responsible for fundamental production work of Δ . n_{γ_i} sensors \emptyset are set in the $\theta_i (i = 1, \dots, \gamma_i)$, which are responsible for collecting status data of production environment of θ_i . And the number of sensors in θ_i is not less than 1. When $\emptyset_i (i = 1, \dots, n_{\gamma_i})$ completes the data collection of round j , the collected data will be transmitted to the server terminal of Δ for data analysis after the process of filtering offensive data of each layer and transmitting data layer by layer. Suppose that the range of qualified data amount for Δ is $(\beta_{je} - x, \beta_{je} + x)$. And x is greater than 0. But the actual data amount is α_{je} . If α_{je} is in the range of qualified data amount, α_{je} is qualified. Therefore, workers can use these data to do effective data analysis. On the contrary, if α_{je} is not in the range of qualified data amount, the hierarchical private cloud architecture starts balance regulation of data amount. If this regulation is successful, then the users can use the collected data for effective data analysis. And A_{sensor} is not F_{sensor} . Instead, if this regulation is not successful when RD is equal to MRT , environment monitoring workers can identify the F_{sensor} .

B. DEFINITION 1 DATA SOURCE

Suppose that the n sensors, which are at the bottom of the hierarchical private cloud architecture, are responsible for collecting data and transmitting data to servers of θ_i where $\emptyset_i (i = 1, \dots, n)$ is located. These n sensors are expressed as the set \emptyset of (1).

$$\emptyset = \{\emptyset_1, \emptyset_2, \emptyset_3, \dots, \emptyset_n\}, \quad \forall \emptyset_i \in \{0, 1\}, 1 \leq i \leq n \quad (1)$$

If the value of $\emptyset_i (i = 1, \dots, n)$ of (1) is 1, the sensor works normally. But, if the value of $\emptyset_i (i = 1, \dots, n)$ is 0, the sensor is faulty.

C. DEFINITION 2 DATA FILTERING MECHANISM

In round j , the data amount collected by each $\emptyset_i (i = 1, \dots, n)$ is expressed as the set η of (2).

$$\eta = \{\alpha_1^j, \alpha_2^j, \alpha_3^j, \dots, \alpha_n^j\}, \quad \forall \alpha_i^j \geq 0 \quad (2)$$

1) THE DATA AMOUNT WAITING TO BE FILTERED

Suppose that there are n sensors in θ_i . For θ_i , at the end of the data collection of round j , the amount of the arriving data without being filtered is expressed as (3).

$$\alpha_{jgi}^{beforeFilter} = \sum_{i=1}^n \alpha_i^j + LOI + \Psi, \quad \forall i \in 1, \dots, n \quad (3)$$

2) DATA FILTERING

Suppose that $\xi = [\xi_1(x), \xi_2(x), \xi_3(x)]$ represents a three-tier filtering mechanism of data, which consists of production unit layer, factory layer, and enterprise layer. Therefore, the hierarchical filtering process of hierarchical private cloud architecture is shown in (4).

$$\left(\left(\sum_{i=1}^n \alpha_i^j + LOI + \Psi \right) \xi_1(x) \xi_2(x) \xi_3(x) - \sum_{i=1}^n \alpha_i^j \right) / LOI \approx 1 \quad (4)$$

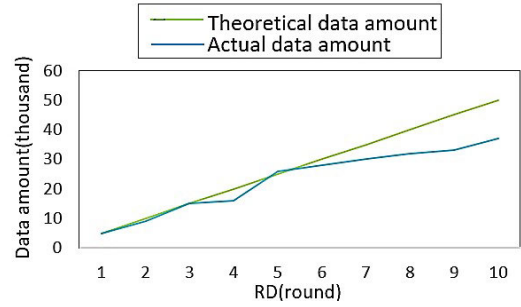


FIGURE 2. Comparison chart of theoretical (qualified) data collection amount and actual data collection amount.

D. DEFINITION 3 DATA AMOUNT REVIEW

Suppose that there are n sensors and m clouds of production unit layer. When the data amount of round j collected by θ_i and ε_i meets (5) and (6) respectively, the data amount is qualified.

$$\left(\left(\left(\sum_{i=1}^n \emptyset_i \alpha_i^j + LOI \right) \beta_{jg}^{i-1} \right) \cup_{ij}^{g-1} \geq 1 \right) \wedge \left(\left(\left(\sum_{i=1}^n \emptyset_i \alpha_i^j + LOI \right) \beta_{jg}^{i-1} \right) \cap_{ij}^{g-1} \leq 1 \right) \quad (5)$$

$$\left(\left(\left(\sum_{i=1}^m \alpha_{jg}^i + LOI \right) \beta_{jg}^{i-1} \right) \cup_{ij}^{g-1} \geq 1 \right) \wedge \left(\left(\left(\sum_{i=1}^m \alpha_{jg}^i + LOI \right) \beta_{jg}^{i-1} \right) \cap_{ij}^{f-1} \leq 1 \right) \quad (6)$$

E. DEFINITION 4 DATA COLLECTION STATUS AND ABNORMAL SENSOR A_{sensor}

Suppose that the qualified data amount THE of θ_i in round j is β_{jg}^i , as shown by the green line in Fig. 2. And the actual data amount collected in round j is α_{jg}^i , as shown by the blue line in Fig. 2. Therefore, we can analyze α_{jg}^i as follows (the following mechanisms are applicable to all layers).

- 1) If $\alpha_{jg}^i / \beta_{jg}^i \in [\cup_{ij}^g, \cap_{ij}^g]$, then the data amount of round j collected by θ_i is qualified. Therefore, the hierarchical private cloud architecture is in balance. And the data collection status α_{status}^{gi} of θ_i of round $j + 1$ is "true".
- 2) If $\alpha_{jg}^i / \beta_{jg}^i \notin [\cup_{ij}^g, \cap_{ij}^g]$ and $\alpha_{jg}^i / \beta_{jg}^i < \cup_{ij}^g$, then the data amount of round j collected by θ_i is not qualified. Therefore, the hierarchical private cloud architecture is out of balance. The data collected by A_{sensor} is too little. Thus, the data collection status α_{status}^{gi} of θ_i of round $j + 1$ is "increase". A_{sensor} is shown in (7).

$$A_{sensor} = \left\{ i \mid \min \left\{ \emptyset_1 \alpha_1^j, \emptyset_2 \alpha_2^j, \emptyset_3 \alpha_3^j, \dots, \emptyset_n \alpha_n^j \right\} \right\}, \quad i = 1, \dots, n \quad (7)$$

- 3) If $\alpha_{jg}^i / \beta_{jg}^i \notin [\cup_{ij}^g, \cap_{ij}^g]$ and $\alpha_{jg}^i / \beta_{jg}^i > \cap_{ij}^g$, then the data amount of round j collected by θ_i is not qualified. At this time, the hierarchical private cloud architecture is out of balance. The data collected by A_{sensor} is too

much. Therefore, the data collection status α_{status}^{gi} of round $j + 1$ of θ_i is “decrease”. A_{sensor} is shown in (8).

$$A_{sensor} = \left\{ i | \max \left\{ \vartheta_1 \alpha_1^j, \vartheta_2 \alpha_2^j, \vartheta_3 \alpha_3^j, \dots, \vartheta_n \alpha_n^j \right\} \right\}, \quad i = 1, \dots, n \quad (8)$$

F. DEFINITION 5 BALANCE REGULATIONS

We assume that MRT is 3. And $\tau(x) = [\tau_1(x), \tau_2(x), \tau_3(x)]$ represents three rounds of balance regulations of data amount. x is a tuple of parameters. The three elements of $\tau(x)$ have strict constraints of order. When the data amount of round j is unqualified, hierarchical private cloud architecture starts to regulate the amount of data. After no more than MRT times of regulations, if the amount of data collected by the terminal server of Δ is qualified, the balance regulation of data amount is successful. Moreover, the successful regulation is shown in (9).

$$\begin{aligned} & \left(\left(\sum_{j=1}^3 ((\tau_j(x) (\sum_{i=1}^n \vartheta_i \alpha_i^j + LOI) \beta_{ff}^{i-1}) / \cup_j^e \geq 1) \right. \right. \\ & \quad \wedge ((\tau_j(x) (\sum_{i=1}^n \vartheta_i \alpha_i^j + LOI) \beta_{ff}^{i-1}) / \cap_j^e \leq 1) \left. \right) \geq 1) \\ & \quad \wedge \left(\left(\sum_{j=1}^3 ((\tau_j(x) (\sum_{i=1}^n \vartheta_i \alpha_i^j + LOI) \beta_{ff}^{i-1}) / \cup_j^e > 1) \right. \right. \\ & \quad \left. \left. \wedge ((\tau_j(x) (\sum_{i=1}^n \vartheta_i \alpha_i^j + LOI) \beta_{ff}^{i-1}) / \cap_j^e < 1) \right) \leq 3 \right) \quad (9) \end{aligned}$$

G. DEFINITION 6 FAULT SENSOR

After MRT times of balance regulations of data amount, if (9) is not met, then the A_{sensor} is F_{sensor} . Therefore, users should inform the equipment maintenance workers to repair the F_{sensor} . The judgment rule of F_{sensor} is shown in (10).

$$\begin{aligned} & \left(\sum_{j=1}^3 ((\tau_j(x) (\sum_{i=1}^n \vartheta_i \alpha_i^j + LOI) \beta_{ff}^{i-1}) / \cup_j^e \geq 1) \right. \\ & \quad \left. \wedge ((\tau_j(x) (\sum_{i=1}^n \vartheta_i \alpha_i^j + LOI) \beta_{ff}^{i-1}) / \cap_j^e \leq 1) \right) < 1 \quad (10) \end{aligned}$$

IV. SYSTEM OVERVIEW AND METHOD

A. SYSTEM MODEL

This paper proposes the hierarchical private cloud architecture, which aims to achieve hierarchical transmission of data, hierarchical filtering of Ψ , and real-time monitoring of F_{sensor} . Fig. 3 shows the system model of this architecture, which is divided into two parts: 1) The control module from the cloud of upper layer to the cloud of lower layer; 2) The control module from the cloud of lower layer to sensors. The operation process of this model includes six points as follows.

