# Anomaly Detection Method of Healthcare Internet of Things Gateway Supporting Edge Computing

Zixiu Zou[1,2], Yi Hu[3], Xinyao Liu[4], and Shufeng Zhuo[5(✉)]

[1] Fuzhou Institute of Technology, Fuzhou 350506, Fujian, China
[2] SEGI University, 47810 Petaling Jaya, Selangor, Malaysia
[3] Henan Information Engineering School, Zhengzhou 450000, China
[4] Imperial Vision Technology Company Limited, Fuzhou 350001, China
[5] The Internet of Things and Artificial Intelligence College, Fujian Polytechnic of Information Technology, Fuzhou 350003, China
55607@qq.com

**Abstract.** As the link between the perception layer and the network layer, the Internet of Things gateway is of great significance to the safe and stable operation of the healthcare Internet of Things. Once the gateway is abnormal, it will directly affect the information transmission in health care work. Therefore, an anomaly detection method for the gateway of the Internet of Things in health care supporting edge computing is proposed. Several representative gateway status indicators are selected by using the maximum uncorrelation method, and the gateway anomaly detection task is unloaded to the edge server by using edge computing. An anomaly detection model based on SOFM neural network and random forest is constructed to realize the anomaly detection of the Internet of Things gateway in health care. The experimental results show that the determination coefficients of the six types of samples of this method are more than 0.9, which is close to 1, which shows that this method has better anomaly detection performance of the Internet of Things gateway in health care.

**Keywords:** Edge Calculation · Medical Care · Internet Of Things · Gateway Is Abnormal · Test Method

## 1 Introduction

What is the Internet of Things? The notes attached to the 2010 Chinese government work report explained: "The Internet of Things refers to the use of information sensing devices (radio frequency identification RFID, infrared sensors, global positioning systems, laser scanners, etc.). According to the agreed protocol, any item is connected to the Internet for information exchange and communication, so as to realize intelligent identification, positioning, tracking, monitoring and management. It is a network that extends and expands on the basis of the Internet." This explanation may appear in the medical and


S. Wang (Ed.): IoTCare 2022, LNICST 501, pp. 239–254, 2023.
https://doi.org/10.1007/978-3-031-33545-7_17

health field in the future: the mobile phone, watch or belt we carry with us suddenly sends out a signal to remind our health problems; this signal can be sent to the hospital, and if the situation is urgent, the ambulance will go directly. to your place; If it is not too serious, community doctors can call and directly view medical files, conduct remote consultations, and make appointments for medical treatment; They can even deliver medicines to homes according to prescriptions through pharmaceutical logistics. The Internet of Things can play an important role in the application of "barcode" patient identity management, mobile medical orders, electronic entry of symptoms and signs, mobile drug management, mobile test specimen management, mobile medical record management, data storage and transfer, infant theft prevention, nursing process, clinical pathway and other management in the medical and health field [1, 2].

With the rapid development of the Internet of Things in the healthcare industry, the gateway of the Internet of Things, which connects the sensing network and the traditional communication network, is playing an important role. As the manager of the Internet of Things, the gateway controls the operation of the whole Internet of Things, and its management authority and reliability requirements are the highest. Because the Internet of Things usually works in a complex environment, and the gateway is the most important field equipment of the Internet of Things, some inevitable important or urgent problems will inevitably appear in its work. Many factors will make it difficult for the gateway to work normally all the time [3]. Once the gateway is abnormal, the whole network will be paralyzed. All sensor nodes managed by the gateway listen to network messages for a long time, resend data repeatedly, channel congestion, data collision and other phenomena, which affect the effectiveness of medical care tasks. In response to the above situation, anomaly detection of healthcare Internet of Things gateways has attracted great attention from both academia and engineering. However, due to the huge amount of gateway status information, and each piece of status information contains a large number of attributes, it is extremely difficult to label each piece of information; with the continuous development of network applications, the amount of data will increase exponentially, and the central the system appears powerless. Faced with the above situation, a method for anomaly detection of healthcare Internet of Things gateways supporting edge computing is studied. The overall structure of the method is as follows:

(1) The maximum uncorrelation method is used to select several representative gateway status indicators, and edge computing is used to unload the gateway anomaly detection task to the edge server. An anomaly detection model based on SOFM neural network and random forest is constructed to realize anomaly detection of healthcare IoT gateway.
(2) During the experiment, the gateway status indicators are set, and the IoT gateway anomaly detection experiment is carried out through sample preparation and task unloading scheme determination. The anomaly detection structure is obtained, and the performance of this method is verified.
(3) Summarize the full text, analyze the limitations of the anomaly detection method of the healthcare IoT gateway that supports edge computing, and further explain the future work.

