A local failure identification technology of Industrial 4.0 server based on spark big data processing

Lixia Hao

Department of Information Engineering,
Hebei Chemical and Pharmaceutical College,
Shijiazhuang 050026, China
Email: haolixia2018@163.com

Abstract: The traditional server local failure identification method has the problems of long identification time and high identification error rate. Therefore, this paper proposes the Industrial 4.0 server local failure identification technology based on spark big data processing. Firstly, the spark programming model is used to obtain the server node data distribution, and the LMD method is used to extract the local features of Industrial 4.0 server. Secondly, the redundant parameters of failure characteristics are eliminated by PF component screening. Then, the type of failure fault is determined by judging the threshold selection. Finally, the current sensor is used to determine the fault location and complete the local failure identification of the server. The results show that the total recognition time of this method is no more than 16 s, and the recognition error rate is 0.025, which shows that this method has good recognition performance.

Keywords: Industry 4.0 theory; local failure; spark programming model; big data processing.

Reference to this paper should be made as follows: Hao, L. (2022) 'A local failure identification technology of Industrial 4.0 server based on spark big data processing', *Int. J. Internet Manufacturing and Services*, Vol. 8, No. 3, pp.195–207.

Biographical notes: Lixia Hao graduated from the Hebei University of Technology, in 1999. Currently, she is an Associate Professor of the Hebei Chemical and Pharmaceutical Vocational and Technical College. Her research interests include big data technology and application, and in-depth learning.

1 Introduction

In recent years, due to the rapid rise of new technologies of the internet, the industrial system of the traditional internet has also changed and expanded rapidly. With this, the magnitude of logs and data in the industry is also different. In the same breath, TB and Pb data can no longer be processed by the traditional single machine. Therefore, finding an effective and universal data processing method has become an urgent problem in the real world. Spark came into being. Spark is an iterative distributed computing engine (Qiao, 2020; Lai et al., 2020; Xu et al., 2020). The reason why it is faster than Mr is that the

intermediate job step results of spark can be stored in the memory of the host, and HDFS will no longer be accessed frequently, thus reducing disk IO. Therefore, spark makes more use of complex algorithm environments such as iteration. The local failure of the server is caused by problems in the resolution cache, network settings, IP address and DNS address settings, and network service start-up. At present, many articles have studied the local failure of server, and have achieved good results.

Yang et al. (2019) proposed a smart grid data server traffic anomaly detection algorithm based on width learning. The width learning algorithm is used to collect the fault characteristic attributes of server nodes, the cloud computing method is used to classify the fault attributes of server nodes, and the server node fault identification model is constructed through AFD algorithm, Finally, through the server node fault identification model, the server node fault information interaction is realised to complete the server node fault identification. This method can determine the fault identification threshold range and has strong reliability, but the identification effect is limited. Zhang et al. (2021) proposed a new server state detection and fault identification method, applied the narrowband internet of things technology to the server, obtained the hourly operation state information of the server, controlled the server state according to the substrate management controller, and determined the server state according to the relevant indicators such as server core voltage and current, This method can improve the fault identification efficiency of the server, but the system stability is poor. Wang (2020) designed a server fault identification model, obtained the system server fault characteristics through data mining technology, and constructed a server fault classifier using cloud computing technology. This method can effectively improve the accuracy of fault identification, but the efficiency of fault identification is low.

Therefore, this paper proposes a local failure identification technology of Industrial 4.0 server based on spark big data processing. The specific research ideas are as follows.

The first step is the local feature collection of Industrial 4.0 server. The spark programming model is used to obtain the data distribution on the server node, construct the RDD structure, determine the key partition rules, and extract the local features of Industrial 4.0 server by LMD method.

The second step is data feature preprocessing. The dimension of the extracted feature vector is reduced according to the factor analysis method, and the redundant parameters of local failure features of the server are eliminated through pf component screening.

The third step is to determine the type of local failure of Industrial 4.0 server. The server local characteristic pattern recognition sample database is constructed, the server local failure fault discrimination strategy is designed by Bayesian method, and the type of server local failure fault is determined by judgement threshold selection.