Firstly, sensors collect data of monitoring area and transform them. Secondly, the collected data will be sent out. Thirdly, with the help of the WSN, these data will be processed and transmitted. Fourthly, they will be sent to the servers of the “lower private cloud” where the corresponding sensors are located. By using the recognition model of offensive data, the servers clean and filter Ψ . Through mechanism of threshold value, recognition model of sensor, and limit

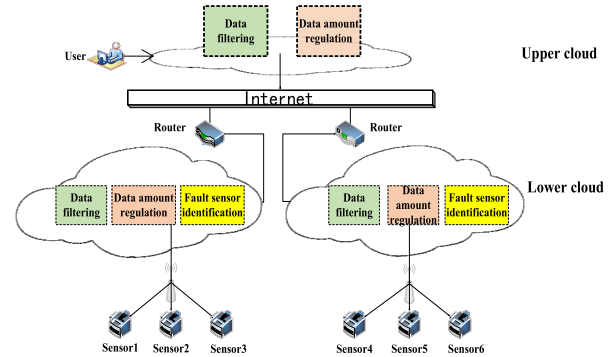


FIGURE 3. Hierarchical private cloud architecture model.

tolerance mechanism, the servers regulate the data amount to monitor F_{sensor} in real time. After this, these data will be sent out again. Fifthly, with the help of transmission lines, they are transmitted to the routers which are responsible for analyzing data packets and obtaining destination address. Finally, using the optimal routing algorithm, these data are sent to the corresponding servers of the “upper private cloud”. At this time, through the recognition model of offensive data based on machine learning, the server filters the Ψ . Relying on the threshold value and the recognition model of sensor based on machine learning, the server adjusts the data collection status to balance and regulate the “lower private cloud”.

Furthermore, there are two modes for balancing and regulating the data amount in the operation process of system model. They are called OM mode and TM mode respectively. For the data collection status (increase, decrease, and true) of next round of ε_i , these two modes have their own regulation rules. The corresponding regulation rules of the two modes are described as follows. Here, for ease of understanding, we take the cloud of the factory layer ε_i as the cloud of top layer Δ to explain.

1) OM MODE

- If the data collection status α_{status}^{fi} of round $j+1$ for ε_i is “increase”.

If the data collection status α_{status}^{gi} of round $j + 1$ for θ_i is “increase”, considering the α_{status}^{fi} of ε_i , the left threshold value of round $j + 1$ of θ_i should plus 0.2 based on \cup_{ij}^g . And, the right threshold value \cap_{ij+1}^g should plus 0.2 based on \cap_{ij}^g . If the data collection status α_{status}^{gi} of round $j + 1$ of θ_i is “decrease”, then θ_i should obey its own status change of data collection. And, it is supposed to take the status change of data collection of ε_i into account. Therefore, the left threshold value \cup_{ij+1}^g of round $j+1$ of θ_i should plus 0.1 based on \cup_{ij}^g . And, the right threshold value \cap_{ij+1}^g should plus 0.1 based on \cap_{ij}^g . If the data collection status α_{status}^{gi} of round $j + 1$ of θ_i is “true”, then θ_i obeys its own status change of data collection. Also, it takes the status change of data collection of ε_i into account. That is, the left threshold value \cup_{ij+1}^g of round $j+1$ of

θ_i plus 0.1 based on \cup_{ij}^g . And, the right threshold value \cap_{ij+1}^g plus 0.1 based on \cap_{ij}^g .

- If the data collection status α_{status}^{fi} of round $j + 1$ for ε_i is “decrease”.

If the data collection status α_{status}^{gi} of round $j + 1$ of θ_i is “increase”, considering the data collection status of ε_i , the left threshold value \cup_{ij+1}^g of round $j + 1$ of θ_i should minus 0.1 based on \cup_{ij}^g . And, the right threshold value \cap_{ij+1}^g should minus 0.1 based on \cap_{ij}^g . If the data collection status α_{status}^{gi} of round $j + 1$ of θ_i is “decrease”, then θ_i is subject to the data collection status change of the layer where it is located. And it takes the status change of data collection of ε_i into account. That is, the left threshold value \cup_{ij+1}^g of round $j + 1$ of θ_i minus 0.2 based on \cup_{ij}^g . And the right threshold value \cap_{ij+1}^g minus 0.2 based on \cap_{ij}^g . If the data collection status α_{status}^{gi} of round $j + 1$ of θ_i is “true”, then θ_i follows the status change of data collection of the layer where it is located. And it takes the status change of data collection of ε_i into account. Therefore, the left threshold value \cup_{ij+1}^g of round $j + 1$ of θ_i minus 0.1 based on \cup_{ij}^g . And the right threshold value \cap_{ij+1}^g minus 0.1 based on \cap_{ij}^g .

- If the data collection status α_{status}^{fi} of round $j + 1$ for ε_i is “true”.

If the data collection status α_{status}^{gi} of round $j + 1$ of θ_i is “increase”, according to the data collection status of ε_i , the left threshold value \cup_{ij+1}^g of θ_i is supposed to plus 0.1 based on \cup_{ij}^g . And the right threshold value \cap_{ij+1}^g plus 0.1 based on \cap_{ij}^g . If the data collection status α_{status}^{gi} of round $j + 1$ of θ_i is “decrease”, then θ_i is subject to the status change of data collection of the layer where it lies. And it takes the status change of ε_i into account. Therefore, the left threshold value \cup_{ij+1}^g of θ_i minus 0.1 based on \cup_{ij}^g . And the right threshold value \cap_{ij+1}^g minus 0.1 based on \cap_{ij}^g . If the data collection status α_{status}^{gi} is “true”, according to the free demands, data analysis workers set threshold values of θ_i for next round casually.

2) TM MODE

- If the data collection status α_{status}^{fi} of round $j + 1$ for ε_i is “increase”, no matter what the data collection status of round $j + 1$ of θ_i is, θ_i follows the status change of ε_i . Therefore, the left threshold value \cup_{ij+1}^g of round $j + 1$ changes to \cup_{ij}^g . And the right threshold value \cap_{ij+1}^g will be added 40 based on \cap_{ij}^g .
- If the data collection status α_{status}^{fi} of round $j + 1$ for ε_i is “decrease”, regardless of what the data collection status of round $j + 1$ of θ_i is, θ_i obeys the status change of ε_i . Thus, the left threshold value \cup_{ij+1}^g of round $j + 1$ of θ_i changes to 0. And the right threshold value \cap_{ij+1}^g of θ_i goes to \cup_{ij}^g .

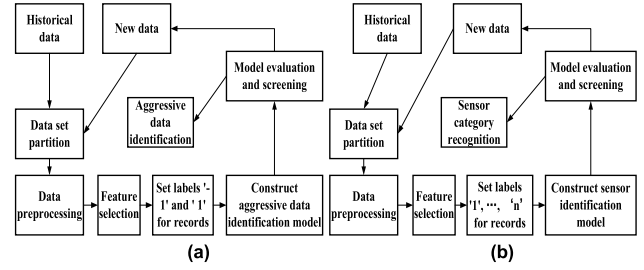


FIGURE 4. Model workflow.

- If the data collection status α_{status}^{fi} of round $j + 1$ for ε_i is “true”, according to the demand, data analysts will set threshold values of θ_i for next round arbitrarily.

B. PROPOSED ALGORITHMS

In this paper, we apply decision tree algorithm of machine learning, threshold value mechanism, and limit tolerance mechanism to the hierarchical private cloud architecture. Among them, decision tree algorithm is a kind of supervised learning. Not only is this algorithm easy to implement, but also users do not need to know too much background knowledge of it. So, it can achieve a feasible and effective classification for a large number of data sources quickly which is an effective classification method. Moreover, the model, which is generated by this algorithm, is an interpretable model. It represents a mapping relationship between object attributes and object values. Each internal node represents a test on an attribute. And each leaf node represents a category. Next, the whole system method of hierarchical private cloud architecture will be explained as follows.

