



Adaptive industrial control data analysis based on deep learning

Caihong Zhang¹ · Shengxiao Niu¹

Received: 19 September 2022 / Revised: 6 February 2023 / Accepted: 13 February 2023 / Published online: 20 March 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

The fault detection and diagnosis of industrial production process is of great significance to the reliability and safety of modern industrial systems. The data-driven fault detection and diagnosis method can perform statistical analysis and feature extraction on massive industrial control data, and divide the state of the system into normal operation state and fault state. It has attracted extensive attention from academia and industry. In order to achieve accurate and fast identification of industrial production faults, this paper proposes an adaptive multi-task deep learning fault detection model. The model uses an adaptive reweighting module and an improved adaptive pooling method to improve the process of label information acquisition and feature extraction, and improve the performance of the model. To evaluate the method proposed in this paper, the Tennessee Eastman (TE) system, a standard test industrial process in the chemical industry, was selected. The comparison results with the existing work show that the method studied can effectively process industrial control data and can be applied to monitor the occurrence of complex faults in industrial production processes.

Keywords Industrial Control Data · Industrial production process · Fault Detection · Adaptive deep learning · Adaptive pooling algorithm

1 Introduction

With the rapid development of modern industry, the degree of automation and complexity of industrial production process is getting higher and higher. While large-scale industrial processes bring huge economic benefits to the country, due to the high coupling and complexity of the system, any small disturbance in the system may lead to the paralysis of the entire system, resulting in huge economic losses and even casualties. Therefore, fault detection and diagnosis technology is indispensable for all industrial processes, and it is the guarantee that the industrial process can operate reliably and safely according to the production plan. However, complex industrial processes are inherently nonlinear, dynamic, multi-modal, multi-period, high-dimensional, intermittent and other characteristics, making fault detection and diagnosis extremely challenging, and traditional methods are difficult to adapt to the complexity of actual industrial processes. At the same time, as a large number of

new instruments, networked instruments and sensing technologies are applied to the entire manufacturing process, new variables are brought to fault detection and diagnosis. How to adopt effective fault detection and diagnosis methods to ensure the stable operation of industrial processes is an urgent and challenging problem.

In modern industrial production systems, failures are manifested when the monitored parameters of the process deviate from the expected range. According to the existing research, fault detection and diagnosis techniques are divided into three categories: model-based methods, knowledge-based methods and data-based methods. However, the model- and knowledge-based methods are limited by the difficulty of modeling and knowledge accumulation, and cannot be applied to dynamic and complex industrial production process systems. With the rapid development of distributed control, communication technology and data collection technology, a large number of observed variables in the system can be collected and stored, which promotes the research of data-based fault detection and diagnosis methods.

The data-based fault detection and diagnosis method can perform statistical analysis and feature extraction on massive industrial data, and divide the state of the system

✉ Shengxiao Niu
niusxiao@163.com

¹ Handan Polytechnic College, Handan 056001, China

into normal operation state and fault state, which can be regarded as a pattern recognition task. Fault detection is to judge whether the system is in the expected normal operating state and whether the system has abnormal faults, which is equivalent to a binary classification task. Fault diagnosis is to determine which fault state the system is in when a fault occurs, which is equivalent to a multi-classification task. Therefore, the research of fault detection and diagnosis technology is similar to pattern recognition, which is divided into 4 steps: data acquisition, feature extraction, feature selection and feature classification. (1) The data acquisition step is to collect signals from the process system that may affect the process state, including process variables such as temperature and flow rate; (2) The feature extraction step is to map the collected raw signals into recognizable system state information; (3) The feature selection step is to extract variables related to state changes; (4) The feature classification step is to perform fault detection and diagnosis on the features selected in the previous steps through an algorithm. In the context of big data, traditional data-based fault detection and diagnosis methods are widely used. However, these methods share some common disadvantages: feature extraction requires a lot of expert knowledge and signal processing techniques. At the same time, for different tasks, there is no unified procedure to complete them. Furthermore, conventional machine learning-based methods are shallow in structure and limited in ability to extract high-dimensional nonlinear relationships of signals.

With the rapid development of computer technology, especially the improvement of computing power, deep learning has received more and more attention. Deep learning originated from the study of artificial neural networks. Compared with shallow neural networks, the “depth” of deep learning is reflected in the network structure. Deep learning can model complex data and mine hidden features in data by building deeper structures, scalable hidden units and nonlinear activation functions. The learned features are usually deepened layer by layer, and high-level features are more abstract than low-level features and have stronger feature expression capabilities. At present, deep learning is widely used in the fields of computer vision [1, 2], natural language processing [3, 4] and object detection [5, 6]. In the field of fault detection and diagnosis in the industrial production process, based on auto encoder (AE), deep belief network (DBN), convolutional neural network (CNN) and recurrent neural network (recurrent neural network) neural networks, RNN) deep learning methods are widely used. The advantage of deep learning is that it can automatically perform feature engineering, and the whole process does not require manual intervention, which reduces the dependence of feature extraction on professional knowledge. In addition, the features learned by deep learning models are

deepened layer by layer, and deeper features perform better when used for prediction, detection, and classification tasks. Moreover, deep learning can learn in an end-to-end fashion, which means that the model is not limited to a specific task, so it can be adapted to different tasks by fine-tuning the network structure and parameters. That is, deep learning is easy to adapt to new problems and has higher robustness.