## 2 Research on Anomaly Detection of Internet of Things Gateways Based on Edge Computing

The ultimate goal of the Internet of Things is to realize the interconnection of all things in the world and the barrier free information exchange between people, people and things, and things and things. As an Internet interconnection device in the Internet of Things, the gateway of the Internet of Things plays a connecting role, realizing the protocol conversion and data interaction between the sensing network and the communication network. Gateway is also known as inter network connector and protocol converter. The default gateway realizes network interconnection at the network layer. It is the most complex network interconnection device and is only used for the interconnection of two networks with different high-level protocols. The structure of the gateway is similar to that of the router, except for the interconnection layer. Gateway can be used for both wide area network interconnection and LAN interconnection. To go from one room to another, one must pass through a door. Similarly, sending information from one network to another must also pass through a "gateway", which is the gateway. As the name suggests, a gateway is a "gateway" that connects a network to another network, that is, a network gate. Once there is a problem with the gateway, it will directly affect the communication quality of the entire Internet of Things, so it is necessary to perform accurate status detection on the gateway.

### 2.1 Determination of Gateway Status Indicators

There are many indicators that can reflect the status of the gateway. In the past, one or two indicators were selected for anomaly detection, which has great limitations, making the accuracy of anomaly detection not high. In the face of this situation, the maximum uncorrelation method is used to select several representative gateway status indicators.

If indicator $s_1$ and other indicators $s_2, s_3, \ldots, s_m$ are independent, it means that $s_1$ cannot be replaced by other indicators, so the reserved indicators should be as small as possible. Under the guidance of this method, a method of maximum irrelevance is derived. The maximum irrelevance method mainly selects indicators according to the relationship between the complex correlation coefficient and the set critical value, where the complex correlation coefficient refers to the degree of correlation between an indicator and other indicators [4]. The basic principle is as follows: Firstly, the correlation coefficient matrix $Y$ of the sample is obtained, and then according to the correlation coefficient $y_{ij}$, the complex correlation coefficient $z_i^2$ is obtained (the complex correlation coefficient refers to the degree of correlation between one indicator and other indicators), and the critical value $T$ is defined. Judging the relationship between the complex correlation coefficient $z_i^2$ and the critical value $T$, if $z_i^2 > T$, the index can be removed. The specific calculation process is as follows:

Step 1: Find the correlation coefficient matrix $Y$ of the sample, such as:

$$Y = \{y_{ij}\}_{mm} = \begin{Bmatrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ \cdots & & & \\ y_{m1} & y_{m2} & \cdots & y_{mm} \end{Bmatrix} \tag{1}$$

$$y_{ij} = \frac{s_{ij}^2}{\sqrt{\frac{s_{ii}^2 s_{jj}^2}{m}}}, i, j = 1, 2, \ldots, m \tag{2}$$

where, $y_{ij}$ reflects the linear correlation between $s_i$ and $s_j$.

Step 2: Calculate the complex correlation coefficient $z_i^2$ according to $y_{ij}$. The value of $z_i$ can be calculated by $Y$. The specific steps are as follows: $Y$ is divided into blocks in the following way, and $Y$ blocks can be expressed as the following formula:

$$Y = \begin{bmatrix} Y_{-m}\ y_m \\ y_m^T\ 1 \end{bmatrix} \tag{3}$$

At this time, the main diagonal element of $Y$ is 1, so the complex correlation coefficient of each index can be calculated according to formula $z_i^2 = y_m^T Y_{-m} y_m$, and $z_1^2, z_2^2, \ldots, z_m^2$ can be obtained.

Step 3: Determine the relationship between the critical values $T$ and $z_i^2$. First determine the critical value $T$. The critical value $T$ is generally the F test of $z_i^2$. If $\alpha = 0.1$ is taken, if $F > F_{0.10}$, it means that the multiple correlation is significant. Finally, judge the relationship between $T$ and $z_i^2$. If $z_i^2 > T$, delete the index.

Finally, due to different dimensions, the indicators cannot be put together for comparison and analysis, so it is necessary to standardize each indicator, and the calculation formula is as follows:

$$\hat{s}_{ij} = \frac{s_{ij} - (\bar{s}_j)}{\sqrt{\mathrm{var}(s_j)}} \tag{4}$$

In the formula, $\hat{s}_{ij}$ represents the standardized index; $\bar{s}_j$ represents the mean of the $j$ index; $\mathrm{var}(s_j)$ represents the mean square error of the $j$ index; $s_j$ represents the $j$ index.