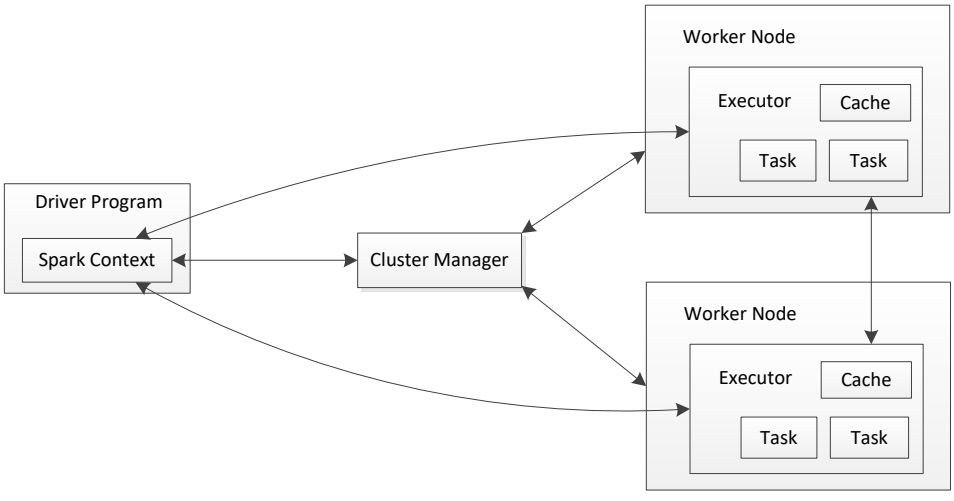
The fourth step is local failure identification of Industrial 4.0 server. Use the current sensor to obtain spark big data, determine the location of local failure of the server, and complete the identification of local failure of the server.

The fifth step is to verify the effectiveness of the local failure identification method of Industrial 4.0 server through experiments, and summarise the full text.

2 Partition model construction of spark RDD under the background of Industry 4.0

Industry 4.0, that is, the industrial internet, is an application management formed by the deep integration of IT technology and industrial manufacturing technology. Its essence is based on the interconnection and interworking of personnel, equipment, materials, control system and information system, and realises intelligent control, operation optimisation and organisational change through comprehensive collection of production data, edge computing, industrial mechanism modelling and cloud analysis and sharing (Liu, 2020; Tang et al., 2020). Through the deep integration of industrial internet and manufacturing, the optimisation of quality, efficiency, cost and inventory is generated to realise intelligent production, network coordination and flexible small batch diversified customised production, so that the factory can serve customers with the lowest cost, the shortest delivery time and the highest quality. The essence of Industry 4.0 is intelligence (Du et al., 2019; Zečević and Colin, 2019; Rodrigues et al., 2018; Song et al., 2019).