With the development of deep learning methods, deep learning algorithms have gradually replaced traditional methods of industrial control data analysis and processing. The multi-task deep learning (MTDL) method based on deep learning is also used to solve the classification and identification problem of fault detection in the industrial production process. Multi-task learning exploits the valuable information contained in multiple related tasks to improve performance on all related tasks [7]. Therefore, this paper proposes an adaptive multi-task deep learning method that introduces a multi-label system. Aiming at the problems of fuzzy or even loss of feature information extraction in common pooling algorithms, the proposed method improves the adaptive pooling method through the selection of parameters to improve the flexibility of feature extraction. To evaluate the performance of our method, we compare our method with existing work. The test results in the TE process show that the adaptive deep learning model proposed in this paper can effectively improve the fault detection effect of existing methods. The designed adaptive pooling algorithm can improve the flexibility of feature extraction and improve the detection performance of the model.

2 Related work

Data-driven fault detection can be viewed as a single-class anomaly detection task. The single-class anomaly detection method describes the normal data distribution area through statistical analysis, machine learning and other techniques, and samples outside this area are considered as abnormal data. One type of research chooses traditional machine learning methods such as support vector data description (SVDD) to achieve fault detection [8–10].

Another type of research uses deep learning methods to mine industrial control data for fault detection and diagnosis in complex industrial production processes. The powerful feature extraction ability of CNN solves the problem of insufficient artificial feature expression ability. Many scholars use CNN to study fault detection and diagnosis in chemical process. Wu et al. [11] proposed a fault diagnosis model based on a deep convolutional neural network, by stacking monitoring data of multiple time periods to form an input form of time dimension \times variable dimension. Similar to this idea, CNN-based fault diagnosis methods have been

applied to reactive distillation [12, 13], heat pump systems [14], and semiconductor manufacturing processes [15]. AE is an unsupervised learning algorithm that enables feature extraction on data without labels. Since the autoencoder (AE) is trained by unsupervised learning, the pure AE-based method cannot be applied as a classifier. Feature extraction can be performed through AE, and then a classifier such as a softmax layer or a Support Vector Machine (SVM) can be added after the model to achieve fault detection and diagnosis. References [16, 17] input the features extracted by SAE to the softmax layer to fine-tune the structural parameters in a supervised learning manner to achieve fault detection and diagnosis. Aiming at the time-domain and frequency-domain features of complex chemical processes, Lv [18] combined a stacked sparse auto encoder (SSAE) and an SVM classifier, and used SSAE to extract the correlation between monitoring variables and between samples. The temporal correlations are then employed for fault classification using the SVM classifier. Guo et al. [19] proposed a fault detection and diagnosis method based on SAE, which can extract the complete feature representation of incomplete data, retain the main features of the data in low-dimensional space, and implement fault diagnosis through SVM classifier.

DBN is a probabilistic generation model, which acts like an auto-encoder and can perform high-dimensional feature representation on the input data in the process of unsupervised learning, and DBN can add a classification layer at the end to perform supervised training on the data, which implements the classification algorithm. Wang et al. [20] optimized the DBN model through the DE algorithm. The optimized method has faster training speed, higher accuracy and more robustness. Ref. [21] proposed an Extended Deep Trust Network (EDBN), which considers the dynamic information of the data and combines the original data and hidden features to solve the problem that DBNs tend to lose valuable information in the original data. Wei et al. [22] adopted dropout technology to solve the overfitting problem of traditional DBN-based fault diagnosis methods, and the fault diagnosis rate of the proposed DBN-dropout method was higher than DBN and other deep learning-based methods. Yu et al. [23] proposed an unstable neuron-based DBN (UN-DBN) method, which first uses normal data to train a DBN model, and then obtains the result by integrating the hidden layers of unstable neurons in some samples. Feature representations that aid in fault detection. Tang et al. [24] proposed a DBN-based Fisher discriminant sparse representation method, which outperformed methods such as SVM and BP neural network in fault diagnosis performance in industrial processes with a large number of monitoring variables.

The above studies demonstrate the powerful adaptive feature extraction and classification capabilities of deep learning in solving industrial control data fault detection and analysis tasks. However, these studies are all used under the single-label system to diagnose single-target faults. In the context of big data, the single-label system not only separates the connection between different faults in industrial production process scenarios, but also makes it difficult to completely describe a wide variety of status information such as fault location, type, and degree.

