Fault Prediction of Process Industry Based on Fuzzy Clustering

Tao Mi, Luoyifan Zhong, Xin He, Yi Sun, Chenyang Yan and Hao Yue

March 29, 2022

Fault Prediction of Process Industry Based on Fuzzy Clustering

1st Tao Mi
School of computing science and Information Engineering
QILU Institute of Technology
Jinan, China
1006468498@qq.com

Luoyifan Zhong
School of computing science and Information Engineering
QILU Institute of Technology
Jinan, China
860133458@qq.com

Xin He
School of computing science and Information Engineering
QILU Institute of Technology
Jinan, China
1121952316@qq.com

4th Yi Sun
School of computing science and Information Engineering
QILU Institute of Technology
Jinan, China
728062172@qq.com

5th Chenyang Yan
School of computing science and Information Engineering
QILU Institute of Technology
Jinan, China
1871152939@qq.com

6th Hao Yue
School of computing science and Information Engineering
QILU Institute of Technology
Jinan, China
917971551@qq.com

***Abstract*—In the process of process industry production, once the failure will often bring heavy losses to the enterprise and the country in terms of human resources and financial resources, so it is very important to give early warning of the failure and take corresponding solutions. In this paper, a real-time fault prediction algorithm based on incremental model update is proposed for the continuous growth of industrial data flow, real-time update of objects, complex and variable attributes, and value attenuation with time. The algorithm consists of several steps, including data preprocessing, data flow clustering based on sliding window, abnormal data judgment, fault prediction and model prediction update. In each stage of the algorithm, Spark framework is used for parallel acceleration, so as to improve the efficiency of fault prediction.**

Keywords—data preprocessing, data stream clustering, abnormal data judgment, fault prediction, model prediction update

I. INTRODUCTION

In the process of process industry production, once the failure will often bring heavy losses to the enterprise and the country in terms of human resources and financial resources, so it is very important to give early warning of the failure and take corresponding solutions. We can monitor the production status in real-time through the process industry control management system and store the status data in the database in the form of data flow. These status data not only contain the description of the current production situation but also contain the prediction of the future production situation. Through online real-time and efficient fault monitoring of these status data, we can give early warning of possible future faults, and also dig out useful knowledge for actual production to predict future production conditions, so as to better assist production decision-making.

In boiler production, industrial data flow is obtained through data integration of all production links. The data flow collected in the actual production will inevitably have noise, so the accuracy of the data should be detected first, the wrong data should be corrected, and the data should be cleaned. Then the sequence of all links should be mined to get the correct sequence.

In this paper, the algorithm based on the difference extremum is used for timing calculation and timing adjustment of the original data. Firstly, the reference link is set, and then the delay time distance between the reference link and other links is calculated. Finally, the sequence order of each link is summarized, which lays a foundation for subsequent association rule screening.

For data flow, the problem of data value attenuation needs to be considered. The front-line production personnel are more concerned about the changes of boiler state in the recent period of time, so this paper adopts the data flow clustering algorithm based on sliding window to retain the data changes in the recent period. However, due to the numerous discrete states of industrial data, complicated and difficult to predict, we cannot set parameters in advance. Therefore, this paper proposes a parameter adaptive fuzzy clustering algorithm based on grid division, which can not only realize parameter adaptive, but also reduce clustering difficulty through grid division. The discrete degree of data can be reduced by clustering each link separately. Then, the new data flow is clustered continuously, and the clustering results before and after the arrival of data flow are compared whether there are obvious changes. If there are obvious changes, the concept migration can be determined. At this time, the boiler production state may change greatly, and the system will give an alarm. Then we conduct *Independent principal Component Analysis* (ICA) and *Principal Component Analysis* (PCA) on the data, and obtain the corresponding threshold value of the process industry through calculation. If the threshold value exceeds the threshold value, the system will issue an alarm. We also continue to carry out subsequent correlation chain mining, model prediction and other algorithms to predict the possible working conditions of the whole process industry, so as to provide workers with more information to assist decision-making in production.

II. RELATED WORKS

Nowadays, big data processing technology is widely applied in various fields, and its development and improvement are concerned by many experts [1]. With the widespread use of big data processing [2-4], parallel

This work was supported the Major Scientific and Technological Innovation Projects of Shandong Province (Grant No.2019JZZY010111).

computing has become more and more valued because of its characteristics. Parallel computing refers to the process of using multiple computing resources to solve computing problems simultaneously [5-7]. It is an effective means to improve the computing speed and processing ability of computer system [8-10]. Parallel computing frameworks such as OpenMP, Hadoop and Spark play an important role in many industries [11].