Fig. 4 shows the workflow of system model for hierarchical private cloud architecture. For sample division, we use `train_test_split` of python for each model. Fig. 4 (a) is the working process of offensive data identification model. This model classifies the mixed data of every layer to identify offensive data of each layer. After identifying the offensive data, the hierarchical private cloud architecture needs to filter them. Besides, Fig. 4 (b) shows the working process of the sensor recognition model. This model works at θ_i , which is responsible for identifying A_{sensor} and F_{sensor} . The workflow of these two models will be elaborated as follows.

In the workflow of offensive data identification model, we establish a data set firstly. The source of the data set is the log data saved by enterprises and the new data. Secondly, 80% of the sample data are divided into the training set. In addition, 20% of the sample data are divided into the test set. Thirdly, we should carry out data preprocessing which aims to clean up abnormal data and outliers. It can achieve data consistency. Besides, each value can be kept as two decimal places, the purpose of which is to ensure format standardization. Fourthly, we need perform feature division on the data set. So, feature selection is supposed to be carried out. For this, we select n features as the main features. Fifthly, we use “-1” and “1” to mark offensive data and non-offensive data respectively. Sixthly, we train the data of training set. After

getting the trained model, we input each group of feature vectors of the test set into the model. Then, we are able to test the effectiveness of the obtained model. Seventhly, through the value of AUC and the evaluating accuracy, we evaluate the classification ability of the model. After the construction and the data verification of six models, we can select the model which has the highest AUC as the optimal model. Finally, we input new data to the model regularly to update and optimize the offensive data identification model continuously.

In the workflow of sensor recognition model, by integrating the data that from each sensor, we construct the data set firstly. Secondly, we divide this data set. 80 percent of the data are set as the training set. And, 20 percent of the data are set as the test set. Thirdly, by cleaning up abnormal data and outliers, we perform the data preprocessing. Fourthly, we analyze the data set to select n features as the main features. Fifthly, to mark training set and test set, we use “1”, “2”, “3”, ..., “ n ” to represent sensors respectively. Sixthly, we train the data of the train set to obtain model.

Then, according to the evaluation accuracy and AUC, we evaluate the classification ability of the model. Seventhly, the sensor recognition model which has the highest AUC is selected as the optimal model. Finally, we put new data into the model regularly, which aims to update and optimize the obtained model continuously.

The workflow of these two models is summarized as follows. In the workflow of the offensive data recognition model and the sensor recognition model, we integrate the historical data and the new data that have new features into the data set firstly. Secondly, we divide the data set into training set and test set. Thirdly, let the data set be formatted and consistent. Fourthly, we perform feature selection and set class labels for the data set. Fifthly, to get the parameter values of the models, we input the training set to the models to train the data. Sixthly, we input the test set to the models to verify, evaluate, and screen the obtained models. Finally, we optimize the models regularly according to the updated training set.

Next, we will describe the balance regulation algorithm of the hierarchical private cloud architecture in Algorithm 1, Algorithm 2, and Algorithm 3. Among them, Algorithm 2 and Algorithm 3 use the two modes (OM mode and TM mode) respectively. The characteristic of OM mode is that the lower-layer clouds obey the data collection status changes of the upper-layer cloud where the lower-layer clouds are located. Also, the upper clouds and the lower clouds take data collection status changes of each other into account. On the contrary, the characteristic of TM mode is that the lower-layer clouds obey the data collection status changes of the upper-layer cloud absolutely. And they sacrifice their own status changes.

In Algorithm 1, the execution process is described as follows. Firstly, ε_i obtains the data collected by θ_i , as shown in ① of Fig. 5. The amount of these data is $\alpha_{j+1}^{beforeFilter}$. Secondly, the offensive data recognition model processes these data to identify Ψ , which can be seen in ⑤ of Fig. 5. Thus, ② of Fig. 5 shows that the Ψ can be filtered. Thirdly, ζ of Fig. 5 is

Algorithm 1 Cloud-to-Cloud Balance Regulation Algorithm (Mode TM)

Input: $\cup_{ij+1}^f, \cap_{ij+1}^f$

Output: MRT, α_{status}^{fi}

- 1: Obtain data collected by θ_i in round j
 - 2: Identify Ψ
 - 3: Filter Ψ
 - 4: Categorize the data
 - 5: Calculate β_{jf}
 - 6: **while** true **do**
 - 7: **if** $RD = MRT$ **then**
 - 8: Pass BRT to θ_i
 - 9: Jump out of this loop
 - 10: **end if**
 - 11: Obtain data collected by θ_i in round $j+1$
 - 12: Identify Ψ
 - 13: Filter Ψ
 - 14: Calculate α_{j+1f}^i
 - 15: **if** $\alpha_{j+1f}^i / \beta_{jf} < \cup_{ij+1}^f$ **then**
 - 16: Modify α_{status}^{fi} of the round $j+2$ to “increase”
 - 17: Adjust the threshold values of next round
 - 18: **else if** $\alpha_{j+1f}^i / \beta_{jf} > \cap_{ij+1}^f$ **then**
 - 19: Modify α_{status}^{fi} of the round $j+2$ to “decrease”
 - 20: Adjust the threshold values of next round
 - 21: **else**
 - 22: Modify α_{status}^{fi} of the round $j+2$ to “true”
 - 23: Set threshold values of next round according to the actual demand
 - 24: **end if**
 - 25: Add new data to the backup data set
 - 26: **end while**
-

calculating the theoretical data amount β_{jf} . Next, it starts the data amount balance regulation. Firstly, ε_i obtains the data of round $j+1$ collected by θ_i . And the amount of these data is $\alpha_{j+1}^{beforeFilter}$, as shown in ① of Fig. 5. Secondly, identify and process Ψ , which is depicted in ⑤ and ② of Fig. 5. Thirdly, calculate the data amount α_{j+1f}^i of this round. And, ③ and ④ of Fig. 5 show the steps of modifying the data collection status α_{status}^{fi} of round $j+2$. Finally, according to TM mode, input threshold values to perform continuous balance regulations, which is represented in ⑦ of Fig. 5. Then, some key steps in this algorithm and their main contributions are below.

Line 8 enables the signal of number of data collection times to be transmitted from the upper layer to the lower layer. The upper layer and lower layer coordinate and restrict each other, which uses the distributed computing and virtualization capabilities of cloud computing to make the calculations in hierarchical private cloud architecture of enterprise more efficient and stable. Lines 15 to 24 describe the process of adjusting the data collection status and the threshold values of upper clouds according to the TM mode when the upper clouds are out of balance. The dynamic balance regulation of data collection between upper and lower clouds is realized. To compare with

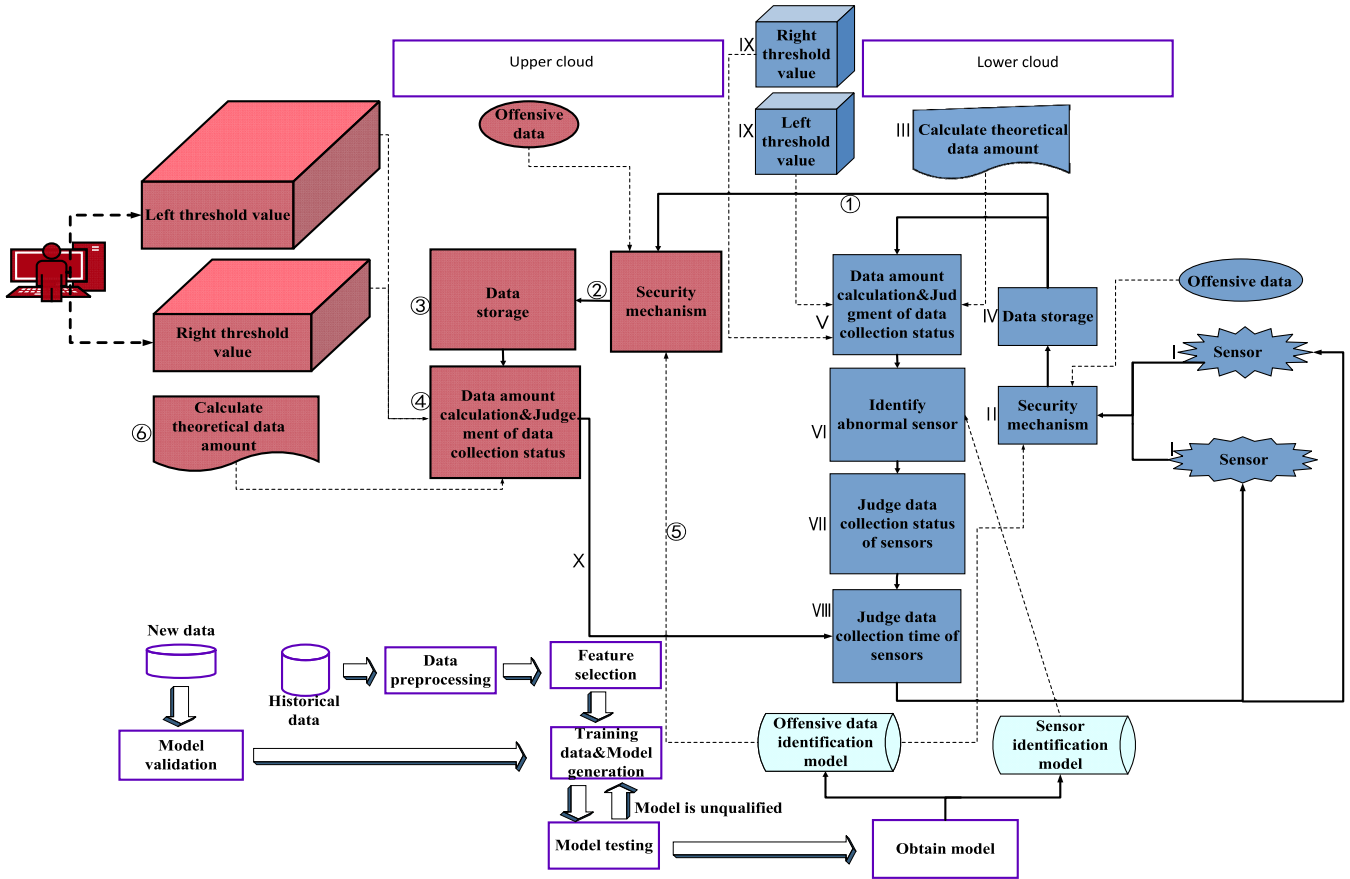


FIGURE 5. Balance regulation architecture diagram of data amount in hierarchical private cloud architecture.