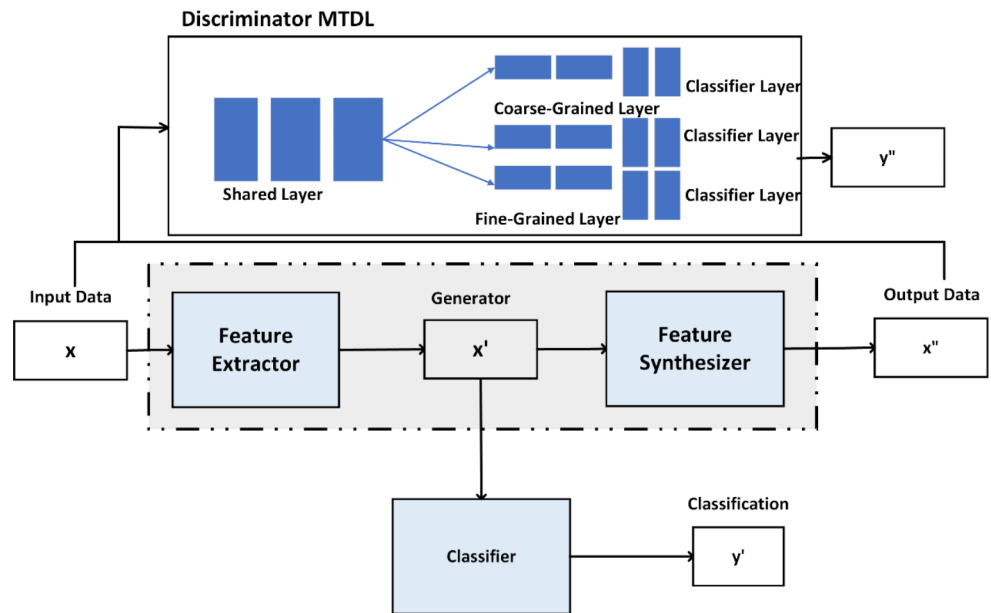
3 Adaptive deep learning method

Deep learning can automate feature engineering without human intervention. In addition, deep learning is easier to adapt to new problems and more robust. However, existing methods are often limited to single-label systems for fault diagnosis. In order to improve the accuracy of fault detection in industrial production process systems, this paper proposes an adaptive multi-task deep learning fault detection model, which can analyze and process industrial control big data to ensure the safe and reliable operation of industrial processes. The detection model mainly consists of two parts, namely the generator and the discriminator, where the generator consists of a feature extractor and a feature synthesizer, as shown in Fig. 1. Input data should be formatted as two-dimensional data. For feature extraction, an improved adaptive pooling method is proposed to improve the flexibility of feature extraction. And a Siamese neural network is designed to evaluate the local spatial distribution through an adaptive reweighting module and using the class label information with different confidence levels. At the same time, the discriminative MTDL algorithm is used in the discriminator to fully consider the local information of the sample. The structure of discriminative MTDL can be seen in the top module of Fig. 1. The first three convolutional layers in the discriminative MTDL network are shared layers, and the last three layers are coarse-grained layers and fine-grained layers.

3.1 Discriminative multi-task deep learning

Based on the discriminative MTDL method, this paper uses an adaptive reweighting module to discriminate with softmax loss to obtain local spatial distribution information from samples. In order to combine class label information and constraint loss, a Siamese neural network with two loss functions is designed, namely discriminative softmax loss and contrastive loss. The specific functions of these two loss functions are: (1) discriminative softmax loss, which uses class label information and local spatial distribution

Fig. 1 The Overall Workflow of Our Method



information of samples; (2) contrast loss, which helps to effectively prevent overfitting problems. Given a dataset X consisting of N samples from M different classes, the discriminative softmax loss can be expressed as

$$Sloss(x_i, \theta, y_i) = - \sum_{t=1}^M 1\{y_i == t\} \log \hat{p}_t = -\log \hat{p}_{y_i} = -\log \frac{e^{\theta_t^T x_i}}{\sum_{t=1}^M e^{\theta_t^T x_i}}$$

In the formula: x_i represents the network output corresponding to the training sample x^i . $1\{y_i == t\}$ is the indicator function, if $y_i == t$ is true, the result is 1, otherwise the result is 0. θ is the parameter of the network layer; θ_t is the weight of the t -th network output, $t = 1, \dots, M$. \hat{p}_{y_i} is the predicted probability.