As different frameworks play different roles in different scenarios, through investigation and experiments, we find that Spark framework is more suitable for data mining than programming models of other frameworks for coal-fired boiler data [12]. Spark is a programming framework similar to Hadoop, and both use the MapReduce programming model. Because Spark is a parallel computing framework that uses memory units, its computing efficiency is tens or even hundreds of times higher than Hadoop. In addition, Spark is easier and simpler to code than Hadoop for the same scenario. The characteristics of Spark are more consistent with the production data of coal-fired boilers. In the fault prediction algorithm part, Spark framework is used to carry out parallel design of programs. Through the design and development of the process industry fault prediction system based on fuzzy clustering, the loss can be avoided in advance and the production decision-making of coal-fired boilers can be guided to optimize the production process and improve the production efficiency of enterprises.

As there are many links in the actual production process of the process industry, the data flow obtained from different links is also very messy [13]. In many cases, if the original data is directly mined and calculated, the time for fault discovery will be greatly prolonged. In addition, wrong fault information may be mined, which makes the efficiency of fault prediction very low. Therefore, the clustering algorithm is firstly adopted in this paper to reduce the degree of dispersion of data and the difficulty of calculation, and a few types of representative data are used to represent the data of all links [14-16]. Common clustering algorithms include partition-based clustering algorithm, hierarchical clustering algorithm, density-based clustering algorithm [17], grid-based clustering algorithm [18], model-based clustering algorithm and grid density-based clustering algorithm. In addition, some novel clustering algorithms have been proposed by scholars at home and abroad, such as spectral clustering algorithm based on Markov random Walk, clustering algorithm based on deterministic annealing, and algorithm combining intelligent optimization algorithm with clustering. Different types of clustering algorithms have their own advantages and disadvantages, so far, no one clustering algorithm can have a good clustering effect on all data sets.

Therefore, we propose an efficient fuzzy C-means clustering algorithm called G-FCM based on grid with unknown cluster number. G-FCM algorithm can realize parameter adaptive clustering, and greatly reduce the cost of clustering through mesh division. Rl-FCM [19] algorithm is an extension of traditional *Fuzzy Clustering algorithm* (FCM). It has the characteristic of parameter self-adaptation and can realize clustering without input parameters, but it costs too much for high-dimensional big data. The clustering algorithm based on grid density is a common clustering algorithm. This algorithm not only has the advantage of density clustering algorithm to find arbitrary shape clusters, but also has the high efficiency of grid clustering algorithm. In addition, the algorithm is insensitive to noise data and has good denoising ability. Moreover, the grid density clustering algorithm has a strong scaling ability, so it is very suitable for clustering large-scale data sets. Therefore, grid clustering is introduced into RL-FCM algorithm to solve the complex problem in the case of large amount of data.

Classification of fault diagnosis is different from different angles [20, 21], and the classification of fault diagnosis has undergone many changes. The classification of fault diagnosis types can be traced back to 1999. Frank, a scholar, studied and summarized fault diagnosis technologies and divided them into three categories: those based on analytic model, those based on knowledge and those based on signal processing [22]. With the development of related technologies, more and more experts and scholars have studied them. Finally, the classification of fault diagnosis is determined into three types: analysis model based on qualitative method, analytical model based on quantitative method, and process historical data.

In the production state prediction stage [23-25], it is necessary to calculate the correlation degree for different production stages, so as to find the optimal correlation chain. The discovery of association chain requires a lot of calculation, so it is also a time-consuming link in the whole system. In order to reduce unnecessary time, it is very important to select a suitable association rule algorithm. In the context of the era of big data, the research of experts and scholars on mining algorithms is more and more in-depth, among which the most famous classic is the Mining algorithm for Apriori association rules published by R. Agrawal et al. in 1994. By searching data sets for many times and combining conditional probability, frequent item sets with minimum support can be generated. Finally, association rules are generated by minimum confidence filtering.