the periodic equipment detection, this adaptive regulation method reduces the disadvantages of untimely monitoring and strengthens the security of data collection. In line 12, the upper clouds identify the offensive data according to their own recognition abilities under TM mode. It establishes the multi-layer security mechanism, which increases the security of hierarchical private cloud architecture.

In Algorithm 2, θ_i firstly obtains the data collected by ϑ_i in round j . The amount of these data is $\alpha_{jg}^{beforeFilter}$ like I of Fig. 5. Secondly, input these data to the offensive data recognition model. Its purpose is to identify the data to filter offensive data Ψ , which can be seen from II of Fig. 5. Thirdly, input the filtered data into the sensor recognition model to identify data category like VI of Fig. 5. Fourthly, calculate the theoretical data amount β_{jg}^i , as shown in III of Fig. 5. At this time, the architecture starts the balance regulation of data amount. When the *BRT* signal shows that *RD* reaches *MRT*, faulty sensor should be identified, which is represented in VI of Fig. 5. Otherwise, gather the data collected by ϑ_i in round $j+1$ firstly. The data amount is $\alpha_{j+1g}^{beforeFilter}$, which is shown in I of Fig. 5. Secondly, II of Fig. 5 displays the step of inputting these data into the offensive data recognition model to identify and filter Ψ . Thirdly, calculate the data amount α_{j+1g}^i of this round. Then, modify the next round's α_{status}^{gi}

of θ_i according to α_{j+1g}^i , like IV and V of Fig. 5. At this moment, if the data collection status of ε_i or θ_i is not “true”, the architecture is supposed to identify the abnormal sensor A_{sensor} , which can be seen from VI of Fig. 5. Then, modify the data collection status and T_i of A_{sensor} , as shown in VII, VIII, and X of Fig. 5. Finally, use the data balancing rules of OM mode to adjust the threshold values, the purpose of which is to perform continuous balance regulations, which is shown in IX of Fig. 5. Next, the key steps of Algorithm 2 and their main contributions are below.

Lines 16 to 22 show the process of adjusting the threshold values and the data collection status of lower clouds according to the OM mode when the lower clouds are out of balance. Thus, the dynamic balance regulation of data collection between upper and lower clouds can be achieved. Line 12 and line 17 depict that the lower clouds identify the offensive data and abnormal sensors based on their own offensive data recognition abilities and abnormal sensors recognition abilities in OM mode. It ensures the architecture continuously learns independently to make itself acquire improvement and progress, thus automatically raising the recognition accuracy of offensive data and abnormal sensors persistently. It helps to adapt to various faults and avoid over monitoring, which shows high completeness.

Algorithm 2 Cloud-to-Sensor Balance Regulation Algorithm (Mode OM)

Input:
 BRT signal of ε_i
 α_{status}^{fi}
 Threshold values of θ_i

Output: A_{sensor}

- 1: Obtain data collected by θ_i in round j
- 2: Identify Ψ
- 3: Filter Ψ
- 4: Categorize the data
- 5: Calculate the theoretical data amount β_{jg}^i of θ_i
- 6: **while true do**
- 7: **if** $RD == MRT$ **then**
- 8: Identify F_{sensor}
- 9: Break
- 10: **end if**
- 11: Obtain the data collected by θ_i in round j+1
- 12: Identify Ψ
- 13: Filter Ψ
- 14: Calculate α_{j+1g}^i
- 15: Modify α_{status}^{gi} of next round of θ_i
- 16: **if** $\alpha_{status}^{fi} \neq \text{“true”}$ or $\alpha_{status}^{gi} \neq \text{“true”}$ **then**
- 17: Identify A_{sensor}
- 18: Modify α_{status}^{si} and T_i of A_{sensor}
- 19: Adjust threshold values of next round of θ_i according to OM mode
- 20: **else**
- 21: Set threshold values of next round according to the demand freely
- 22: **end if**
- 23: **end while**

In Algorithm 3, firstly, θ_i obtains the data of round j collected by θ_i , which is displayed in I of Fig. 5. Secondly, II of Fig. 5 shows that the offensive data recognition model identifies these data to filter Ψ . Thirdly, the sensor recognition model identifies and classifies these data that have been filtered like VI of Fig. 5. Fourthly, calculate the theoretical data amount β_{jg}^i , as shown in III of Fig. 5. Next, start the balance regulation of data amount. When the BRT signal shows that RD has reached MRT , the faulty sensor need be identified, which is shown in VI of Fig. 5. Otherwise, gather the data of round j+1 collected by θ_i firstly, which is depicted in I of Fig. 5. Secondly, input these data into the offensive data recognition model to identify and filter Ψ , which can be seen from II of Fig. 5. Thirdly, IV and V of Fig. 5 show the step of calculating the data amount α_{j+1g}^i collected in this round. Fourthly, modify θ_i 's data collection status α_{status}^{gi} of next round according to α_{j+1g}^i . When the data collection status of ε_i is not “true”, the A_{sensor} should be identified like VI of Fig. 5. At the same time, the data collection status and data collection time of A_{sensor} of next round should be modified, as shown in VII, VIII, and X of Fig. 5. Finally, IX of Fig. 5 displays the step of using the balance

Algorithm 3 Cloud-to-Sensor Balance Regulation Algorithm (Mode TM)

Input:
 α_{status}^{fi}
 BRT Signal of ε_i
 Threshold Values of θ_i

Output: A_{sensor}

- 1: Obtain Data Collected by θ_i in Round j
- 2: Identify Ψ
- 3: Filter Ψ
- 4: Categorize the Data
- 5: Calculate the Theoretical Data Amount β_{jg}^i for θ_i
- 6: **while True do**
- 7: **if** $RD == MRT$ **then**
- 8: Identify F_{sensor}
- 9: Break
- 10: **end if**
- 11: Obtain Data Collected by θ_i in Round j+1
- 12: Identify Ψ
- 13: Filter Ψ
- 14: Calculate α_{j+1g}^i
- 15: Modify α_{status}^{gi} of Next Round of θ_i
- 16: **if** $\alpha_{status}^{fi} \neq \text{“true”}$ **then**
- 17: Combine α_{status}^{fi} and α_{status}^{gi} to Identify A_{sensor}
- 18: Modify α_{status}^{si} and T_i of A_{sensor}
- 19: Adjust θ_i 's Threshold Values of Next Round According to TM Mode
- 20: **else**
- 21: Set Threshold Values of Next Round According to the Demand Freely
- 22: **end if**
- 23: **end while**

method of TM mode to adjust the threshold values. Next, carry out continuous balance regulations. Besides, the key steps of Algorithm 3 and their main contributions are as follows.

Lines 16 to 22 are the process of adjusting the data collection status and the threshold values of lower clouds based on TM mode when the lower clouds are out of balance. It contributes to the dynamic balance regulation of data collection between upper and lower clouds. To compare with the real-time physical monitoring, this self-adaptive regulation architecture can adjust the internal balance system adaptively and reduce the unnecessary waste of resources of humans and hardware. Moreover, the TM mode provides guiding significance for stream computing companies. Line 12 and line 17 show that in TM mode, the lower clouds identify the offensive data and abnormal sensors according to their own offensive data recognition abilities and abnormal sensors recognition abilities respectively. Thus, with the idea of using technology for energy consumption, there is no need for manual tracking. It helps to avoid unnecessary maintenance operations, shortening the cost of maintenance.

TABLE 2. Experiment configuration.

Hardware Configuration	Software configuration
Windows 10	
Intel (R) Core (TM) i5-8250U	Pycharm2020.2.3 (Commjunity Edition)
CPU @1.60GHZ 1.80GHZ	

V. EXPERIMENT AND ANALYSIS

In the experiments, we verify the performance of hierarchical private cloud architecture for the following characteristics: 1) Hierarchical filtration of offensive data; 2) Real-time monitoring of fault sensors through real-time regulations of data amount. The experiments include two parts: experiment with simulation data and experiment with real data. For the former, it uses threads to simulate sensors which are responsible for generating data. In addition, we use the literature's data as data source for the experiment with real data. For easy understanding, we take the factory layer as the enterprise layer. Of course, the data analysts are set at the factory layer.