The goal of discriminative MTDL is to exploit discriminative information from training samples. Therefore, an adaptive reweighting module ω_k is proposed to add local spatial distribution information to the proposed Siamese neural network. Additionally, a reweighting module that assigns weights to each sample based on the confidence between pairs of samples can both minimize the distance between similar samples and maximize the distance between different samples. Then the discriminative softmax loss function of this method is defined as:

$$SL(x_i, x_j, \phi_1, \phi_2, y_i, y_j) = \sum_{i,j,k} \omega_k (Sloss(x^i, \phi_1, y_i) + Sloss(x^j, \phi_2, y_j))$$

where: x_i and x_j are the network outputs corresponding to the training samples x^i and x^j . y_i and y_j are the class labels corresponding to x_i and x_j . ϕ_1 and ϕ_2 are the parameters of the two softmax losses. The adaptive reweighting

module ω_k improves the classification performance by providing specific local distribution information.

3.2 Adaptive reweighting

To evaluate the local spatial distribution of samples more effectively, an adaptive reweighting module is introduced in this paper. In order to design the adaptive reweighting module ω_k , high-level representations are considered to characterize samples of the same class, instead of samples of different classes. Therefore, $pCON(x_i, x_j)$ can be defined as follows:

$$pCON(x_i, x_j) = \sum_{x_j \in N_k(x_i)} \frac{\max(k - d(x_i), d(x_i, x_j))}{\max(k - d(x_i), d(x_i, x_i))}$$

where: $N_k(x_i)$ is the set of k nearest neighbors of x_i , and $\max(k - d(x_i), d(x_i, x_j))$ is the reachable distance from x_i to x_j . That is, if x_i and x_j are close enough, the reachable distance is $k - d(x_i)$; if x_i is far from x_j , the reachable distance is $d(x_i, x_j)$.

Since $pCON(x_i, x_j)$ is a probabilistic algorithm, the adaptive weighting module ω_k can be defined as:

$$\omega_k = G \left(\sum_{t=1}^T pCON(x_i, x_j) \right)$$

Where: T represents the number of iterations, G represents the transformation of the local Gaussian statistic, which can be used to scale the probability value. Discriminant MTDL iterations cannot have complex procedures. But can converge to similar results with sufficiently long iterations.

3.3 Adaptive pooling algorithm

3.3.1 Disadvantages of common pooling algorithms

Since the difference between different types of pooling algorithms only depends on the selected convolution kernel, the pooling algorithm is usually regarded as a convolution operation, and the feature matrix S after the pooling operation is:

$$S = \sum_{i=1, j=1}^c \gamma_{ij} x_{ij}$$

In the formula: γ_{ij} is the parameter of the convolution kernel. If it is an average pooling algorithm, take $\gamma_{ij} = \frac{1}{c^2}$, and c is the size and step size of the pooling domain. If it is maximum pooling, then set the maximum eigenvalue. The parameter is 1, and the others are set to 0.

If there are two different cases for the given pooling domain. For example, in the first case, the feature value of a region is Y , and the feature value of its adjacent parts is 0. Obviously, the feature information at Y is the most dense. Therefore, if the average pooling is used to process it at this time, the feature information will be greatly weakened. Similarly, if the eigenvalues of four adjacent locations in a certain area are 0, $Y1$, $Y2$, and $Y3$, respectively. If the feature information of $Y1$, $Y2$ and the correlation between $Y1$, $Y2$, and $Y3$ are directly ignored, and the maximum value $Y3$ is selected by the maximum pooling algorithm, a large amount of important feature information will be lost, which is not worth the loss. Therefore, whether the maximum pooling algorithm or the average pooling algorithm is used to aggregate the features of the pooling domain, it will weaken the representation of its global features.

In order to prevent the above problems from happening as much as possible, we try to use a random pooling algorithm, and after thinking about the correlation between $Y1$, $Y2$, and $Y3$, the final eigenvalues are selected according to the calculated probability values. However, the feature information in the final eigenvalues is not complete. It does not include all the feature information in $Y1$, $Y2$, $Y3$, and there are missing. And in the pooling layer, the eigenvalues obtained by pooling also need to be input into the neural network of the lower layer for calculation. It can be seen that the random pooling algorithm still has the problem of information loss, and cannot completely solve the problem in the second case above.

3.3.2 The principle of adaptive pooling algorithm

From the above analysis, we know that for the traditional pooling algorithm, the parameter values of the convolution

kernel are fixed during the training of the convolutional neural network. Also, there is no way to select appropriate parameters based on training samples and changes that occur during training. Therefore, this method will cause problems such as unclear feature information extracted and missing important feature information, and it is still not the best method [25].

It can imitate the autonomous learning ability of the human brain, which is the most important feature of artificial neural networks. Among them, the human brain mainly completes the process of autonomous learning by receiving external information through reading and listening. Similarly, the artificial neural network also has a learning process, which is a process of training the obtained sample set, and can also be understood as a search process for complex functional relationships. If the functional formula defined by the neural network can satisfy the training sample set, it means that the network model training is completed. On the contrary, the network model parameters are adjusted accordingly, and the process does not stop until the defined functional relationship can satisfy the training sample set [26].