### 2.2 Anomaly Detection Task Offloading Based on Edge Computing

In the face of a large number of anomaly detection tasks of medical and health Internet of Things gateways with multiple indicators, the traditional central system has been unable to meet the needs of rapid data processing, and edge computing can effectively solve this problem by establishing nodes near the data source to reduce the data transmission delay and divert the tasks of the computing center. Edge computing has the characteristics of real-time, high bandwidth, heterogeneity, etc. by extending the computing power closer to the end user, it makes up for the shortcomings of cloud computing and is the optimization and expansion of cloud computing [5]. With the advent and development of the Internet of Things, edge computing, as a developing computing paradigm, is considered to be one of the key architectures of the next generation communication network. Edge computing architecture is to add edge servers between terminal devices and cloud servers to expand services at the edge of the network. The system architecture of edge computing is generally divided into terminal layer, edge layer and cloud layer.

(1) The terminal device layer, including various mobile terminals connected to the edge network and many Internet of Things devices, such as smartphones, various sensors,

and smart cars. At the terminal layer, the device is not only a data consumer, but also a data provider. In order to reduce the terminal service delay, only the perceptual capabilities of various terminal devices are considered, and the computing capabilities are not considered. Therefore, several devices at the terminal layer collect various raw data and transmit them to the upper-layer architecture, where data tasks are stored and calculated.

(2) The edge layer is the core of the three-tier architecture. It is located at the edge of the network and consists of edge nodes widely distributed between terminal devices and the cloud. It usually includes base stations, access points, routers, switches and edge gateways. The edge layer supports the terminal to access downward, store and calculate the data uploaded by the terminal, connect upward with the cloud, and upload the processed data to the cloud. Because the edge layer is close to users, data transmission to the edge layer is more suitable for real-time data analysis and intelligent processing, which is more efficient and secure than cloud computing. Among the joint services of edge cloud computing, cloud computing server is still a very powerful data processing center.

(3) The cloud layer, composed of multiple high-performance servers and storage devices, has powerful computing and storage capabilities, and can play an important role in areas where there are many data analysis services such as regular maintenance and business decision support. The cloud computing center has the function of storing the data uploaded by the edge computing layer for a long time. In addition, analysis tasks that cannot be processed by the edge layer or other heavy computing tasks can also be implemented in the cloud. The cloud module can also dynamically adjust the edge computing layer according to the control strategy. Deployment strategy.

The edge computing system architecture was proposed in the Edge Computing White Paper 3.0 released in December 2018. The edge computing reference architecture presents the architectural content from different perspectives in a multi-view manner, as shown in Fig. 1, and displays each layer through multiple perspectives function.

The bottom device layer of the framework connects the whole framework, including management services, data lifecycle services and security services. Management services provide unified management and monitoring, and provide information to the management platform. Data lifecycle services provide integrated management for the preprocessing, analysis and execution of machine data. Security services can flexibly deploy and optimize data services to meet the real-time needs of business.

Computing offload is performed by migrating the computing tasks on the terminal device to the extended cloud or edge platform. The computing platform assists the terminal to complete the user's task request, and returns the computing results to the specified device. The task unloading technology mainly includes two problems: unloading decision and resource allocation. The main research points of unloading decision are whether to unload the task, the unloading destination and the unloading amount of the task. The main research points of resource allocation are how much communication and computing resources the server needs to allocate to the task. Generally speaking, there are a series of application tasks with different amounts of data that need to be executed on the terminal device. Firstly, the device needs to detect whether there is a server in