III. Fault prediction of process industry

A. Data preprocessing

The industrial data flow collected in process industrial production is basically complete, but one or several sampling cycles may be lost in a few cases, and there will still be some industrial noise in the collected data flow when there is no failure of the acquisition equipment. Therefore, data cleaning should be carried out on production status data first [26-28], including empty value filling, error checking and other operations. Perform an overall filter on boiler data. Then need to sequential adjustment of state data, because in the process industry in the flow of industrial data collected in each record is every aspect at the same time the state of the data, but does not consider the sequence of each link between the order and state the influence of the propagation delay, data time dislocation, delays may occur. If we want to mine the correlation between these links, we must require that the data of interaction between these links is corresponding to the correct. Therefore, it is necessary to discover the time sequence of boiler flow objects and adjust the time sequence of data. There is a correlation between the boiler process objects, which is usually one-way. The state of the previous link changes and the next link responds. This is reflected in the process object, which is that each link has a chronological order.

However, in the real-time data collection of boiler status, data collection is based on the same moment, and due to the complex production equipment and many links, there is a great lag in the change of parameters, and there is a relative delay in the state transmission. The timing sequence of the collected data is in disorder. The data of each link at the same time should not be the data generated by the response of other links, showing asynchrony. Moreover, the temporal relationship between these links will lay a foundation for the subsequent screening of knowledge discovery. Timing adjustment is mainly divided into two steps. The first step is to mine the timing sequence between each link and determine the timing relationship, including the sequence of links and the relative delay interval. In the second step, according to the obtained time sequence relation, the corresponding time translation of the old data is carried out to obtain the time sequence data.

The production process of process object includes many complex systems and many production steps. Each link connects with each other to form the whole production process. Among them, there will be state propagation between links. When one link has state fluctuation, the other link will also have state fluctuation soon. This is also an important sign of boiler state fluctuation. According to the characteristics of state fluctuation of flow objects, this paper designs a timing sequence discovery algorithm based on the difference extremum. The algorithm is mainly divided into three steps. Then the delay interval time of each link is calculated based on the results of extremum points between links. Finally, according to the relative delay interval of each link, the sequential order of each link is calculated.

According to the obtained delay spacing array q, the time order L of each link is obtained. Take the data X_m of the benchmark link *m* as the benchmark, then the data of other links at time *t* isX_n:

$$X_n(t) = X_n(t + \Delta t_{mn}) \tag{1}$$

If Δt_{mn}>0, then all data of X_n as a whole pull Δt_{mn} time; If Δt_{mn}<0, then all data of X_n will be pulled down at Δt_{mn} moment; If Δt_{mn} is equal to 0, then the X_n data does not need to be moved. After adjustment, time sequence data can be obtained. Time sequence data represents the data that the subsequent links respond to when the state of the link changes at the same time. In the subsequent mining of association chain, the time order between links can also filter the generated association rules.

B. *Data stream clustering algorithm based on sliding window*

Due to the large number of discrete states in industrial data, mining association chain directly from the original data will not only be very time-consuming, but also lead to mining excessive association rules, and then the density of available value of knowledge discovery results is too low. Reducing the discrete state in the industrial data is the most important place to solve this problem, and the classification and aggregation method can be used for the industrial data. And in the process of industrial production, the operating conditions of equipment include normal, basically normal and failure and other states. You can use a small number of data to represent the data state of all links. In this paper, clustering algorithm is used to classify and summarize industrial data automatically, and each link carries out clustering operation independently. The clustering result obtained is no longer the specific value, but the corresponding clustering category of the value. Moreover, the data flow clustering algorithm based on sliding window is adopted in this paper, which can better represent the boiler production status information in the recent period of time. With the continuous arrival of industrial data flow, the boiler working condition information will be updated iteratively.

The basic idea is to take all data points as the initial cluster center, that is, the number of data points is the initial number of clusters. After that, use the cluster's mixed ratio of α _k, which is like the weight of a cluster, and finally discard the number of data points in which these clusters α _k values are less than 1. The algorithm can get the optimal cluster number iteratively until convergence .

The new data set is fed into the parameter adaptive method to realize clustering. The objective function of the algorithm is as follows:

$$\mathrm{J}\ (\mathrm{U}.\alpha.\lambda_1.\lambda_2.\mathrm{V}) = \\ \sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\,d^2{}_{ik} - \gamma_1\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\ln\alpha_k + \\ \sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\ln\mu_{ik}\,\gamma_3 n\sum_{k=1}^{c}\alpha_k\ln\alpha_k - \\ \lambda_1\left(\sum_{k=1}^{c}\mu_{ik}-1\right) - \lambda_2\left(\sum_{k=1}^{c}\alpha_k-1\right)) \tag{2}$$

Where V represents the set of its clustering center, V=represents its Euclidean distance, represents its fuzzy partition matrix, is its mixing ratio, and represents the probability that a data point belongs to the KTH cluster. as learning function, to learn to adjust the influence of the bias of entropy.