Figure 1 Spark workflow

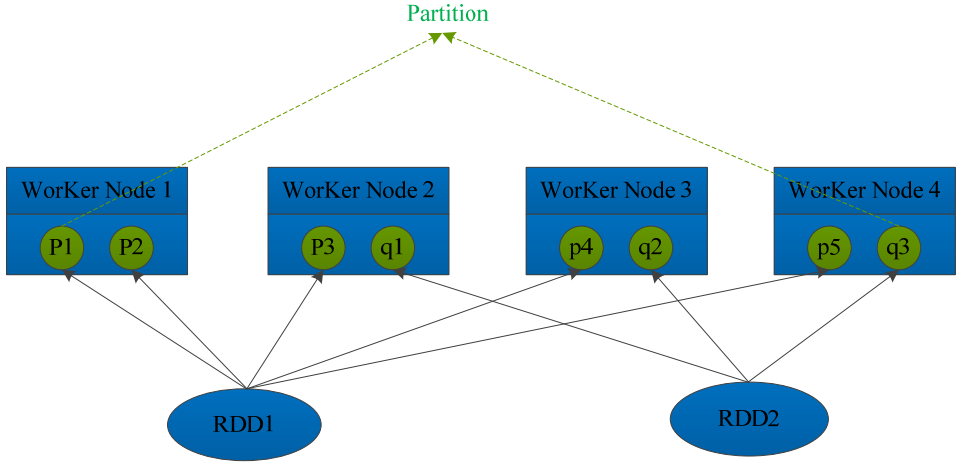


Although the map reduce programming model can process a large amount of data, show good parallelisation and processing efficiency, and provide the guarantee of data fault tolerance and reliability, because map reduce is calculated based on disk, the calculation results in the map stage will fall, and reduce needs to pull the required data from the disk, The result data of the reduce phase will also be written to the disk, and a lot of disk I/O will be carried out in this process (Tian et al., 2019; Yao et al., 2020). The spark programming model is to solve the bottleneck problem of map reduce in the fast computing scenario. Spark programming model is a distributed computing framework and can be based on memory operation. Spark, a memory-based computing mode, is very suitable for the operation and implementation of complex algorithms such as machine learning, and has more diversified and complex operators. The most important data structure of spark is RDD, which is an elastic and distributed dataset. The basic process of spark operation is shown in Figure 1. The cluster manager is the soul and is responsible for scheduling (Yang et al., 2019). The application (spark program written by

the user) in Figure 1 is composed of a driver module and several actuator modules. The driver module has the spark context and can schedule tasks (Zhang et al., 2020; Singh et al., 2019; Zarindast and Sharma, 2021; Duan et al., 2019). The actuator module is independent of the spark context, that is, it is our work node and executes the tasks of the application:

In spark, the data in an RDD is divided into different partitions. When the job runs, the data in RDDS of different partitions will be distributed in different cluster nodes. Figure 2 shows the data distribution of two RDDS on different nodes.

Figure 2 Partition model of spark RDD (see online version for colours)



Spark has two different partition methods: hash partition and range partition. Generally speaking, only data in the form of < key, value > can be partitioned. Moreover, when a job has the processing logic of shuffle operation, such as group by key, reduce by key and other operators, the partition operation will be triggered to re disrupt the partition of data. The partition rule is to partition according to the key. Extract the local feature vector of the server according to the key partition rules.

3 Research on local failure identification technology of Industrial 4.0 server based on spark big data processing

3.1 Server local feature vector extraction

Based on the local feature vector of the server and the key partition rules, the fault diagnosis needs to use the LMD method to extract the local features of the server (Kang et al., 2019). Fault diagnosis needs to be based on the server local feature vector. In this paper, LMD method is used to extract the server local feature. For the signal data collected by sensors in the server structure, it is divided into multiple linear combinations by LMD method, and the PF component of each linear combination is consistent. Based on the LMD method, the original signal decomposition formula can be obtained:

$$x = \sum_{p=1}^{k} \psi_p + \mu_k \tag{1}$$

In the formula, x represents the original signal, ψ represents the energy matrix, μ represents the average trend of the signal, k represents the number of signal samples, and p represents the type of eigenvector.

Compared with the conventional method, the LMD feature extraction method has the advantages of less iterations and low computational complexity. In the process of feature extraction, the server running signal will be complicated due to a variety of noise interference factors. Therefore, in the process of original signal decomposition, the decomposition result of LMD method contains some false pf components. Based on this, the server local failure fault identification will greatly reduce the accuracy of fault identification. Under normal conditions, the decomposed real pf component has strong correlation with the original signal. Therefore, in the process of eliminating false pf components, PF components with poor correlation with the original signal are selected and removed to improve the accuracy of server local failure fault identification. Through the inner product calculation, the correlation coefficient between two discrete signals is obtained in the finite length state:

$$c = \frac{|< X, Y >|}{\|X\|^2 \|Y\|^2} \tag{2}$$

In the formula, X, Y represents discrete signal and c represents signal correlation coefficient. The calculation result of equation (2) reflects the correlation between discrete signals, and the greater the calculation result, the greater the correlation between them. When two discrete signals are completely linear, the calculation result of correlation coefficient is 1.