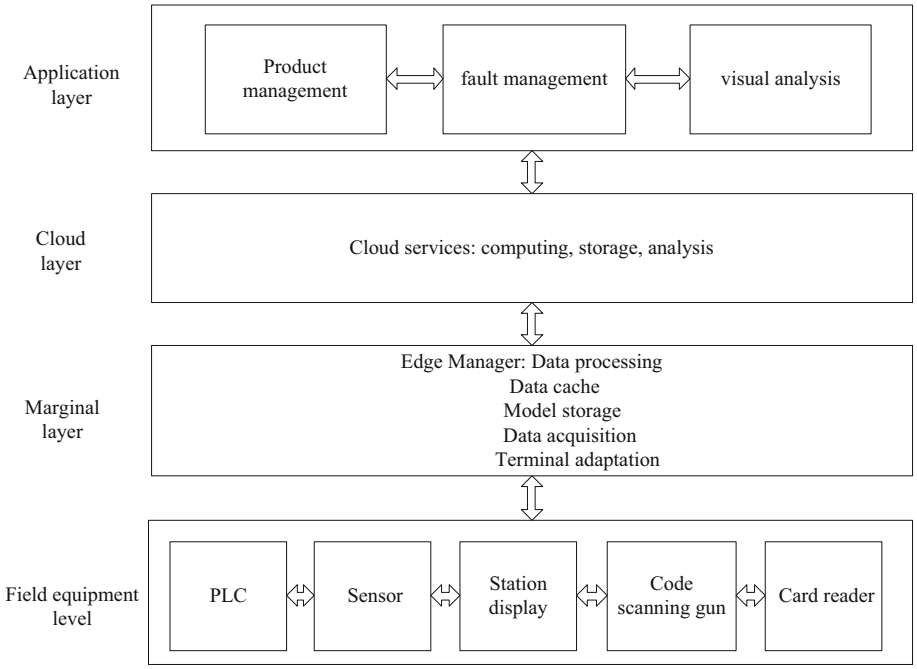


**Fig. 1.** Edge computing system architecture

the environment that can perform the uninstallation operation. Then, considering the partial uninstallation and binary uninstallation of the task, determine whether the task can be uninstalled, how to uninstall it, and when to uninstall it. The task on the terminal device can be executed locally or on the edge server. If the resources on the edge side are insufficient or cannot meet the demand, the task can be further sent to the cloud server. For local execution, only the computing power of the device itself needs to be considered and the calculation results are output. For remote execution, submit the computing task to the upper-layer server through Wi-Fi and other methods, so that the server will allocate computing resources, communication resources and storage resources for the received task, and execute the computing task. The task result is returned to the user and the occupied resources are released.

After completing task clustering and server resource integration, we need to consider the problem of system resource allocation, that is, which server the task needs to be allocated to perform the most reasonable. Because the communication and computing resources of mobile edge computing server are relatively limited, and the terminal devices are usually heterogeneous, considering the dynamic nature of task unloading and resource load, the optimization goal of most current research work is to comprehensively consider the measurement of delay and load balance when the task does not exceed the bandwidth, storage and computing capacity of the server [6].

**Load Balancing**

In mobile edge computing, there are many load factors that affect task offloading and resource allocation. We assume that $A_i$ represents the CPU utilization of the $i$ server, $B_i$ represents the memory utilization of the $i$ server, and $C_i$ represents the bandwidth utilization of the $i$ server. Assuming that each server can normally receive requests from resource requesters, the load of each server is defined as follows:

$$D_i = w_1 A_i + w_2 B_i + w_3 C_i \tag{5}$$

Calculate the average load of all servers. The formula is as follows:

$$\overline{G} = \frac{\sum_{i=1}^{N} D_i}{N} \tag{6}$$

Among them, $w_1$, $w_2$, $w_3$ represent the weight coefficient; $\overline{G}$ represents the average load; $N$ represents the number of servers.

Therefore, the load balance degree is expressed as formula (7). The more average the value of the load balance degree is, the more balanced the load distribution of the entire edge computing network system is, and the more average the task distribution is.

$$H = \sqrt{\frac{\sum_{i=1}^{N} \left(D_i - \overline{G}\right)}{N}} \tag{7}$$

In the formula, $H$ represents the load balance degree.

**Time Delay**

The processing speed of the server for different exception detection tasks is different, and the execution time will affect the user's quality of service. Therefore, the main optimization goal is to effectively reduce the completion time of the task. Assuming that $P_j$ represents the amount of data of exception detection task $j$, the server's execution time is defined as:

$$Q_{ij} = \frac{P_j}{A_i} \tag{8}$$

In the formula, $Q_{ij}$ represents the time when the $i$ server executes the anomaly detection task $j$.

Since the resource requester has requested multiple anomaly detection tasks, the calculation time of the task execution required for all mobile devices to complete the task is expressed as follows:

$$U = \sum_{i=1}^{N} \sum_{j=1}^{M} Q_{ij} \tag{9}$$

where, $U$ represents the total execution time of anomaly detection task; $M$ represents the number of tasks.

Based on the above description, the task unloading problem in the mobile edge computing scenario is usually NP problem, which is difficult to solve directly. At present, most of the research to solve this kind of problem is to consider the use of intelligent colony algorithm. The intelligent colony algorithm used here is the fireworks algorithm [7]. Firstly, in the whole feasible solution space, an indefinite number of initial fireworks populations are randomly generated, each fireworks is equivalent to a feasible solution, and then the corresponding fitness function value of each fireworks is determined. By generating different fireworks explosion radius, the fireworks set is updated, and the next generation of explosion sparks and Gaussian mutation sparks are generated. Whether the algorithm meets the cycle end condition is judged, and if so, the search is stopped. Otherwise, select a certain number of individuals in the candidate set as a new fireworks population to enter the iteration of the next process. From the above process, we can see that the fireworks algorithm has an adaptive radius adjustment mechanism, and has certain exploration and mining capabilities. The adaptation of anomaly detection task unloading and fireworks algorithm parameters is shown in Table 1 below.