When the data is fed in, the number of data points is used as the number of initial clusters to solve the initialization problem. The mixture ratio is constantly updated through the objective function, and clusters with a mixture ratio value less than 1 / data points are discarded so that the optimal number of clusters can be automatically found based on the data structure. Given the number of iterations t, the initial value is t=1.The initial learning rate is set as .When the number of clusters is stable, the competition of mixing ratio will stop. The update formula is as follows:

$$\gamma_1{}^{t}{=}e^{-t/10} \tag{3}$$

$$\gamma_2{}^{t}{=}e^{-t/100} \tag{4}$$

$$\gamma_3{}^{t}{=}\min\left(\frac{\sum_{k=1}^{c} exp(-\delta n|\alpha_k{}^{new}-\alpha_k{}^{old}|)}{c}.\ \frac{1-max\left(\frac{1}{n}\sum_{i=1}^{n}\mu_{ik}\right)}{(-max\alpha_k{}^{old}\sum_{t=1}^{c}\alpha_t{}^{old}\ln\alpha_t{}^{old})}\right) \tag{5}$$

By taking the partial derivative of the Lagrangian in eq. 5 with respect to u_{ik} and setting them to be zero, it becomes $\frac{\partial\,\mathrm{f}}{\partial\,\mathrm{u}_{\mathrm{ik}}} = \mathrm{d}^2{}_{\mathrm{ik}} - \gamma_1 \ln \alpha_{\mathrm{k}} + \gamma_2(\ln \mathrm{u}_{\mathrm{ik}} + 1) - \lambda_1{=}0$ and then $\ln \mathrm{u}_{\mathrm{ik}} = \frac{-\mathrm{d}^2{}_{\mathrm{ik}} + \gamma_1 \ln \alpha_{\mathrm{k}} + \lambda_1 - \gamma_2}{\gamma_2}$. Thus, the updating equation for u_{ik} is obtained as follows:

$$u_{ik} = exp(\frac{-d^2{}_{ik}+\gamma_1\ln\alpha_k}{\gamma_2})/\sum_{t=1}^{c} exp(\frac{-d^2{}_{it}+\gamma_1\ln\alpha_t}{\gamma_2}) \tag{6}$$

Thus, the updating equation for can be obtain as follows:

$$\propto^{(new)}{}_k = \frac{1}{n}\sum_{i=1}^{n} u_{ik} + \frac{\gamma_3}{\gamma_1}\propto^{(old)}{}_k(\ln\propto^{(old)}{}_k - \\ \sum_{t=1}^{c}\propto^{(old)}{}_k\ln\propto^{(old)}{}_k) \tag{7}$$

The steps of the final clustering algorithm can be expressed as follows:

Step1： Set the initial value as the following formula and let t=1 enter the iterative process:$\boldsymbol{c^{0}=n}$,$\boldsymbol{v^{0}{}_{k}=x_{i}}$,$\boldsymbol{\propto^{0}{}_{k}=\frac{1}{n}}$.

Step2: Update $\boldsymbol{v^{(t-1)}{}_{k}}$， $\boldsymbol{\propto^{(t-1)}{}_{k}}$， $\boldsymbol{c^{(t-1)}}$ by eq9.

Step3: Update $\boldsymbol{\gamma_{1}{}^{t}}$， $\boldsymbol{\gamma_{2}{}^{t}}$ by eq6 and eq7.

Step4: Calculation $\boldsymbol{\propto^{(t)}{}_{k}}$ by eq10 according to $\boldsymbol{\propto^{(t-1)}{}_{k}}$ and $\boldsymbol{u^{(t)}{}_{ki}}$.

Step5: Updated $\boldsymbol{\gamma_{3}{}^{t}}$, according to $\boldsymbol{\propto^{(t-1)}{}_{k}}$， $\boldsymbol{\propto^{(t)}{}_{k}}$ by eq8.

Step6:Calculate $\boldsymbol{v^{t}{}_{k}}$ through $\boldsymbol{u^{(t)}{}_{ki}}$ by eq11.