A. EXPERIMENT WITH SIMULATION DATA

In the experiment with simulation data, the factory layer has a cloud ε_i . Two clouds θ_1 and θ_2 are set at the production unit layer. Moreover, two sensors ϑ_1 and ϑ_2 are set at θ_1 . ϑ_3 and ϑ_4 are set at θ_2 . To generate data, two threads are used to simulate the sensors ϑ_1 and ϑ_2 of θ_1 . And another two threads are used to simulate the sensors ϑ_3 and ϑ_4 of θ_2 . They collect and transmit data according to their respective data collection time T_1, T_2, T_3 , and T_4 .

This experiment includes 11 rounds of data collections. And Table 2 shows the relevant configuration information of it. Since we monitor the faulty sensors by regulating the data amount in real time, we need to firstly verify the effectiveness of real-time data amount regulations for hierarchical private cloud architecture. Thus, we carried out corresponding experiments with simulation data for this. Fig. 6 shows the results of experiment, which describe the balance regulation process under OM mode and TM mode separately. According to the definition 4, this regulation process includes three statuses: "increase", "decrease", and "true". So, to express the charts concisely, we use "1", "-1", and "0" to represent "increase", "decrease", and "true" respectively. Moreover, other variables are designed as follows: 1) *THE* is calculated by (11); 2) *MRT* is set to 3; 3) The limit regulation tolerance (*TE*) is shown as (13); 4) The preset total regulation times is set to 9. Also, the mutation rate (*VNR*) is represented by (14). For (11), after classification, all data of each class are regarded as an element to form the set η . n is the number of sensors in a cloud or the number of lower clouds in an upper cloud. If the data amount of an element is the largest, then $\max(\eta)$ is equal to the data amount of this element.

$$THE = \frac{(\min(\eta) + \max(\eta))}{2}n \quad (11)$$

Fig. 6 shows that the hierarchical private cloud architecture has carried out 9 rounds of balance regulations of data

amount, which succeeded twice. Thus, an *APE* requires 3 or 4 rounds of data amount regulations under OM mode ((a) (c) (e)). Instead, under TM mode ((b) (d) (f)), there are 5 times of successful balance regulations. That is, an *APE* requires one round of balance regulation under TM mode. So, we can conclude that the TM mode improves 33.33% balance acuity compared with the OM mode. Therefore, according to the results of experiment, we can find that the hierarchical private cloud architecture has successfully achieved real-time balance regulations of data amount. By using (12), we can also evaluate the balance regulation efficiency (ω). For this, the smaller the ω is, the poorer the performance of balance regulation is. Based on this, we can get the balance regulation efficiency of OM mode which is $\log_{10} 0.3$. And, the balance regulation efficiency of TM mode is $\log_{10} 0.6$, which shows that the TM mode is higher than the OM mode in this aspect. Moreover, by comparing (a) (c) (e) and (b) (d) (f) of Fig. 6, we find that the cloud of factory layer changes its data collection status continuously to regulate the clouds of production unit layer. Also, the clouds of production unit layer adjust the collection form continuously to regulate the data collection time of sensors. Thus, the cloud-to-cloud regulation mechanism and the cloud-to-sensor regulation mechanism work well in the hierarchical private cloud architecture. The architecture proposed in this paper has good robustness. And, servers of all layers work coordinately, continuously, and consistently, which ensures the security, consistency, and integrity of the data transmission process (12), as shown at the bottom of the next page.

For the next experiments, we analyze the effectiveness of real-time monitoring for fault sensors. Fig. 7 is the VF-IF chart of data collection amount under OM mode. During this process, for the first round, we calculate the qualified data amount *THE* using (11). Then, before the second round of data collection, data users input left threshold value \cup_{i2}^f and right threshold value \cap_{i2}^f of the second round. When the second round is over, it can be seen that the third round of data collection status α_{status}^f of ε_i is "increase". It means that the amount of data collected by ε_i is much smaller than *THE* in the second round. At this time, to balance data amount, hierarchical private cloud architecture learns autonomously for identification of abnormal sensors A_{sensor} . Before the third round of data collection, according to the regulation mechanism of mode OM, data users input the left threshold value \cup_{i3}^f and the right threshold value \cap_{i3}^f . At the end of the third round of data collection, the fourth round of collection status α_{status}^f of ε_i is "decrease", which means that the regulation has not been successful. But, at this time, data users can select data for data analysis. Simultaneously, hierarchical private cloud architecture studies autonomously for recognition of A_{sensor} , which aims to balance the data amount. After this, before the fourth round of data collection, according to the OM mode, data user input the left threshold value \cup_{i4}^f and the right threshold value \cap_{i4}^f of round four. When the fourth round of data collection is over, the fifth round of

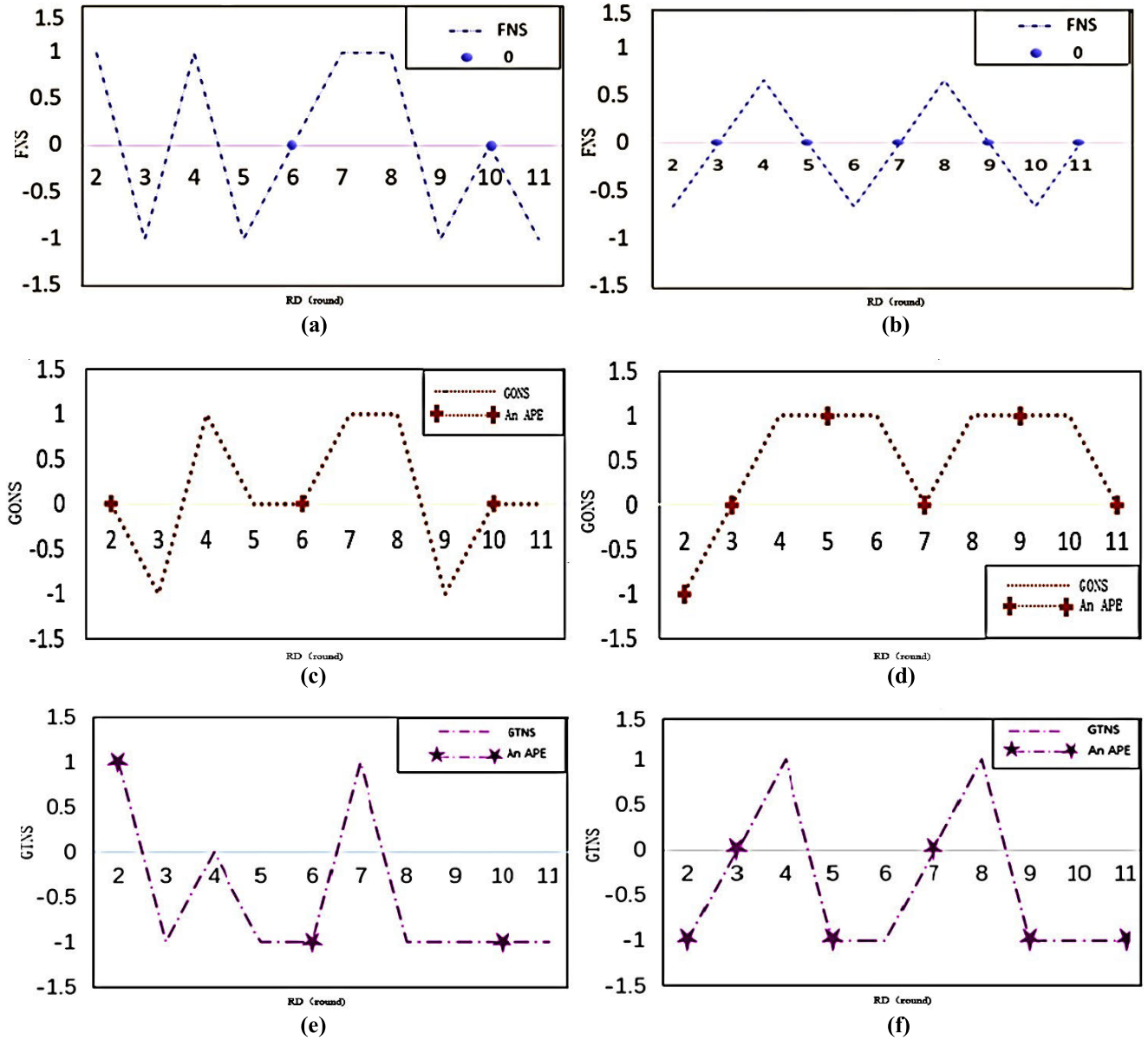


FIGURE 6. Data collection status change diagram.