To sum up, the false pf component decomposed by LMD method is removed by correlation coefficient. In the actual operation, the correlation coefficients between all pf components and the original signal are calculated in advance, and the average correlation coefficient is selected as the evaluation threshold. If the correlation coefficient is greater than the threshold, the PF component is retained; otherwise, the PF component is eliminated. The threshold λ is expressed as:

$$\lambda = \frac{\max(c)}{\eta} \tag{3}$$

In the formula, η represents the total number of PF components. The false pf component screening strategy designed above is verified and applied to the signal inspection process to clarify the application effect of the strategy.

$$x = x_1 + x_2 \tag{4}$$

In the formula, x_1, x_2 are the two sub signals decomposed by the original signal, where:

$$x_1 = \begin{cases} 5\cos 2t \cos 100t, & 0 \le t \le 0.34 \\ 5\cos 4t \cos 200t & 0.34 \le t \le 0.68 \end{cases} \tag{5}$$

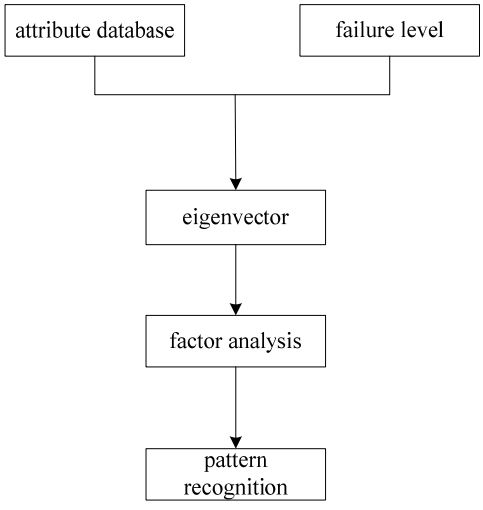
$$x_2 = \sin 750t, 0 \le t \le 0.68 \tag{6}$$

where cos is the cosine function and t is the time. Through the above calculation method, the original signal is divided into one margin and four pf components, and the PF component that does not contain the original signal component is selected to eliminate the false component caused by the endpoint effect. The verification shows that the PF component screening method proposed in this paper has excellent effect. Through the above processing method, the final server local feature vector extraction result is obtained.

3.2 Determination of the type of server local failure

Server local failure identification needs to be based on the server local eigenvector sample library. According to the research, the actual establishment process of sample database is shown in Figure 3.

Figure 3 Establishment of server feature vector recognition sample database



Using the LMD feature extraction method, the feature vector of the server is preliminarily obtained. In order to reduce the computational complexity of pattern recognition fault diagnosis model, factor analysis is used to reduce the dimension of the extracted feature vectors, obtain the final feature vectors, and summarise them to form a pattern recognition sample database.

Dimension reduction analysis is one of the key links in the establishment of sample database. According to the original feature vector, the correlation index between multiple features is calculated. The closer the calculation result is to 1, it means that the similarity between the two battery feature vectors is very high. In the subsequent recognition process, one of the two feature vectors can represent the other at random. At this time, one feature vector can be retained. On the contrary, the closer the correlation index is to 0, the greater the difference between the two eigenvectors. The two eigenvectors need to be completely preserved. Through the above operations, the extracted feature vector is processed to build a pattern recognition feature sample library.