Step7: When the cluster center in V has almost no change in the last two iterations, that is, Max$\boldsymbol{\|v^{t}{}_{k}-v^{(t-1)}{}_{k}\|<\varepsilon}$, means the completion of clustering.The data represented by the center of gravity of each grid is brought in to complete the whole clustering.

C. *Robust ICA-PCA detection method*

This chapter is the most important part of fault prediction for process industry. In the previous stage of data pretreatment and data stream clustering, we processed the boiler data and obtained the data which is convenient for fault prediction. When in data stream clustering monitoring stage, the concept of data has time to move, can send out alarm prompt, traditional industrial fault prediction methods, are subject to the actual working condition of most industrial production in the process of collecting data by change, the influence of the factors such as noise interference, the system adjustment, thus affect the efficiency and accuracy of the fault prediction. Therefore, we applied robust ICA-PCA algorithm . The advantage of this method is that it considers the complex situation of process industry production, introduces the fault diagnosis method combining two loops, and performs robust ICA real-time measurement on the data of over-parameter adaptive clustering and initial fault judgment. Then PCA decomposition was performed for the error terms that followed gaussian distribution, and corresponding statistics were constructed to monitor them. In this paper, the effectiveness of the robust ICA-PCA method is demonstrated by comparing with the traditional PCA algorithm.

D. *Model prediction update*

In the actual production of process industry, timeliness is highly required. If the data mining knowledge takes too long, even if the production knowledge is mined, it will be worthless. Therefore, it is necessary to shorten the mining time as much as possible and extract knowledge quickly and effectively. However, association rule mining is the most time-consuming in the whole process of knowledge discovery, so this paper adopts FP-growth algorithm.

Compared with Apriori algorithm, it needs frequent search in the original data set, and the efficiency of the algorithm is relatively low. However, FP-growth algorithm uses tree structure and only needs to search the original data set twice without generating intermediate candidate sets, which greatly reduces the retrieval times, thus shortening the mining time and improving the efficiency of the algorithm.

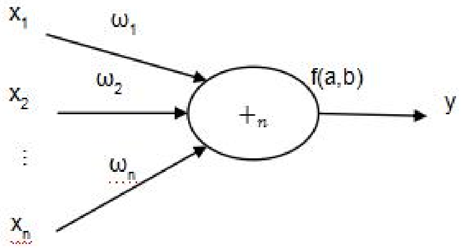

Fig. 1. FNT neuron operator

Initialize and continuously improve the FNT mathematical model according to the preset set of operators and information sets. As shown in Fig. 1., the steps of FNT model generation are as follows: firstly, a flexible neural tree and corresponding structural parameters are randomly initialized. Secondly, the tree structure and parameters are improved by *particle swarm optimization* (PSO), *probability enhancement program evolution* (PIPE) and other algorithms to generate a new flexible neural tree, and then the tree structure and various parameters are improved again. Such iterations continue until the set threshold value is reached, that is, an optimal model is obtained. The flexible excitation function is defined as follows:

$$\mathrm{f}(\mathrm{a_i},\mathrm{b_i},\mathrm{x})=\mathrm{e}^{-(\frac{\mathrm{x}-\mathrm{a_i}}{\mathrm{b_i}})^{2}} \qquad (8)$$

The input formula of flexible node $+_{\mathrm{n}}$ is defined, and the excitation sum is as follows:

$$net_n=\sum_{m=1}^{n}\omega_m\times x_m \qquad (9)$$

Flexible neural tree adopts special tree instruction set, including function instruction set F and leaf instruction set T, as follows:

$$I=F\cup T=\{+_2,+_3,\ldots,+_n\}\cup\{x_1,x_2,\ldots,x_n\}(10)$$

By using the above function formula, the key links that need to be adjusted, as well as the corresponding target parameters (attributes) and input parameters are firstly identified. Using the multi-layer tree structure in flexible neural tree. Input parameters are passed in the input layer, and the internal neurons, after excitation and calculation, output the expression of target parameters in the output layer. This expression can predict the change of boiler parameters in the future.

IV. Experimental analysis

In this section we present some experimental examples with synthetic and real data sets to show the performance of the proposed G-FCM algorithm. The comparisons between G-FCM, FCM, RL-FCM are also made, by using fuzziness index is 2.