**Table 1.** Anomaly detection task offloading problem and parameter adaptation of fireworks algorithm

| Fireworks algorithm | Uninstall problems |
|---|---|
| Individual dimension | Number of user tasks |
| Individual | Single unloading scheme |
| Population | Collection of different unloading schemes |
| Fireworks location | Unloading schemes for different user tasks |
| Fitness value | Combined value of load balancing and task delay |

The specific process is as follows:

Step 1: Initialize. Determine the fireworks individual dimension $M$ (number of anomaly detection tasks) and the value range $N$ (number of servers) of each dimension, as well as the number of fireworks in each generation and other algorithm parameters. Through the reverse learning strategy, a certain number of fireworks are generated as the initial fireworks population.

Step 2: Calculate the fitness value. The location information of fireworks is the unloading decision variable. Under the current unloading decision, the resource allocation variable is solved by convex optimization formula (7), and then the total delay cost value of the task is obtained. As the fitness value of this fireworks individual, the fitness value of all individuals is solved.

Step 3: Solve the explosion radius $r$ and the explosion number $k$.

Step 4: Generate offspring fireworks. Within the explosion radius $r$, $k$-number fireworks are generated by random strategy as offspring fireworks.

Step 5: Generate mutation sparks. Through the mutation operator, the mutant firework individual is generated.

Step 6: Select the next generation of fireworks individuals and update the optimal value of the population. The optimal fireworks are selected among the parent fireworks, offspring fireworks and mutant fireworks to enter the next generation, and other fireworks are selected as the next generation fireworks through the championship strategy, and the optimal value of the population is updated.

Step 7: Judge whether the termination conditions are met. That is, if the maximum number of iterations is reached, the cycle will exit, and the minimum value of the total delay cost of the task and the corresponding unloading decision and resource allocation scheme will be searched. Otherwise, repeat steps 2 to 6 until the termination conditions are met.

Step 8: Output the optimal solution, that is, the anomaly detection task unloading scheme that can meet both the load balancing requirements and the delay requirements.

### 2.3 Gateway Anomaly Detection Based on Deep Learning

After the above exception detection task is uninstalled, the exception detection task is executed on each edge server. The detection algorithm used here is a combination of SOFM neural network and random forest algorithm. Clustering is carried out through SOFM neural network, and different clusters are divided. The samples in the cluster are sampled for data balancing. Finally, the balanced data set is trained with random forest algorithm to obtain multiple classifiers [8]. Each edge server has a classifier based on random forest for edge operation, and the prediction category corresponding to the maximum value of the calculated comprehensive weight is the final detection result.

**SOFM Neural Network**

SOFM neural network, the full name of self-organizing feature mapping neural network, is a neural network model proposed by Finnish scholar Kohonen study [9]. The SOFM network has two layers, namely the input layer and the output layer. The output layer is usually composed of a one-dimensional or two-dimensional network matrix. In actual use, the SOFM neural network will learn to map the input data to the corresponding position of the output layer over a period of time. The neuron areas activated by different input data are also different. If the data structure is similar, the adjacent areas will be activated. The difference in the activation area will reasonably distinguish the input data. This process achieves the purpose of learning by changing the connection weights, and these processes are done automatically internally, so this method is called self-organizing feature mapping.

The learning process of SOFM neural network is to input training samples in the input layer. The network can self-organizing adjust the weight vector between neurons, so that the weight vector changes with the change of input mode, and finally make the neurons in the output layer more sensitive to input data. All connection weight vectors are separated from each other to form a set that can represent the input mode, so as to achieve the effect of self-organizing clustering [10].

The gateway status indicator sample has a relatively obvious feature in the data structure, that is, the number of abnormal samples is much smaller than the number of

normal samples. Generally speaking, when the ratio of positive and negative samples is greater than 4:1, if these data sets are directly classified and trained, then The prediction results of the trained classification model in actual use will be heavily biased towards the sample category with a high proportion, resulting in poor performance of the algorithm model. In this regard, the state index data of unbalanced gateway is processed based on SOFM neural network.

For an unbalanced data set, find the $g$ nearest neighbors of each minority sample $v_i$, and select $h$ samples from the $g$ nearest neighbors of the minority sample (ensure that $g > h$). These $h$ samples are assumed to be $v_1, v_2, \ldots, v_h$, and then interpolate through the interpolation formula (see Eq. 10) to get new samples. In this way, it is equivalent to adding $h$ new samples to each minority sample.

$$\hat{v} = v_i + f(0,1)\big(u_j - v_i\big), j = 1, 2, \ldots, h \tag{10}$$

Among them, $\hat{v}$ represents the new sample; $v_i$ represents the sample point of the minority category, $u_j$ represents the sample point selected from the $g$ nearest neighbors of $v$; $f(0, 1)$ represents the number between (0, 1) randomly generated.