Based on the above sample database, a reasonable discrimination strategy is designed by Bayesian method. For the server fault diagnosis sample data, select a certain type of

sample feature vector to judge the probability that the feature vector belongs to a certain fault type. The specific fault identification calculation formula is expressed as follows:

$$P(w_i \mid \varepsilon) = \frac{P(\varepsilon \mid w_i) P(w_i)}{\sum_{i=1}^{n} P(\varepsilon \mid w_i) P(w_i)} \tag{7}$$

In the formula, i represents the fault sorting, w_i represents the server fault type, ε represents the server sample feature vector, $P(w_i)$ represents the proportion of the server fault sample in the feature sample library, and $P(\varepsilon|w_i)$ represents the probability of determining that the feature vector belongs to the fault type w. The multi-dimensional Gaussian distribution results are calculated based on the local feature sample database of the server:

$$P(\varepsilon \mid w_i) = \frac{1}{x + L(\varepsilon - \tau_i)^{\frac{d}{2}}} \tag{8}$$

In the formula, d represents the feature vector dimension, τ represents the mean vector, and L represents the covariance matrix. The multi-dimensional Gaussian distribution result is calculated according to equation (8) and substituted into equation (7), so as to clarify the relative probability that a certain type of server eigenvector operation data sample belongs to a certain fault type. The output of the server local failure fault identification result is selected according to the judgement threshold. When the relative probability calculation result exceeds the threshold, the type of the server local failure fault is determined.

3.3 Server local failure fault identification and location

Generally speaking, the causes of server failure are complex and often caused by multiple problems. Based on the server failure fault identification results, the specific fault points are located for the convenience of subsequent maintenance personnel.

In the process of server failure fault location, multiple current sensors are set in the server to obtain the spark big data current value flowing through the server in real-time. Set the judgement threshold according to the current value under normal operation. The threshold calculation formula is:

$$F = \sum v|h| \tag{9}$$

In the formula, h represents the threshold calculation factor of the server, v represents the actual current value in the server, and F represents the threshold judgement factor.

The threshold value is obtained according to the calculation result of equation (9). After the fault occurs, the current data collected by each sensor is obtained and compared with the threshold value to obtain the judgement result. When the collected current value is greater than the preset threshold, it indicates that there is fault current in this part, and this section can be divided into fault section. On the contrary, it indicates that the server working area is in normal operation. Through the above calculation, the fault section is marked and sent to the staff monitor to complete the intelligent identification and positioning of server local failure faults.

The specific process of server local failure fault identification is as follows:

1 According to the key partition rules, the LMD method is used to extract the local features of the server, and the false pf component of the features is removed by the correlation coefficient

2 Factor analysis is used to reduce the dimension of the extracted feature vectors, obtain the final feature vectors, and summarise them to form a pattern recognition sample database

3 Bayesian method is used to design a reasonable discrimination strategy, judge the probability of local failure type of server, and calculate the result of multidimensional Gaussian distribution

4 Determine the type of local failure of the server, set the current sensor to obtain the spark big data current value, divide the fault section, and realise the intelligent identification of local failure of the server.

4 Experiment

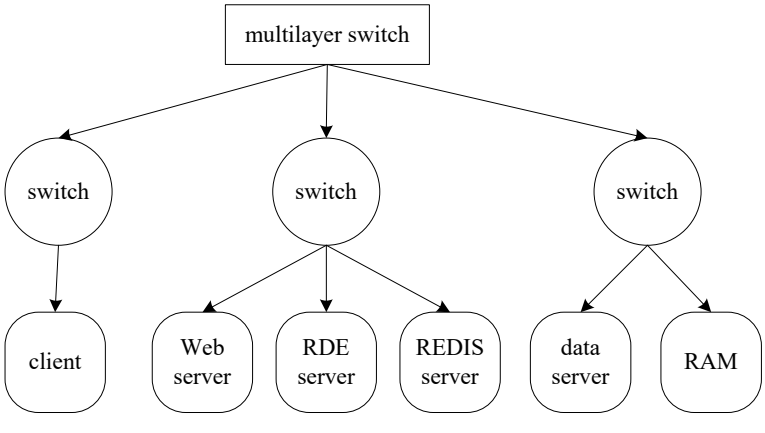
4.1 Experimental scheme

In order to verify the method designed in this paper, simulation experiments are specially carried out to verify the effectiveness of the method from two aspects of function and performance. Before the experiment, first build the simulation environment. The required test server and client environment are shown in Table 1.