Due to its own characteristics, the artificial neural network can continuously supervised training in a large number of training sets, so that the characteristics of the parameters can be modified many times, and finally it can be applied in various scenarios, and can perform automatic training. Adaptive dynamic correction of the convolution kernel parameters.

3.3.3 Implementation method

Based on the adaptive pooling algorithm, this paper improves it and proposes an adaptive improved pooling algorithm. The main steps of the algorithm are: (1) Initialize the parameters in the pooling, that is, initialize the convolution kernel parameter γ ; (2) During training, the pooling value is obtained after the convolution operation; (3) During training iterations, the parameter γ is updated by adopting gradient descent; (4) After multiple iterations and optimization of parameters, the adaptive pooling operation is finally completed. The process of the adaptive pooling algorithm is as follows: the first stage: the industrial control data is used as input, and after the mapping relationship of the discriminative MTDL, the predicted value of the fault category is finally output.

Stage 2: Calculate the difference between the two using a loss function based on the predicted value and the actual value obtained before. Among them, the two loss functions SVMLoss and SoftmaxLoss are the two most widely used functions in the training process of discriminative MTDL. This article uses SoftmaxLoss. Its expression is as follows:

$$S_{\text{loss}} = -\frac{1}{m} \left[\sum_{i=1}^m \sum_{j=1}^k 1\{y^i = j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{\theta_l^T x^{(i)}}} \right] + \frac{\lambda}{2} \sum_{i=1}^k \sum_{j=0}^k \theta_{ij}^2$$

Among them, the final predicted value is recorded as γ ; θ represents the weight between neurons; the indicative function is represented by $1\{\}$, if the expression in $\{\}$ is true, it takes 1, otherwise it takes 0; the weight decay term is represented by the second term in the formula.

In each training iteration, the parameters of the network model are appropriately adjusted based on the obtained loss value, so that the parameters of the loss function can be minimized, so that the predicted result can be closer to the actual value.

The third stage: Based on the loss function, the gradient descent method is used to update the pooling parameter γ to obtain the optimal parameter, that is, to find the minimum value of the loss function. The expression is as follows:

$$\gamma_{n+1} = \gamma_n - \varepsilon_n \cdot \nabla f(\gamma_n)$$

Among them, the step size and learning rate are represented by ε ; the loss function is denoted as $f(\cdot)$. The parameter values of the final pooling algorithm can be obtained after many iterations and updates, as well as continuous convergence to the extreme value of the loss function.

4 Experimental evaluations

4.1 Experimental settings

In this section, the proposed method is validated using an industrial process system in the chemical industry. In this section, the standard test industrial process Tennessee Eastman (TE) system is chosen to verify the performance of the proposed method. The TE system is derived from a real chemical industrial process [27] and mainly consists of five operating units: reactor, condenser, recycle compressor, separator and stripper. Reactants A, B, C, D and E enter the reactor to undergo an irreversible exothermic reaction, and the outlet product is cooled by a condenser and then enters a separator. The separated light component materials are returned to the reactor through the compressor, and the heavy component liquid of the separator flows into the stripper, and finally products G, H and by-product F are obtained. At present, the system has become a benchmark system for testing control and diagnostic methods, and has been widely used in many studies.

In the simulated data, a total of 41 observed variables are monitored, including 22 continuous process variables and 19 component variables. The TE process also includes 21 preset faults, and the first 20 faults are used for monitoring in

this case. Collect the data under normal working conditions and 20 faults in the TE process, and divide it into a training sample set and a test sample set. The training sample set contains 13,480 samples under normal working conditions and 480 samples under each fault. The test sample set contains 960 normal samples and 960 fault samples each. The fault samples are all in the fault state from the 161st sample. The process monitors 41 variables, so the training sample set forms a $23,080 \times 41$ matrix and the test sample set forms a $20,160 \times 41$ matrix.

The experimental model for discriminative MTDL consists of 4 convolutional layers, a fully connected layer and a softmax layer, where each convolutional layer is followed by a max pooling layer and a local response normalization layer. The features extracted from the fully connected layer will be input into the softmax layer for classification. In the experiment, the initial network learning rate is set to 0.05. Regarding the fault detection performance of the proposed method, the fault detection rate (FDR) is mainly used here to evaluate, and its specific calculation formula is as follows:

$$\text{FDR} = \frac{\text{Total number of samples detected}}{\text{Total number of samples of fault}} \times 100\%$$