**Random Forest Algorithm**

Random forest algorithm is actually an improved version of bagging algorithm, which is equivalent to introducing random attribute selection into bagging algorithm based on decision tree. Random forest algorithm is easy to implement in practical application, with good effect and low time cost, which benefits from the two random measures of random forest, that is, based on the random sample selection of bagging algorithm itself and the random attribute selection, the generalization performance of random forest algorithm is further improved [11].

In the iterative process of the random forest algorithm, the self-service sampling method is used for the selection of samples. By sampling the sample set with replacement, the samples drawn each time are put into the inBag. In addition, according to the content of Chapter 2, there are probably 37% of the samples will not be drawn, put them into the outBag. The data in the inBag is trained by the decision tree algorithm to generate a classification model, and then the model effect is tested through the given test set. The evaluation indicators of the model effect generally use the precision rate $\eta$, the recall rate $\mu$, and the F1 value. From the formula (11), the F1 value integrates the precision rate $\eta$ and the recall rate $\mu$, which can measure and evaluate the performance of the model more objectively. The confusion matrix is shown in Table 2.

$$F1 = \frac{2 \cdot \eta \cdot \mu}{\eta + \mu} \tag{11}$$

Among them,

$$\eta = \frac{T_1}{T_1 + T_3} \tag{12}$$

$$\mu = \frac{T_1}{T_1 + T_2} \tag{13}$$

**Table 2.** Confusion matrix

| Project | | Actual results | |
|---|---|---|---|
| | | Positive class | Negative class |
| Prediction results | Positive class | T1 | T2 |
| | Negative class | T3 | T4 |

In the formula, F1 represents the harmonic mean between precision $\eta$ and recall $\mu$.

Assuming that the number of base classifiers in the random forest is $\beta$, the F1 value of each decision tree calculated by the given test set is $\psi_1, \psi_2, \ldots, \psi_\beta$, respectively, and the weights of the decision trees in the random forest are as follows:

$$\lambda_i = \frac{\psi_i}{\sum_{i=1}^{\beta} \psi_i} \tag{14}$$

After giving weight to each decision tree, predict and judge the new sample $\hat{v}$ through these decision trees. Assuming that the probability of each decision tree predicting that the new sample $\hat{v}$ belongs to a certain category $\varepsilon$ is $\xi_a(\hat{v})$, then in the final voting decision, combined with the weight of the decision tree and the prediction probability, the comprehensive weight of sample $\hat{v}$ belonging to category $\varepsilon$ can be obtained as follows [12]:

$$\varpi_\varepsilon(\hat{v}) = \sum_{i=1}^{\beta} \psi_i \times \xi_a(\hat{v}) \tag{15}$$

In the formula, $\varpi_\varepsilon(\hat{v})$ sample $\hat{v}$ belongs to the comprehensive weight of category $\varepsilon$.

Combining the above content, the detailed idea of the classifier can be obtained: in the random forest algorithm [13, 14], multiple sample sets are extracted by the self-help method, and each sample set is trained to obtain a decision tree mode, and then a given test sample set is used to pair the the performance of multiple decision tree models is evaluated, and the evaluation index is the F1 value. According to the principle that the larger the F1 value, the better the model effect, the ratio of the performance index F1 value of each decision tree to the sum of the F1 values of all decision trees is used to determine each decision tree. The weight of the decision tree, then for the prediction of the new sample, first use each decision tree model to judge it, predict the probability that the sample belongs to a certain class, multiply and sum all the decision tree weights and the predicted probability, and get a certain class Finally, the maximum comprehensive weight is calculated by comparison, and the prediction category corresponding to this value is the final result predicted by the random forest algorithm [15].

# 3 Detection Method Application Test

## 3.1 Gateway Status Indicators

There are 9 gateway status indicators selected based on the maximum uncorrelation method, as follows:

(1) Throughput: The packet forwarding capability of the device. It usually refers to the ability of the tested equipment to forward data without packet loss, which is generally expressed as a percentage of the line speed (or passing rate) that can be achieved.
(2) Latency: The time interval between receiving packets and forwarding packets within the throughput range of the device.
(3) Packet loss rate: The ratio of the number of discarded packets to the number of received packets under different loads.
(4) Back to back frame: The maximum number of packets that the device can process without packet loss when it receives transmission at the minimum packet interval.
(5) System recovery: The time for the equipment to resume normal operation after overload. If the network equipment has line speed capability, the test is meaningless.
(6) System reset: The time interval from software reset or power off restart to normal operation of the device. Normal operation refers to the ability to forward data with throughput.
(7) Maximum number of concurrent connections: the maximum number of connections that can be established simultaneously between hosts passing through the conversion gateway or between the host and the conversion gateway.
(8) Breakpoint resume transmission capability: In the event of network interruption, the gateway continues to accurately collect data, cache the data in non-volatile devices, and retransmit the cached data to the industrial cloud platform through the forwarding channel when the network returns to normal. Ability.
(9) Local alarm capability: In the case of network failure, equipment failure, etc., the gateway should provide local alarm information based on configuration information and real-time data, and support multiple alarm types such as switch value and analog value. And when sending monitoring data packets to the device, when the return speed reaches a certain threshold, the device will give an early warning to remind the operator to pay attention.