Table 1 Test server and client environment

Entry	*Testing the server environment*	*Test the client environment*
OS	Windows 10	Windows 10
IIS	Version 6.0	Version 6.0
Framework	Version 4.0	Version 4.0
SQL server	Version 2020	Version 2020

Figure 4 Schematic diagram of simulation environment structure



According to the test server and client, the specific simulation environment is built, and its specific structure is shown in Figure 4.

The number of experimental samples is shown in Table 2.

Table 2 Experimental samples of server fault identification

Number of experiments	*Number of training samples/piece*	*Number of test samples/piece*
10	300	150
20	600	380
30	200	100
40	150	90
50	500	360

Among them, the types of server failures include windows service failure to start, domain server failure, memory failure, motherboard failure, etc.

Select the two indicators of server local failure identification efficiency and server local failure identification accuracy, and conduct the experiment by using spark big data processing identification method, width learning algorithm identification method and narrowband internet of things identification method. See Section 4.2 for the specific process of the indicators.

4.2 Experimental indicators

1 Server local failure identification time: The longer the server local failure identification time, the lower the server local failure identification efficiency. On the contrary, the shorter the server local failure identification time, the higher the server local failure identification efficiency.

2 Misjudgement rate of server local failure identification: The higher the misjudgement rate of server local failure identification, the lower the accuracy of server local failure identification. On the contrary, the lower the misjudgement rate of server local failure identification, the higher the accuracy of server local failure identification.

3 Experimental comparison method: The server local failure identification method based on spark big data processing, the server local failure identification method based on width learning algorithm and the server local failure identification method based on narrowband internet of things are used for experimental verification.

4.3 Result analysis

4.3.1 Server local failure identification efficiency

In order to verify the identification efficiency of Industrial 4.0 server local failure identification, spark big data processing identification method, width learning algorithm method and narrowband internet of things identification method are used to detect the time of server local failure identification. The results are shown in Table 3.

Table 3 Industry 4.0 server local failure identification time

Number of servers	*Total server partial failure identification duration/(s)*		
	Spark Identifies big data processing	*Width learning algorithm*	*Narrowband internet of things method*
10	2.43	21.65	27.45
20	5.66	27.46	56.64
30	12.64	32.57	73.92
40	13.56	67.43	97.67
50	15.12	126.65	167.83

According to the analysis of Table 3, when the number of servers is 10, the total time of server local failure identification of the identification method of width learning algorithm is 21.65 s, the total time of server local failure identification of the identification method of narrowband internet of things is 27.45 s, and the total time of server local failure identification of spark big data processing identification method is 2.43 s. When the number of servers is 30, the total time of server local failure identification of the identification method of width learning algorithm is 32.57 s, the total time of server local failure identification of the identification method of narrowband internet of things is 73.92 s, and the total time of server local failure identification of spark big data processing identification method is 12.64 s. When the number of servers is 50, the total time of server local failure identification of the identification method of width learning algorithm is 126.65 s, the total time of server local failure identification of the identification method of narrowband internet of things is 167.83 s, and the total time of server local failure identification of spark big data processing identification method is 15.12 s. The efficiency of server local failure identification in this paper is obviously short, because this method uses spark big data processing technology to extract server local features, simplify the amount of identification data, make the total identification time of the proposed method significantly lower, and fully prove the superiority of the proposed method.