4.2 Comparative experimental results

The performance of our method is evaluated by comparing it with several popular fault detection schemes. The methods involved in the comparison are: SVDD method, BP neural network, Attentive Dense CNN [28], and deep model without Adaptive Pooling Algorithms (MTDL without APA). The experimental results are shown in Fig. 2. The experimental results are displayed in a bar graph, and each group represents the experimental results of one method. In each set of results, the first bar shows the average fault detection rate for the 20 faults selected in this paper. Because the detection results of FDR vary greatly among different fault types. The figure also shows the FDR for two fault types, corresponding to the 1st fault and the 19th fault, respectively. From the perspective of average FDR, the SVDD algorithm works the worst. BP-NN is significantly improved on the basis of SVDD, which indicates that deep learning algorithms have advantages in processing industrial control data. The MTDL without APA method has a 4.07% lower FDR than the Attentive Dense CNN method. But after adding adaptive pooling algorithm, the performance of our method exceeds that of Attentive Dense CNN. This shows that the adaptive module can improve the detection performance of the whole model. For the FDR of fault type 1, all methods can achieve good detection results, and the FDR exceeds 97%. However, for fault type 19, the FDR of

Fig. 2 Experimental Results on FDR of the Compared Methods

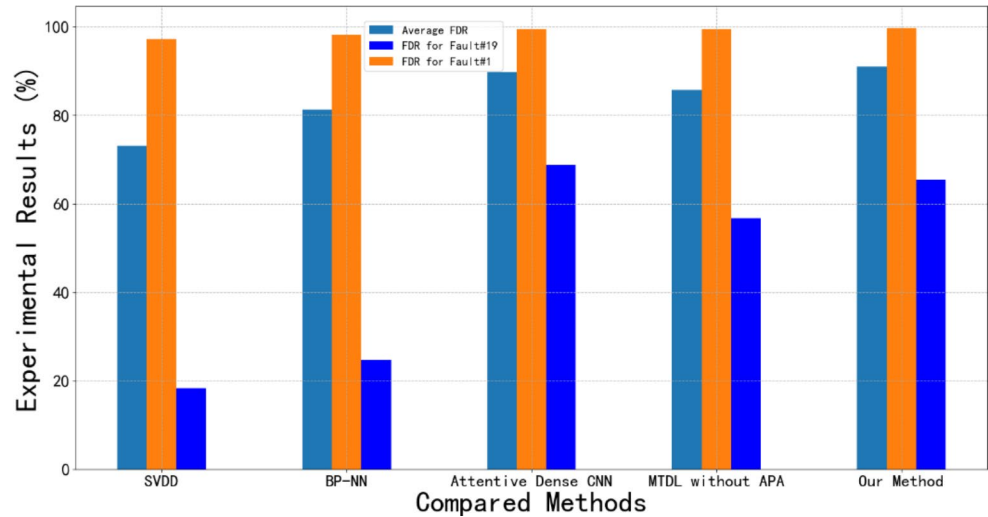
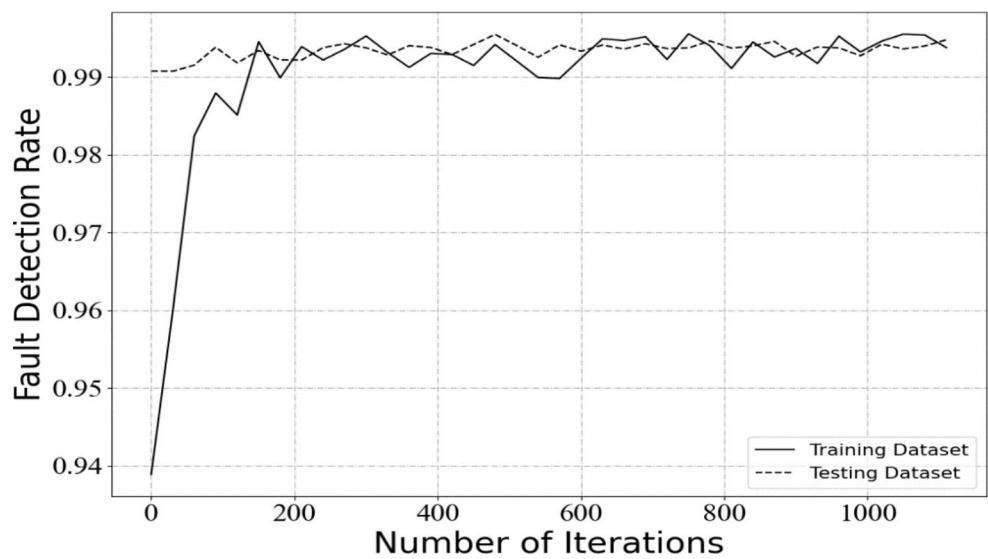


Fig. 3 Fault Detection Rates of Fault#1 on Training and Testing Datasets Over Iterations



SVDD and BP-NN is below 30%. The FDR of our method in fault type 19 is more than 3% lower than that of the Attentive Dense CNN method. Although the method proposed in this paper is not optimal for some fault types, the method in this paper still has the best performance among the methods involved in the comparison.