The Seeds data set consists of 210 instances and 7 attributes, divided into three clusters. G-FCM and RL-FCM obtain correct cluster number with *accuracy rate* (AR) is 0.8952 and 0.8857, respectively. Using proper parameter selections, FCM gives three clusters with AR = 0.8952, respectively. We next make more comparisons of the proposed G-FCM with RL-FCM and FCM. The AR measured by changing the number of clustering centers given is shown in the Table I.

TABLE I. ACCURACY COMPARISON OF SEED DATA SETS.

Algorithm	Number of center	AR
G-FCM	2	0.7530
	3	0.8952
	4	0.7321
RL-FCM	2	0.7652
	3	0.8857
	4	0.7465
FCM	2	0.7325
	3	0.8952
	4	0.7512

Next, we will take two points around the number of correct clustering centers and calculate their average accuracy. The experimental results are shown in Table II. From the above experimental results, we can know that G-FCM and RL-FCM are not affected by the number of clustering centers and can be self-adaptive to complete clustering. The results are not much different from the FCM with the correct parameters of the given point. In the case that the number of correct clustering centers cannot be determined immediately, G-FCM and RL-FCM is more robust than FCM.

In G-FCM, there two main innovations: the first is that all data points are divided into grids before clustering, which much reduces the cost of calculating the points' distance, and the thought of parameter adaptive is adopted all parts of G-FCM, which can avoid the influence by the initialization of key parameters of clustering.

TABLE II. COMPARISON OF AVERAGE ACCURACY

Date set	G-FCM	RL-FCM	FCM
Iris	0.8952	0.9067	0.6933
Breast	0.9356	0.9528	0.5356
Seeds	0.8756	0.8952	0.5952
Aggregation	0.8523	0.8632	0.6165
Car Evaluation	0.8465	0.8562	0.6235
Abalone	0.8752	0.8852	0.6126
Poker Hand85.3	0.8632	08632	0.6325
Query Analytics	0.8632	0.8423	0.5652
Workloads	0.8123	0.8535	0.6435

TABLE III. RUNNING TIME COMPARISON

Date set	G-FCM	RL-FCM
Car Evaluation	4.125	20.256
Abalone	4.525	21.235
Poker Hand85.3	4.985	21.562
Query	4.862	20.632
Analytics Workloads	4.996	21.263

TABLE III compares the alarm time, alarm threshold and statistical amplitude of the two faults of the three algorithms respectively. PCA method has poor detection ability and correspondingly slow response. Ica-pca method not only improves the corresponding sensitivity, but also improves the amplitude range and magnifies the amplitude of the fault. Good monitoring effect has been achieved for both random type faults and step-type faults. Therefore, it can be concluded from the results of TABLE IV that the robust ICA-PCA method used in this chapter has greatly improved the sensitivity.

TABLE IV SENSITIVITY CONTRAST

Methods	Statistic	Failure 1 Statistic| Adjustment	Failure 2 Statistic| Adjustment
PCA	T2	185|24.2	242|24.2
	SPE	170|9.6	424|9.6
ICA-PCA	I2	**160|1580.4**	**167|562.4**
	T2	**166|540.8**	**171|279.5**
	SPE	**168|385.7**	**174|385.7**

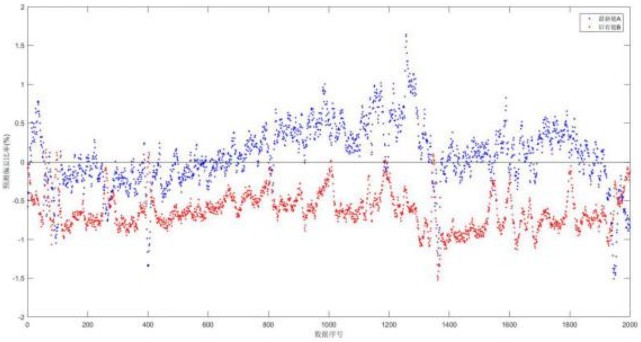

Fig. 2. Prediction deviation ratio diagram for main steam pressure left.