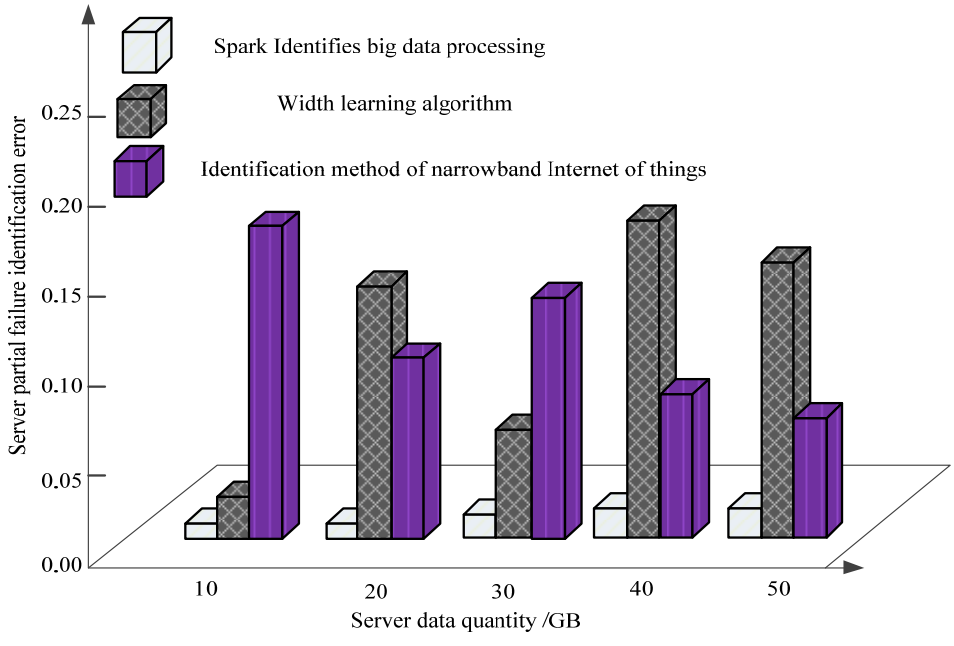
4.3.2 Misjudgement rate of server local failure identification

In order to further verify the accuracy of the identification results of the proposed method, a large number of identification data are used to compare the server local failure identification errors using spark big data processing identification method, width learning algorithm and narrowband internet of things identification method. The identification results are shown in Figure 5.

According to the analysis of Figure 5, there are differences in the misjudgement rate of server local failure identification under different methods. When the amount of server data is 10 GB, the misjudgement rate of server local failure identification of width learning algorithm is 0.03, the misjudgement rate of server local failure identification of narrowband internet of things identification method is 0.19, and the misjudgement rate of server local failure identification of spark big data processing identification method is 0.01; When the amount of server data is 50 GB, the misjudgement rate of server local failure identification of width learning algorithm is 0.17, the misjudgement rate of server local failure identification of narrowband internet of things identification method is 0.132, and the misjudgement rate of server local failure identification of spark big data

processing identification method is 0.025; Compared with the recognition method of width learning algorithm and the recognition method of narrow-band internet of things, the misjudgement rate of spark big data processing recognition method is significantly lower, which fully proves that the proposed method has good recognition performance and can obtain satisfactory recognition results.

Figure 5 Misjudgement rate of server local failure identification (see online version for colours)



5 Conclusions

This paper proposes a local failure identification technology of Industrial 4.0 server based on spark big data processing. The spark programming model is used to obtain the data distribution on the server nodes, construct the RDD structure, determine the key partition rules, extract the local features of Industrial 4.0 server by LMD method, and complete the local feature collection of Industrial 4.0 server. The dimension of the extracted feature vector is reduced according to the factor analysis method, and the redundant parameters of local failure features of the server are eliminated through pf component screening to realise data feature pre-processing. The server local characteristic pattern recognition sample database is constructed, the server local failure fault discrimination strategy is designed by Bayesian method, and the type of server local failure fault is determined by judgement threshold selection. The current sensor is used to determine the location of local failure of the server and complete the identification of local failure of the server. The following conclusions are drawn through experiments:

1 When the number of servers is 50, the total time of server local failure identification of spark big data processing identification method is 15.12 s. The efficiency of server local failure identification in this paper is obviously short, because this method uses spark big data processing technology to extract server local features, simplify the amount of identification data, make the total identification time of the proposed method significantly lower, and fully prove the superiority of the proposed method.

2 When the amount of server data is 50 GB, the misjudgement rate of server local failure identification of spark big data processing identification method is 0.025, which fully proves that the proposed method has good identification performance and can obtain satisfactory identification results.