4.3 Experimental results over number of iterations

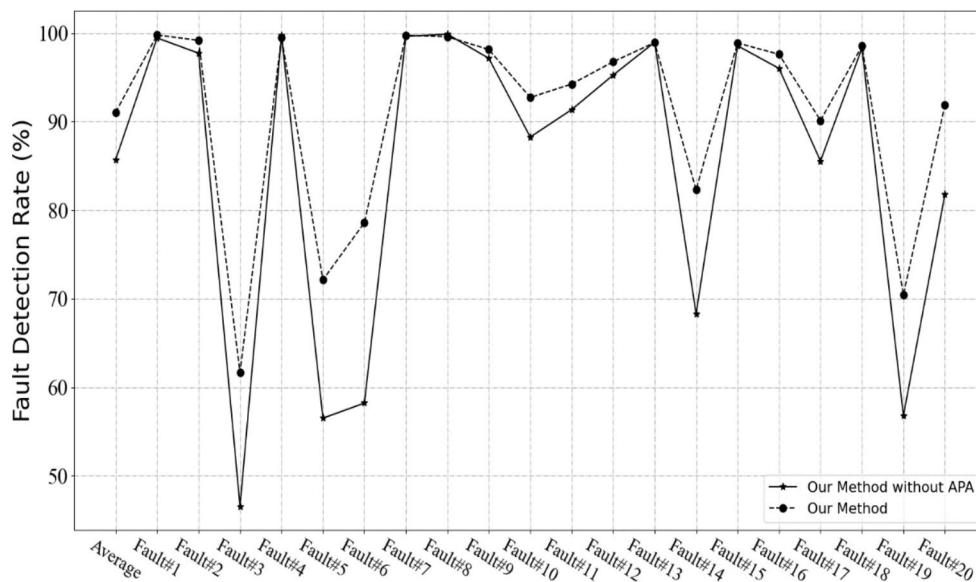
The Fig. 3 shows the accuracy of our method on fault 1 as a function of the number of iterations. Among them, the solid line represents the iteration results on the training dataset, and the dotted line represents the iteration results on the test dataset. It can be seen that the iteration of the method in this paper is relatively fast, and the detection rate can exceed 99% in 150 iterations, and gradually converge. Although the experimental results on the test dataset fluctuate, they all maintain a detection rate of more than 99%. This shows that

the model trained in this paper can be applied to the test data set, and there is no overfitting on the training data set.

4.4 Findings and further analysis

In this Section, the experimental results are further analyzed. Section 4.1 compares the average fault detection rate of the method in this paper and the Discriminant MTDL method that does not use the adaptive pooling algorithm on the data set. It can be seen from the experimental results that after introducing the adaptive pooling algorithm proposed in this paper, the system has increased the fault detection rate by 5.34%. In order to further evaluate the improvement of experimental performance by introducing the adaptive pooling algorithm, Fig. 4 shows the detection results of this method on different error types when the adaptive pooling algorithm is used or not. An obvious result is that the fault detection rate increases in all detection categories

Fig. 4 Experimental Results on Different Fault with/without Adaptive Pooling Algorithm



after introducing the adaptive pooling algorithm. This shows that the adaptive pooling algorithm proposed in this paper can improve the fault detection performance. Another interesting experimental result is that the detection rate of the method in this paper varies greatly on different fault types, and the detection rate on some types of fault types is lower than 80%. This may limit the application of this method in industrial production control. Further work needs to be improved for specific types of failures. It can also be seen from the figure that for fault types with a relatively low detection rate, the introduction of the adaptive pooling algorithm improves system performance more significantly. This shows that the adaptive pooling algorithm proposed in this paper can improve the effect of deep multi-task learning fault detection models.

5 Conclusion

Fault detection and diagnosis of industrial processes is critical to the stability, safety and product quality of industrial systems. The use of industrial control big data to realize industrial production process monitoring and fault detection can effectively ensure production quality and efficiency. Among data-driven methods, deep learning methods are receiving more and more attention. In this paper, an industrial control data processing method based on an adaptive deep learning method is proposed to realize the classification and detection of fault types. The method can deal with the loss of local information in the spatial distribution by using both the class label information and the local information of the sample spatial distribution. Discriminant MTDL is based on a Siamese neural network with shared weights and aims to measure local distributions through an adaptive

reweighting module. At the same time, the improved adaptive pooling method is used to dynamically adjust the optimization parameters to improve the recognition accuracy. The test results in the TE process show that the adaptive deep learning model proposed in this paper can effectively improve the fault detection effect of existing methods. The designed adaptive pooling algorithm can improve the flexibility of feature extraction and improve the detection performance of the model. Although the method proposed in this paper shows good performance on the test data, there are still some issues worthy of further study. In the future, the following aspects will be considered: how to improve the deep network structure to further improve the accuracy of fault detection; In addition, the detection strategy needs to be optimized for specific types of faults to reduce the false negative rate.

Author Contribution Caihong Zhang contributes to the writing and data analysis; Shengxiao Niu contributes to the conception and methodology.

Funding This paper has not received any funding support yet.

Declarations

Conflict of Interest All authors declare that there is no conflicts of interest in this paper.