α_{status}^{fi} of ε_i is “increase”, which shows that the regulation is not successful. At this time, the hierarchical private cloud architecture studies autonomously to identify A_{sensor} to balance the data amount. Before the fifth round of data collection, according to the regulation mechanism of OM mode, data users input the left threshold value \cup_{i5}^f and the right threshold value \cap_{i5}^f of the fifth round. When the fifth round of data collection finishes, we can observe that the sixth round of data collection status α_{status}^{fi} of ε_i is “decrease”. It shows that the regulation fails. Therefore, hierarchical private cloud

architecture begins to study automatically to identify A_{sensor} . At this moment, the number of the regulation is equal to MRT . This means that it has reached the limit tolerable regulation times. However, the balance regulation of data amount has not been successful yet. Thus, the abnormal sensor A_{sensor} is the fault sensor F_{sensor} . Meanwhile, data users should inform the maintenance workers to repair F_{sensor} . After the successful maintenance, according to regulation mechanism of OM mode, data users input the left threshold value \cup_{i6}^f and the right threshold value \cap_{i6}^f of round six. When the sixth round of

$$\omega = \log_{10} \frac{(successful\ balance\ regulation\ times + 1)}{(last\ balance\ regulation\ times - first\ balanced\ regulation\ times + 1)} \tag{12}$$

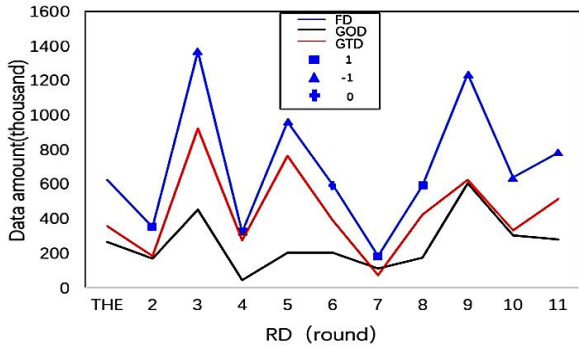


FIGURE 7. Data amount change curve of mode OM.

data collection is over, data collection status α_{status}^{fi} is “true”. Thus, the balance regulation of data amount is successful. So, after several rounds of balance regulations of data amount, the insufficient data of round two is compensated. At this time, the hierarchical private cloud architecture has restored the balance of data amount. It is an *APE* that from the end of round two to the success of regulation of data amount. Next, data users can input the left threshold value \cup_{i7}^f and the right threshold value \cap_{i7}^f according to their requirements. And the hierarchical private cloud architecture will regulate the data amount based on the same principle as the previous *APE*.

Fig. 8 is the VF-IF chart of the data collection amount regulation under TM mode. For the first round, by using (11), hierarchical private cloud architecture calculates the qualified data amount *THE*. Before the second round of data collection, data users input the left threshold value \cup_{i2}^f and the right threshold value \cap_{i2}^f of round two. When the second round is over, data collection status α_{status}^{fi} of round three of ε_i is “decrease”. It means that the amount of data collected by ε_i in this round is much greater than *THE*. Therefore, to balance the amount of data, hierarchical private cloud architecture studies automatically to identify A_{sensor} . Before the third round of data collection, according to the regulation mechanism of mode TM, data users input the left threshold value \cup_{i3}^f and the right threshold value \cap_{i3}^f of round three. When the third round of data collection finishes, data collection status α_{status}^{fi} of ε_i in round four is “true”. This shows that the regulation is successful. So far, by the regulation of data amount, the data shortage of round two has been compensated. Thus, hierarchical private cloud architecture reestablishes the balance. It is an *APE* from the end of round two to the success of regulation. At this time, according to the demand, data users can input the left threshold value \cup_{i4}^f and the right threshold value \cap_{i4}^f of the new round. Similarly, the hierarchical private cloud architecture regulates the data amount based on the same principle as the previous *APE*.

The above verification experiments for monitoring faulty sensors in real time can be summarized as follows.

- For mode OM, the number of adjustments exceeds *TE* at the end of the fifth round. But the data amount has not been balanced. After successful identification of F_{sensor} of hierarchical private cloud architecture, maintenance

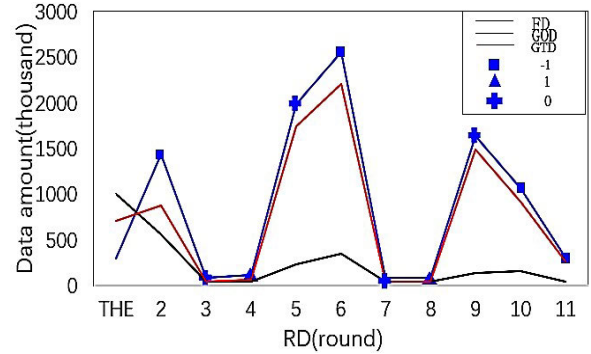


FIGURE 8. Data amount change curve of mode TM.

workers repair it. And then, the architecture regulates the data amount successful soon. It can be seen that the mode OM makes the hierarchical private cloud architecture run slowly and gently, which also makes the staff have a large reaction cycle.

$$TE = \frac{MRT}{\text{Preset total regulation times}} \times 100\% \quad (13)$$

$$VNR = \frac{\alpha_{if}^i}{\beta_{if}^i} \quad (14)$$

- For mode TM, during the balance process, the change curve of data collection amount for θ_2 is closely attached to the change curve of ε_i . This shows that a sensor of θ_2 often performs poorly even though it can meet the balance condition. Also, we can observe that the hierarchical private cloud architecture is very sensitive to F_{sensor} . Once the abnormal sensor A_{sensor} is detected at the beginning, it will be tracked continuously. Thus, TM mode makes the hierarchical private cloud architecture more sensitive and targeted for fault sensor monitoring.
- By comparing the mode OM and the mode TM on fault sensors monitoring, we observe that the mode OM does not confirm the fault sensors until several regulations have been made. Thus, this mode has a long monitoring period. It is slow but reliable. So, mode OM is suitable for the enterprises that are mainly based on batch computing. Instead, mode TM is sensitive to the abnormal sensors which are detected at the beginning continuously. These sensors will be monitored continuously. Thus, this mode has a relatively short monitoring period. It has strong specificity and sensitivity. For this, mode TM is suitable for the enterprises that are mainly based on streaming computing. In summary, both modes have realized real-time monitoring of faulty sensors.

B. EXPERIMENT WITH REAL DATA

In the real data experiment, we apply the high efficiency of mode TM to actual data [35]. In order to verify the effectiveness of hierarchical filtering, we perform the corresponding experiments. The results of experiments have been obtained. Fig. 9 is the comparison chart, which depicts the data amount

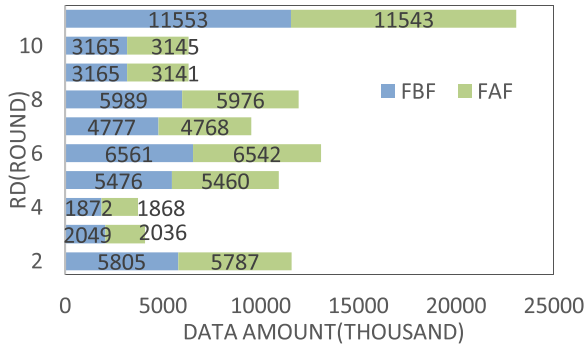


FIGURE 9. Data amount comparison of ε_i that before and after offensive data are filtered.

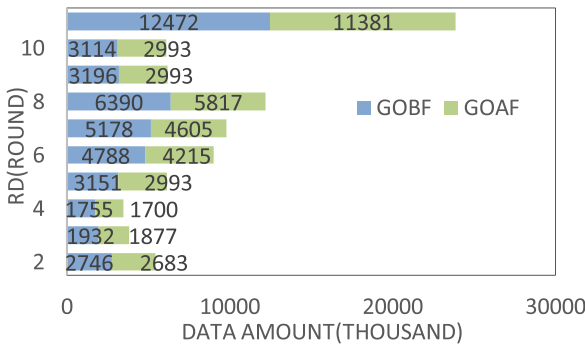


FIGURE 10. Data amount comparison of θ_2 that before and after offensive data are filtered.

of ε_i before and after the offensive data are filtered. It can be observed that ε_i has totally filtered out 146 data. The filtering rate is 0.27%. Furthermore, Fig. 10 is the comparison chart, which displays the data amount of θ_2 before and after the offensive data are filtered. We can see that θ_2 has filtered a total of 3465 data. The filtering rate of offensive data is about 7.75%. To sum up, during this data collection process from sensors to the cloud of the highest layer, the hierarchical private cloud architecture can use filtering mechanism of offensive data of each layer to eliminate offensive data effectively. Therefore, the data security is enhanced in the process of transmission. Also, it is useful for the terminal servers to reduce the risk of being invaded by offensive data. Cloud service providers can provide data services for enterprises by data center more efficiently and safely.

To verify the effectiveness of real-time monitoring of fault sensors, we perform corresponding validation experiments. Correspondingly, we verify and analyze the effectiveness of data amount regulation firstly. The relevant variables are set as follows: 1) *MRT* is set to 3; 2) Preset number of data amount regulation is 9. The results of experiment are described in Fig. 11 which shows the change process of balance regulation of data amount. To begin with, we calculate the qualified data amount *THE* according to (11). Then, hierarchical private cloud architecture begins to carry out the balance regulations. According to Fig. 11, we have succeeded four times during the 9 rounds of balance regulations. That is, an *APE* in this experiment requires 1 to 2 times of regulations.