## 3.2 Sample Preparation

Based on the gateway status index, training samples and test samples are generated with the help of MATLAB, and the results are shown in Fig. 2 below.

It can be seen from Fig. 2 that there are 6 types of samples, and each type of sample includes several small samples.

## 3.3 Task Offloading Scheme

The sample detection tasks in Fig. 2 are evenly distributed to 15 servers, and the number of tasks allocated to each server is shown in Table 3 below.

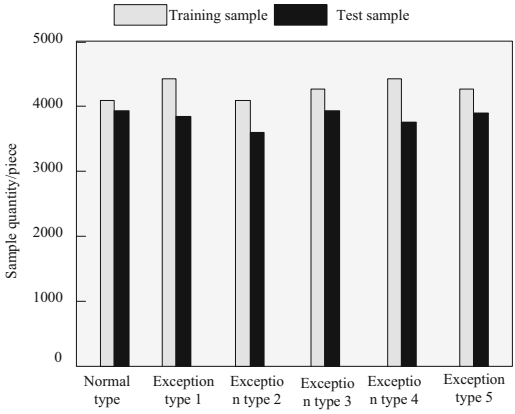


**Fig. 2.** Sample distribution

**Table 3.** Gateway anomaly detection task offloading scheme

| The server | Training tasks | | Test task | |
|---|---|---|---|---|
| | Normal sample | Abnormal sample | Normal sample | Abnormal sample |
| 1 | 464 | 355 | 125 | 156 |
| 2 | 315 | 257 | 144 | 124 |
| 3 | 235 | 220 | 123 | 144 |
| 4 | 235 | 325 | 125 | 133 |
| 5 | 185 | 56 | 120 | 125 |
| 6 | 174 | 87 | 122 | 158 |
| 7 | 152 | 145 | 322 | 187 |
| 8 | 263 | 252 | 214 | 183 |
| 9 | 241 | 145 | 210 | 32 |
| 10 | 187 | 102 | 54 | 55 |
| 11 | 166 | 135 | 87 | 289 |
| 12 | 254 | 66 | 565 | 188 |
| 13 | 146 | 283 | 222 | 176 |
| 14 | 225 | 36 | 345 | 133 |
| 15 | 255 | 255 | 146 | 165 |

### 3.4 Anomaly Detection Results

Use the training samples in Fig. 2 to train the anomaly detection model based on SOFM neural network and random forest, and then input the test samples into the trained model. Some results are shown in Table 4 below.

**Table 4.** Example of anomaly detection results

| Small sample | Type | Comprehensive weight | Detection category |
|---|---|---|---|
| 1 | Normal type | 6.25 | Exception type 5 |
| | Exception type 1 | 4.12 | |
| | Exception type 2 | 3.87 | |
| | Exception type 3 | 5.55 | |
| | Exception type 4 | 5.47 | |
| | Exception type 5 | 7.45 | |
| 2 | Normal type | 2.58 | Exception type 5 |
| | Exception type 1 | 2.47 | |
| | Exception type 2 | 3.68 | |
| | Exception type 3 | 5.36 | |
| | Exception type 4 | 5.47 | |
| | Exception type 5 | 2.58 | |
| 3 | Normal type | 7.45 | Normal type |
| | Exception type 1 | 5.32 | |
| | Exception type 2 | 3.65 | |
| | Exception type 3 | 3.22 | |
| | Exception type 4 | 3.20 | |
| | Exception type 5 | 1.77 | |

### 3.5 Method Performance Test

All abnormal detection results are counted, and then the coefficient of determination, also known as goodness of fit, is calculated, which can effectively reflect the difference between the detection results and the actual results. The closer the coefficient of determination is to 1, the closer the detection result is to the reality, and the result is shown in Fig. 3 below.