This paper compares the existing model (knowledge) at the time of concept transfer with the updated model. Test whether the new model can be more suitable for the current boiler operating conditions and predict state changes more accurately. For example, the material layer temperature B and main steam pressure were selected as the associated chain of the two target links (tail links) for experimental testing. We took the latest 2000 items (the latest data entering the window) from the 5W items in the sliding window and used them to calculate the comparison between the predicted data and the actual data in a recent period of time to further test the prediction effect, such as the prediction deviation ratio. *Normalized Mean Squared Error* (NMSE) is shown in Equation 5.5. The prediction deviation ratio is the deviation between the predicted value and the actual value. The normalized MEAN square error can comprehensively evaluate the accuracy of training model prediction. The value range of normalized mean square error itself is generally between 0 and 1. The smaller the value is, the better the prediction effect is, and the closer the predicted value is to the actual value. If the value is greater than 1, it means that the model prediction is better to simply take the average of all observations as the prediction. In order to observe the prediction effect of the model more vividly and specifically, the visualization image module of Matlab is used to generate trend prediction contrast diagram and prediction deviation ratio diagram, so as to observe the prediction curve more intuitively.

$$NMSE = \frac{\sum_{i=1}^{n}(\widehat{X_n} - X_n)^2}{\sum_{i=1}^{n}(X_n)^2} \qquad (11)$$

For example, the newly generated latest correlation chain A takes main steam pressure on the left as the target link, and primary air flow water supply pressure on the left main steam flow secondary air flow flue gas oxygen-containing air pack pressure on the left main steam pressure. The comparison diagram of trend prediction obtained through Matlab visualization is shown in Fig. 2.

With the rapid development of information technology [29], big data has been flooded in all aspects of our life [30], and all walks of life are producing massive data at all times. Hidden in the data is important information.[31] Especially in the process of industry production, production data contains hints of future working conditions, and these rapidly growing data are characterized by complex and changeable attributes, rapid update of objects, and value attenuation over time [32]. This paper focuses on the realization of real-time state and the improvement of fault detection efficiency. This article is an object in the research of the existing process knowledge discovery algorithm on the basis of the real-time state of improvement, design the process object oriented real-time fault detection method, and a robust fuzzy clustering for large data, the method of using Spark computing framework for parallel speed up the whole algorithm, developed a fuzzy clustering process industry fault prediction system. In this way, the real-time data stream can be predicted quickly and efficiently, and the timeliness and application value of fault detection can be improved.

V. CONCLUSION

This paper mainly studied the production state data of the process industry. These industrial data streams themselves are structured data with standard data types, but there are many discrete states of data, and the value density is low and difficult to mine. According to the characteristics of these data, this paper carried out online fault detection on the industrial data stream collected in real time and prompted the impending faults in real time. And the visual prediction of the future working conditions. Compared with the existing data mining methods of process industry, the real-time fault prediction algorithm designed in this paper jumped out of the offline mode and joins the online state, solved the limitation of the traditional industrial fault prediction method and can predict the occurrence of faults more effectively and achieve good prediction effect.

REFERENCES

[1] Wang J, Chen Q, Gong H. STMAG: A spatial-temporal mixed attention graph-based convolution model for multi-data flow safety prediction[J]. Information Sciences, 2020, 525:16-36.

[2] Z. Lu, N. Wang, J. Wu, M. Qiu, “IoTDeM: An IoT Big Data-oriented MapReduce performance prediction extended model in multiple edge clouds,” Journal of Parallel and Distributed Com., 118, 316-327, 2018

[3] L. Qiu, K. Gai, M. Qiu, “Optimal big data sharing approach for tele-health in cloud computing,” IEEE SmartCloud, 184-189, 2016

[4] M. Liu, S. Zhang, et al., “H infinite State Estimation for Discrete-Time Chaotic Systems Based on a Unified Model,” IEEE Trans. on Systems, Man, and Cybernetics (B), 2012

[5] F. Hu, S. Lakdawala, Q. Hao, M. Qiu, “Low-power, intelligent sensor hardware interface for medical data preprocessing,” IEEE Transactions on Information Technology in Biomedicine 13 (4), 656-663, 2009

[6] G. Wu, H. Zhang, et al. “A decentralized approach for mining event correlations in distributed system monitoring,” JPDC, 73(3), 2013

[7] M. Qiu, D. Cao, H. Su, K. Gai, “Data transfer minimization for financial derivative pricing using Monte Carlo simulation with GPU in 5G,” Int’l J. of Comm. Sys., 29 (16), 2364-2374, 2016

[8] M. Qiu, J. Liu, J. Li, et al., "A novel energy-aware fault tolerance mechanism for wireless sensor networks", IEEE Conf. GCC. 2011

[9] J. Niu, Y. Gao, M. Qiu, Z. Ming, "Selecting proper wireless network interfaces for user experience enhancement with guaranteed probability", JPDC, 72(12), 1565-1575, 2012

[10] J. Wang, M. Qiu, B. Guo, “High reliable real-time bandwidth scheduling for virtual machines with hidden Markov predicting in telehealth platform,” Future Generation Com. Syst. 49, 68-76, 2015

[11] Wang F, Zhang W, Guo H, et al. Automatic Translation of Data Parallel Programs for Heterogeneous Parallelism Through OpenMP Offloading[J]. The Journal of Supercomputing, 2020.