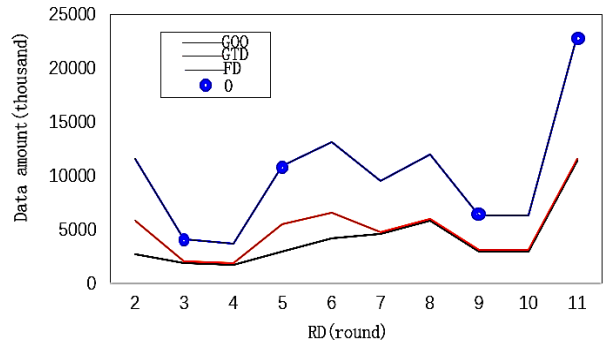


FIGURE 11. System data balance change diagram.

TABLE 3. Data amount balance regulation information.

<i>RD</i>	<i>VNR</i>	α_{status}^{fi}	\cup_{ij}^f	\cap_{ij}^f
2	2.37	-1	1.1	1.2
3	0.83	0	0.0	1.1
4	0.76	1	1.2	1.3
5	2.23	0	1.3	41.3
6	2.68	-1	1.5	1.6
7	1.95	-1	0.0	1.5
8	2.44	-1	0.0	1.5
9	1.29	0	0.0	1.5
10	1.29	1	2.0	2.1
11	4.72	0	2.1	42.1

Therefore, in the real data experiment, the ω of hierarchical private cloud architecture is $\log_{10} 0.5$. It can be easily seen that the architecture achieves real-time balance regulation of data amount when it is tested by real data. From Fig. 11, we also can see that the clouds of production unit layer adaptively recognize abnormal sensors. And, they adaptively regulate the data collection time of the abnormal sensors. Also, the cloud of the factory layer continuously changes the data collection status to regulate the clouds of production unit layer. In summary, clouds at all levels show a strong ability of balance regulation for the data amount.

Moreover, to verify the efficiency for real-time monitoring of fault sensors, we use the obtained actual data to carry out relevant experiments. The results of experiments are shown in Table 3. In this table, the data collection statuses “increase”, “decrease”, and “true” are represented by “1”, “-1”, and “0” respectively. And, the qualified data amount *THE* is calculated by (11) in the first round. Then, before the second round of data collection, according to the demand, data users input the left threshold value \cup_{i2}^f and the right threshold value \cap_{i2}^f of round two freely. When the second

round of data collection is over, the round three's data collection status α_{status}^f of ε_i is "decrease". It means that the amount of data collected by ε_i in the second round is much greater than *THE*. At this time, hierarchical private cloud architecture learns independently to identify abnormal sensor A_{sensor} to balance the data amount. Before the third round of data collection, according to the regulation mechanism of TM mode, data users input the left threshold value \cup_{i3}^f and the right threshold value \cap_{i3}^f of round three. When the third round of data collection ends, the data collection status α_{status}^f of ε_i in the fourth round is "true". Thus, at this time, the regulation is successful. It is an *APE* that from the end of round two to the success of regulation of data amount. Next, according to the demand, the data users can enter the left threshold value \cup_{i4}^f and the right threshold value \cap_{i4}^f for the new round freely. Also, the data amount will be regulated according to the same principle as the previous *APE*. There are 3 rounds of consecutive balance regulations from round 6 to round 8. It shows at this time, the abnormal sensor is likely to be a faulty sensor, which means that the workers need to check and maintain this sensor. After inspection and maintenance, the sensor becomes normal. Then, the hierarchical private cloud architecture becomes balanced soon. In summary, through the above experiments, we can see that the data collected by sensors are used in two directions: 1) Monitor the fault sensors; 2) Provide sample information for data analysis. Therefore, the hierarchical private cloud architecture has good data utilization. Also, we can find that the real-time monitoring mechanism of fault sensors makes us avoid unnecessary maintenance operations, which shortens the time and cost of maintenance. Besides, through the continuous dynamic adjustments and adaptive learning, this architecture can adapt to various faults and can avoid over monitoring, which shows high completeness. Furthermore, the fault sensors are monitored by hierarchical clouds in real time, which shows high real-time and robustness. To sum up, the hierarchical private cloud architecture fully guarantees the security, consistency, and integrity of the data transmission process.

VI. CONCLUSION AND OUTLOOK

A. CONCLUSION

The PTPTM transmits data from collection point to terminal server without regulation mechanisms in the process. It has shortcomings which include two aspects as follows. Firstly, the monitoring of fault sensors is untimely. Secondly, the aggressive data invade the data in transmission. Therefore, this paper proposed the hierarchical private cloud architecture, which can solve these shortcomings by hierarchical transmission of data, hierarchical filtering of offensive data, and real-time monitoring of fault sensors. In this architecture, the real-time monitoring mechanism of fault sensors makes us avoid unnecessary maintenance operations, which shortens the time and cost of maintenance. Besides, to compare with OM mode, the efficiency of identification of fault sensors for TM mode is improved by 2 times. And, TM mode

improves 33.33% acuity of identification. Thus, the TM mode is more suitable for the enterprises that are mainly based on streaming computing. Moreover, each layer can act as a protective screen to counterattack the offensive data, which shows good real-time, robustness, and adaptive ability. In summary, the hierarchical private cloud architecture achieves the filtering of offensive data and the real-time identification of faulty sensors, which guarantees the security, accuracy, and integrity of the data transmission process. It can also help enterprises reduce costs and increase sales. Moreover, the main advantages of the developed method in this paper are the following aspects:

- *Continuous automation improvement and stability.* To compare with other systems, this architecture has the characteristic of achieving automatic continuous self-learning, so that the performance is gradually improved and stable. It can adapt to various faults and avoid over monitoring, which shows high completeness. With the distributed computing and virtualization capabilities of cloud computing, the architecture has the advantages of good computing efficiency and system reliability.
- *Mutual restriction mechanism between layers.* All layers supervise each other and perform their duties according to a certain mode, which reduce the error rate.
- *Multi-level security mechanism.* All layers form their own distinctive aggressive data protection screens through data training. On the basis of perfectly fitting the enterprise production structure, each layer excludes offensive data according to their different abilities.
- *Good data utilization.* The data collected by sensors are used in two directions: 1) Monitor fault sensors; 2) Provide sample information for data analysis.
- *Timeliness.* According to a certain regulation mode, this architecture monitors abnormal sensors in real time by regulating the data collection amount persistently.
- *Reduction of energy waste.* Using machine learning technology and hierarchical regulation ideas, this architecture can avoid unnecessary fault sensors maintenance operations, which shortens the cost of maintenance.

B. OUTLOOK

The authors are expanding the current researches by applying other classification algorithms of machine learning, which aims to improve the balance regulation performance of data amount and find a more convenient method to filter the offensive data. Based on this, we look forward to studying more algorithm mechanisms in cloud service to provide reliable data for customers. Besides, we are also trying to innovate and realize the concepts of "knowledge as a service" and "wisdom as a service".

REFERENCES

- [1] D. Kim, J. Koo, H. Kim, S. Kang, S. H. Lee, and J. T. Kang, "Rapid fault cause identification in surface mount technology processes based on factory-wide data analysis," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 2, Feb. 2019, Art. no. 155014771983280, doi: 10.1177/1550147719832802.