Acknowledgements

This work was funded by Science and Technology Project of Hebei Education Department, No. ZC2021219.

References

Du, H.R., Chen, J.H. and Qi, M.P. (2019) 'A forward secure multi server authentication protocol based on RSA', *Computer Science*, Vol. 46, No. 2, pp.409–413+437.

Duan, J., Li, G. and Asthana, N. (2019) CSI2: cloud server idleness identification by advanced machine learning in theories and practice, Springer, Cham, Vol. 18, No. 2, pp.68–76.

Kang, B.Y., Jie, M.M. and Si, L. (2019) 'Research on multi cloud server authentication scheme based on biometric technology', *Information Network Security*, Vol. 16, No. 6, pp.45–52.

Lai, F.G., Li, J.W. and Dong, Y.Z. (2020) 'PC server fault prediction analysis and maintenance treatment', *Electronic Technology and Software Engineering*, Vol. 13, No. 32, pp.76–89.

Liu, J.C. (2020) 'Troubleshooting server startup failure', *Network Security and Informatization*, Vol. 45, No. 1, pp.152–153.

Qiao, L.Y. (2020) 'Failure analysis and maintenance treatment of PC server', *Electronics World*, Vol. 599, No. 17, pp.207–208.

Rodrigues, M., Santos, M.Y. and Bernardino, J. (2018) 'Big data processing tools: an experimental performance evaluation', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Vol. 12, No. 32, pp.877–889.

Singh, R., Iquebal, M.A. and Mishra, C.N. (2019) 'Development of model web-server for crop variety identification using throughput SNP genotyping data OPEN', *Scientific Reports*, Vol. 9, No. 1, pp.68–76.

Song, B.D., Zhang, L.C. and Jiang, Q.Z. (2019) 'Research on distributed big data analysis algorithm based on spark', *Computer Application and Software*, Vol. 36, No. 1, pp.45–50.

Tang, L., Wang, S.Y., Wang, Y.G., Zhang, T.R. and Dong, Y.J. (2020) 'A server fault auxiliary diagnosis system based on video image recognition', *Science and Technology Bulletin*, Vol. 36, No. 8, pp.58–61+86.

Tian, L., Qi, L.H. and Li, Q. (2019) 'Power streaming big data analysis architecture and application based on spark streaming', *Electric Power Informatization*, Vol. 17, No. 2, pp.23–29.

Wang, H.X. (2020) 'Identification model of server failure of ship cloud computing system', *Ship Science and Technology*, Vol. 41, No. 4, pp.158–160.

Xu, X.G., Lin, C.W., Chen, W.W. and Jiang, Y. (2020) 'Research on fault identification method of power equipment based on augmented reality technology', *Electronic Design Engineering*, Vol. 28, No. 23, pp.155–158+163.

Yang, Y.J., Qiu, Y. and Zhan, L.C. (2019) 'Smart grid data server traffic anomaly detection algorithm based on width learning', *Computer and Modernization*, Vol. 16, No. 9, pp.77–82+ 89.

Yao, T., Zheng, T., Xin, R., Wu, J.Y. and Chen, X. (2020) 'Distribution network operation and maintenance data processing based on Spark', *Information Technology*, Vol. 44, No. 5, pp.165–168.

Zarindast, A. and Sharma, A. (2021) 'Big data application in congestion detection and classification using Apache spark', Vol. 12, No. 35, pp.87–98.

Zečević, S.P. and Colin, T.J. (2019) 'AXS: a framework for fast astronomical data processing based on apache spark', *Astronomical Journal*, Vol. 65, No. 53, pp.123–136.

Zhang, C., Chen, N.K., Zhang, X. and Zhu, X.K. (2021). 'Server Status Detection and Fault Diagnosis System based on NB-iot', *Command Information System and Technology*, Vol. 12, No. 3, pp.96–100.

Zhang, X.J., Zhu, J.H. and Chen, Y. (2020) 'Distributed rough set attribute reduction algorithm based on spark', *Computer Application*, Vol. 40, No. 2, pp.518–523.