[12] Dai H, Li H, Shang W, et al. Logram: Efficient Log Parsing Using n-Gram Dictionaries. IEEE Trans. on Software Eng., 2020, PP(99):1-1.

[13] Dambros J, Trierweiler J O, Farenzena M. Oscillation detection in process industries – Part I: Review of the detection methods[J]. Journal of Process Control, 2019, 78:108-123.

[14] Y. Li, Y. Song, L. Jia, et al., “Intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble learning", IEEE TII, 17 (4), 2833-2841, 2020

[15] H. Qiu, T. Dong, T. Zhang, J. Lu, G. Memmi, M. Qiu, “Adversarial attacks against network intrusion detection in iot systems,” IEEE Internet of Things Journal 8 (13), 10327-10335, 2020

[16] H. Qiu, K. Kapusta, Z. Lu, M. Qiu, G. Memmi, “All-Or-Nothing data protection for ubiquitous communication: Challenges and perspectives,” Information Sciences 502, 434-445, 2019

[17] Sander J. Density-Based Clustering[M]. Springer US, 2011.

[18] Schikuta E. Grid-Clustering: An Efficient Hierarchical Clustering Method for Very Large Data Sets[J]. IEEE, 1993

[19] Miin-Shen, Yang, Yessica, et al. Robust-learning fuzzy c-means clustering algorithm with unknown number of clusters[J]. Pattern Recognition, 2017.

[20] Qiao N, Liang D, Chen Y, et al. Study on Fault Diagnosis and Troubleshooting Method of Gearbox Noise. Mechanical Management and Development, 2020.

[21] Mazouji R, Khaloozadeh H, Arasteh M, Fault Diagnosis of Broken Rotor Bars in Induction Motors Using Finite Element Analysis, 11th Power Electronics, Drive Systems, and Tech. Conf. (PEDSTC). 2020

[22] Frank P.M. Analytieal and qualitative model-based fault diagnosis-A Survey and some new results[J]. European J. of Control,1996,2:6-23.

[23] M. Qiu, H. Li, E. Sha, "Heterogeneous real-time embedded software optimization considering hardware platform", ACM sym. on Applied Comp., 1637-1641, 2009

[24] M. Qiu, M. Guo, M. Liu, et al., "Loop scheduling and bank type assignment for heterogeneous multi-bank memory", Journal of Parallel and Distributed Computing, 69 (6), 546-558, 2009

[25] M. Qiu, J. Niu, L. Yang, X. Qin, S. Zhang, B. Wang, “Energy-aware loop parallelism maximization for multi-core DSP architectures,” IEEE/ACM Conf. GCC, 2010

[26] H. Qiu, Q. Zheng, et al., "Topological graph convolutional network-based urban traffic flow and density prediction", IEEE ITS, 2020

[27] M. Qiu, K. Gai, et al., "Privacy-preserving wireless communications using bipartite matching in social big data", FGCS, 87, 772-781,2018

[28] H. Qiu, M. Qiu, M. Liu, G. Memmi, "Secure health data sharing for medical cyber-physical systems for the healthcare 4.0", IEEE journal of biomedical and health informatics 24 (9), 2499-2505, 2020

[29] M. Qiu, C. Xue, Z. Shao, E. Sha, “Energy minimization with soft real-time and DVS for uniprocessor and multiprocessor embedded systems,” IEEE DATE Conf., 1-6, 2007

[30] M. Qiu, K. Zhang, M. Huang, “Usability in mobile interface browsing,” Web Intelligence and Agent Systems, 4 (1), 43-59, 2006

[31] M. Qiu, C Xue, et al., "Efficient algorithm of energy minimization for heterogeneous wireless sensor network", IEEE EUC, 25-34, 2006

[32] X. Wei, H. Guo, X. Wang, et al., Reliable Data Collection Techniques in Underwater Wireless Sensor Networks: A Survey, IEEE Comm. Surveys & Tutorials 24 (1), 404-431, 2021