- [2] J. Y. Lee, J. S. Yoon, and B.-H. Kim, "A big data analytics platform for smart factories in small and medium-sized manufacturing enterprises: An empirical case study of a die casting factory," *Int. J. Precis. Eng. Manuf.*, vol. 18, no. 10, pp. 1353–1361, Oct. 2017, doi: [10.1007/s12541-017-0161-x](https://doi.org/10.1007/s12541-017-0161-x).
- [3] Y. Padmanaban and M. Muthukumarasamy, "Scalable grid-based data gathering algorithm for environmental monitoring wireless sensor networks," *IEEE Access*, vol. 8, no. 1, pp. 79357–79367, 2020, doi: [10.1109/ACCESS.2020.2990999](https://doi.org/10.1109/ACCESS.2020.2990999).
- [4] Y. Luo, Y. Duan, W. Li, P. Pace, and G. Fortino, "A novel mobile and hierarchical data transmission architecture for smart factories," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3534–3546, Aug. 2018, doi: [10.1109/TII.2018.2824324](https://doi.org/10.1109/TII.2018.2824324).
- [5] G.-X. Liu, L.-F. Shi, and D.-J. Xin, "Data integrity monitoring method of digital sensors for Internet-of-Things applications," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4575–4584, May 2020, doi: [10.1109/JIOT.2020.2967504](https://doi.org/10.1109/JIOT.2020.2967504).
- [6] V. Garcia-Font, C. Garrigues, and H. Rifà-Pous, "A comparative study of anomaly detection techniques for smart city wireless sensor networks," *Sensors*, vol. 16, no. 6, p. 868, Jun. 2016, doi: [10.3390/s16060868](https://doi.org/10.3390/s16060868).
- [7] S. Pavithra, S. Ranya, and S. Prathibha, "A survey on cloud security issues and blockchain," in *Proc. 3rd Int. Conf. Comput. Commun. Technol. (ICCCCT)*, Feb. 2019, pp. 1–5, doi: [10.1109/ICCCCT.2019.8824891](https://doi.org/10.1109/ICCCCT.2019.8824891).
- [8] C.-J. Su and S.-F. Huang, "Real-time big data analytics for hard disk drive predictive maintenance," *Comput. Electr. Eng.*, vol. 71, no. 1, pp. 93–101, 2018, doi: [10.1016/j.compeleceng.2018.07.025](https://doi.org/10.1016/j.compeleceng.2018.07.025).
- [9] J. Leoni, M. Tanelli, S. C. Strada, and T. Berger-Wolf, "Ethogram-based automatic wild animal monitoring through inertial sensors and GPS data," *Ecological Informat.*, vol. 59, no. 1, Sep. 2020, Art. no. 101112, doi: [10.1016/j.ecoinf.2020.101112](https://doi.org/10.1016/j.ecoinf.2020.101112).
- [10] P. K. Illa and N. Padhi, "Practical guide to smart factory transition using IoT, big data and edge analytics," *IEEE Access*, vol. 6, no. 1, pp. 55162–55170, 2018, doi: [10.1109/ACCESS.2018.2872799](https://doi.org/10.1109/ACCESS.2018.2872799).
- [11] Z. Ullah, S.-T. Lee, and J. Hur, "A novel fault diagnosis technique for IPMSM using voltage angle," *Proc. IEEE Energy Convers. Congr. Expo. (ECCE)*, Sep. 2018, pp. 3236–3243, doi: [10.1109/ECCE.2018.8557375](https://doi.org/10.1109/ECCE.2018.8557375).
- [12] F. Ali, S. El-Sappagh, S. M. R. Islam, A. Ali, M. Attique, M. Imran, and K.-S. Kwak, "An intelligent healthcare monitoring framework using wearable sensors and social networking data," *Future Gener. Comput. Syst.*, vol. 114, no. 1, pp. 23–43, Jan. 2021, doi: [10.1016/j.future.2020.07.047](https://doi.org/10.1016/j.future.2020.07.047).
- [13] R. X. Gao, L. Wang, M. Helu, and R. Teti, "Big data analytics for smart factories of the future," *CIRP Ann.*, vol. 69, no. 2, pp. 668–692, 2020, doi: [10.1016/j.cirp.2020.05.002](https://doi.org/10.1016/j.cirp.2020.05.002).
- [14] J. Havinga, P. K. Mandal, and T. van den Boogaard, "Exploiting data in smart factories: Real-time state estimation and model improvement in metal forming mass production," *Int. J. Mater. Forming*, vol. 13, no. 5, pp. 663–673, Sep. 2020, doi: [10.1007/s12289-019-01495-2](https://doi.org/10.1007/s12289-019-01495-2).
- [15] J. Huang, J. Zhou, Y. Luo, G. Yan, Y. Liu, Y. Shen, Y. Xu, H. Li, L. Yan, G. Zhang, Y. Fu, and H. Duan, "Wrinkle-enabled highly stretchable strain sensors for wide-range health monitoring with a big data cloud platform," *ACS Appl. Mater. Interfaces*, vol. 12, no. 38, pp. 43009–43017, Sep. 2020, doi: [10.1021/acsami.0c11705](https://doi.org/10.1021/acsami.0c11705).
- [16] A. Acemese, C. D. Vecchio, L. Glielmo, G. Fenu, and F. A. Pellegrino, "A combined support vector machine and support vector representation machine method for production control," in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 512–517, doi: [10.23919/ECC.2019.8796111](https://doi.org/10.23919/ECC.2019.8796111).
- [17] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surveys*, vol. 41, no. 3, pp. 1–58, Jul. 2009, doi: [10.1145/1541880.1541882](https://doi.org/10.1145/1541880.1541882).
- [18] M. V. Mahoney and P. K. Chan, "Learning nonstationary models of normal network traffic for detecting novel attacks," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2002, pp. 376–385, doi: [10.1145/775047.775102](https://doi.org/10.1145/775047.775102).
- [19] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly detection in wireless sensor networks: A survey," *J. Netw. Comput. Appl.*, vol. 34, no. 4, pp. 1302–1325, 2011, doi: [10.1016/j.jnca.2011.03.004](https://doi.org/10.1016/j.jnca.2011.03.004).
- [20] 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," *Int. J. Geographical Inf. Sci.*, vol. 26, no. 8, pp. 1373–1392, Aug. 2012, doi: [10.1080/13658816.2012.654493](https://doi.org/10.1080/13658816.2012.654493).
- [21] J. Su, Y. Long, X. Qiu, S. Li, and D. Liu, "Anomaly detection of single sensors using ocsvm_knn," in *Proc. Int. Conf. Big Data Comput. Commun.*, 2015, pp. 217–230, doi: [10.1007/978-3-319-22047-5_18](https://doi.org/10.1007/978-3-319-22047-5_18).
- [22] F. Liu, X. Cheng, and D. Chen, "Insider attacker detection in wireless sensor networks," in *Proc. 26th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, May 2007, pp. 1937–1945, doi: [10.1109/INFCOM.2007.225](https://doi.org/10.1109/INFCOM.2007.225).
- [23] P. Cheng and M. Zhu, "Lightweight anomaly detection for wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 8, Aug. 2015, Art. no. 653232, doi: [10.1155/2015/653232](https://doi.org/10.1155/2015/653232).
- [24] Q. Yu, L. Jibin, and L. Jiang, "An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 1, Jan. 2016, Art. no. 9653230, doi: [10.1155/2016/9653230](https://doi.org/10.1155/2016/9653230).
- [25] C. Désir, S. Bernard, C. Petitjean, and L. Heutte, "One class random forests," *Pattern Recognit.*, vol. 46, no. 12, pp. 3490–3506, Dec. 2013, doi: [10.1016/j.patcog.2013.05.022](https://doi.org/10.1016/j.patcog.2013.05.022).
- [26] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006, doi: [10.1162/neco.2006.18.7.1527](https://doi.org/10.1162/neco.2006.18.7.1527).
- [27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: [10.1145/3065386](https://doi.org/10.1145/3065386).
- [28] R. Socher, C. C.-Y. Lin, A. Y. Ng, and C. D. Manning, "Parsing natural scenes and natural language with recursive neural networks," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 1–9.
- [29] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognit.*, vol. 58, no. 1, pp. 121–134, Oct. 2016, doi: [10.1016/j.patcog.2016.03.028](https://doi.org/10.1016/j.patcog.2016.03.028).
- [30] P. S. S. de Souza, F. P. Rubin, R. Hohemberger, T. C. Ferreto, A. F. Lorenzon, M. C. Luizelli, and F. D. Rossi, "Detecting abnormal sensors via machine learning: An IoT farming WSN-based architecture case study," *Measurement*, vol. 164, no. 1, Nov. 2020, Art. no. 108042, doi: [10.1016/j.measurement.2020.108042](https://doi.org/10.1016/j.measurement.2020.108042).
- [31] Z. Wang and L. Chiang, "Monitoring chemical processes using judicious fusion of multi-rate sensor data," *Sensors*, vol. 19, no. 10, p. 2240, May 2019, doi: [10.3390/s19102240](https://doi.org/10.3390/s19102240).
- [32] S. Wang, J. Wan, D. Li, and C. Liu, "Knowledge reasoning with semantic data for real-time data processing in smart factory," *Sensors*, vol. 18, no. 2, p. 471, Feb. 2018, doi: [10.3390/s18020471](https://doi.org/10.3390/s18020471).
- [33] R. Chamarajagar and A. Ashok, "Integrity threat identification for distributed IoT in precision agriculture," in *Proc. 16th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2019, pp. 1–9, doi: [10.1109/SAHCN.2019.8824841](https://doi.org/10.1109/SAHCN.2019.8824841).
- [34] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganec, and A. V. Vasilakos, "Fog computing for sustainable smart cities: A survey," *ACM Comput. Surveys*, vol. 50, no. 3, pp. 1–43, Oct. 2017, doi: [10.1145/3057266](https://doi.org/10.1145/3057266).
- [35] A. I. Arafat, T. Akter, M. F. Ahammed, M. Y. Ali, and A.-A. Nahid, "A dataset for Internet of Things based fish farm monitoring and notification system," *Data Brief*, vol. 33, no. 1, Dec. 2020, Art. no. 106457, doi: [10.1016/j.dib.2020.106457](https://doi.org/10.1016/j.dib.2020.106457).



CHAO-HSIEN HSIEH received the B.S. degree in industrial engineering from Yuan Ze University, Taoyuan City, Taiwan, in 1996, the M.S. degree in computer science from Oklahoma City University, Oklahoma City, in 2001, and the Ph.D. degree from the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, in 2015. After many years at research and development working experiences, he received the Ph.D. degree. He is currently an Associate Professor with the School of Cyber Science and Engineering, Qufu Normal University, Qufu, Shandong, China. His research interests include cloud computing, blockchain, and database.



ZIYI WANG is currently pursuing the bachelor's degree with the School of Cyber Science and Engineering, Qufu Normal University, Qufu, Shandong, China. Her research interests include cloud computing, machine learning, and intelligent optimization algorithm.

...