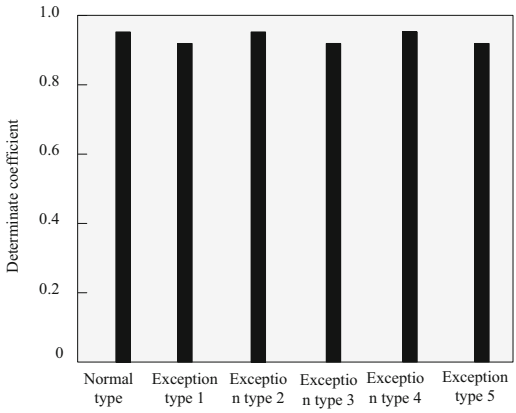


**Fig. 3.** Method performance test results

It can be seen in Fig. 3 that under the application of the studied method, the determinable coefficients of the six types of samples are more than 0.9, which is close to 1, which illustrates the performance of the studied anomaly detection method.

## 4 Conclusion

(1) To sum up, once the gateway is abnormal, the entire Internet of Things will be paralyzed. Faced with this situation, this paper proposes an anomaly detection method for healthcare IoT gateway that supports edge computing.
(2) This method assigns the anomaly detection task to the edge server, and implements gateway anomaly detection through the detection model on the server and the classifier. Finally, through testing, the determination coefficients of the test results are all above 0.9, which proves the effectiveness of the method.
(3) However, the SOFM neural network used in this method has some defects, such as slow learning convergence speed, and the network performance is sensitive to initial conditions. However, this paper does not propose a solution to overcome the defects of SOFM neural network, so there are some limitations. In the future, this problem needs to be studied in depth to optimize the comprehensive performance of SOFM neural network, so as to maximize the quality of anomaly detection of healthcare IoT gateway.

## References


1. Wang, W., Zhang, X., Wang, S.-H., Zhang, Y.-D.: Covid-19 diagnosis by WE-SAJ. Syst. Sci. Control Eng. **10**(1), 325–335 (2022). https://doi.org/10.1080/21642583.2022.2045645
2. Huang, C., Wang, W., Zhang, X., Wang, S.-H., Zhang, Y.-D.: Tuberculosis diagnosis using deep transferred EfficientNet. IEEE/ACM Trans. Comput. Biol. Bioinf. (2022). https://doi.org/10.1109/TCBB.2022.3199572
3. Nikseresht, M., Mollamotalebi, M.: Providing a CoAP-based technique to get wireless sensor data via IoT gateway. Comput. Commun. **172**(2), 155–168 (2021)
4. Peng, C., Chen, J., Vijayakumar, P., et al.: Efficient distributed decryption scheme for IoT gateway-based applications. ACM Trans. Internet Technol. **21**(1), 1–23 (2021)
5. Cui, E., Yang, D., Wang, H., et al.: Learning-based deep neural network inference task offloading in multi-device and multi-server collaborative edge computing. Trans. Emerging Telecommun. Technol. **33**(7), 4485–4505 (2022)
6. Huang, L., Ran, J., Wang, W., et al.: A multi-channel anomaly detection method with feature selection and multi-scale analysis. Comput. Netw. **185**(8), 107645 (2020)
7. Song, L., Zheng, T., Wang, J., et al.: An improvement growing neural gas method for online anomaly detection of aerospace payloads. Soft. Comput. **24**(6), 1–13 (2020)
8. Jiang, Y., Yu, Y., Peng, X.: Online anomaly detection in DC/DC converters by statistical feature estimation using GPR and GA. IEEE Trans. Power Electron. **35**(10), 10945–10957 (2020)
9. Bhuvaneswari, A., Selvakumar, S.: Anomaly detection framework for Internet of Things traffic using vector convolutional deep learning approach in fog environment. Futur. Gener. Comput. Syst. **113**(1), 255–265 (2020)
10. Podgorelec, B., Turkanovi, M., Karakati, S.: A Machine learning-based method for automated blockchain transaction signing including personalized anomaly detection. Sens. (Basel, Switzerland) **20**(1), 147 (2020)
11. Tanuska, P., Spendla, L., Kebisek, M., et al.: Smart anomaly detection and prediction for assembly process maintenance in compliance with industry 4.0. Sensors **21**(7), 2376 (2021)
12. Shen, H.M., Zhou, G.J.: Repair of erasure codes of distributed storage data based on decision tree model. Comput. Simul. **39**(6), 473–477 (2022)
13. You, X., Hu, X., Feng, Z., et al.: Recognizing protein-metal ion ligands binding residues by random forest algorithm with adding orthogonal properties. Comput. Biol. Chem. **98**(1), 1–5 (2022)
14. Cui, X., Wang, S., Jiang, N., et al.: Establishment of prediction models for COVID-19 patients in different age groups based on Random Forest algorithm. QJM Mon. J. Assoc. Phys. **114**(11), 795–801 (2021)
15. Hosseinzadeh, M., Rahmani, A.M., Vo, B., et al.: Improving security using SVM-based anomaly detection: issues and challenges. Soft. Comput. **25**(4), 3195–3223 (2021)