References

1. Sun X, Li XG, Li JF, Zhuo L (2017) Review on deep learning based image super-resolution restoration algorithms. *Acta Automatica Sinica* 43(5):697–709
2. Zhang QS, Zhu SC (2018) Visual interpretability for deep learning: a survey. *Front Inform Technol Electron Eng* 19(1):27–39

3. Ren X, Zhou Y, Huang Z, Sun J, Yang X, Chen K (2017) A novel text structure feature extractor for chinese scene text detection and recognition. *IEEE Access* 5:3193–3204
4. Xue-Feng X, Guo-Dong Z (2016) A survey on deep learning for natural language processing. *Acta Automatica Sinica* 42(10):1445–1465
5. Druzhkov PN, Kustikova VD (2016) A survey of deep learning methods and software tools for image classification and object detection. *Pattern Recognit Image Anal* 26(1):9–15
6. Han J, Zhang D, Cheng G, Liu N, Xu D (2018) Advanced deep-learning techniques for salient and category-specific object detection: a survey. *IEEE Signal Process Mag* 35(1):84–100
7. Yolcu G, Oztel I, Kazan S, Oz C, Palaniappan K, Lever TE, Bunyak F (2019) Facial expression recognition for monitoring neurological disorders based on convolutional neural network. *Multimedia Tools and Applications* 78(22):31581–31603
8. Zhao YP, Xie YL, Ye ZF (2021) A new dynamic radius SVDD for fault detection of aircraft engine. *Eng Appl Artif Intell* 100:104177
9. Wang B, Mao Z (2020) A dynamic ensemble outlier detection model based on an adaptive k-nearest neighbor rule. *Inform Fusion* 63:30–40
10. Wang K, Lan H (2020) Robust support vector data description for novelty detection with contaminated data. *Eng Appl Artif Intell* 91:103554
11. Wu H, Zhao J (2018) Deep convolutional neural network model based chemical process fault diagnosis. *Comput Chem Eng* 115:185–197
12. Ge X, Wang B, Yang X, Pan Y, Liu B, Liu B (2021) Fault detection and diagnosis for reactive distillation based on convolutional neural network. *Comput Chem Eng* 145:107172
13. Li C, Zhao D, Mu S, Zhang W, Shi N, Li L (2019) Fault diagnosis for distillation process based on CNN–DAE. *Chin J Chem Eng* 27(3):598–604
14. Eom YH, Yoo JW, Hong SB, Kim MS (2019) Refrigerant charge fault detection method of air source heat pump system using convolutional neural network for energy saving. *Energy* 187:115877
15. Lee KB, Cheon S, Kim CO (2017) A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes. *IEEE Trans Semicond Manuf* 30(2):135–142
16. Jiang P, Hu Z, Liu J, Yu S, Wu F (2016) Fault diagnosis based on chemical sensor data with an active deep neural network. *Sensors* 16(10):1695
17. Lv F, Wen C, Bao Z, Liu M (2016), July Fault diagnosis based on deep learning. In *2016 American control conference (ACC)* (pp. 6851–6856). IEEE
18. Lv F, Wen C, Liu M, Bao Z (2017) Weighted time series fault diagnosis based on a stacked sparse autoencoder. *Journal of Chemometrics*, 31(9), e2912
19. Guo C, Hu W, Yang F, Huang D (2020) Deep learning technique for process fault detection and diagnosis in the presence of incomplete data. *Chin J Chem Eng* 28(9):2358–2367
20. Wang Y, Zhang J, Deng F (2017), June A Method of Fault Diagnosis Based on DE-DBN. In *Chinese Intelligent Automation Conference* (pp. 209–217). Springer, Singapore
21. Wang Y, Pan Z, Yuan X, Yang C, Gui W (2020) A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network. *ISA Trans* 96:457–467
22. Wei Y, Weng Z (2020) Research on TE process fault diagnosis method based on DBN and dropout. *Can J Chem Eng* 98(6):1293–1306
23. Yu J, Yan X (2019) Whole process monitoring based on unstable neuron output information in hidden layers of deep belief network. *IEEE Trans Cybernetics* 50(9):3998–4007
24. Tang Q, Chai Y, Qu J, Ren H (2018) Fisher discriminative sparse representation based on DBN for fault diagnosis of complex system. *Appl Sci* 8(5):795
25. Momeny M, Jahanbakhshi A, Jafarnezhad K, Zhang YD (2020) Accurate classification of cherry fruit using deep CNN based on hybrid pooling approach. *Postharvest Biol Technol* 166:111204
26. Mohamed O, Khalid EA, Mohammed O, Brahim A (2017), October Content-based image retrieval using convolutional neural networks. In *First International Conference on Real Time Intelligent Systems* (pp. 463–476). Springer, Cham
27. Faris H, Sheta A (2013) Identification of the tennessee eastman chemical process reactor using genetic programming. *Int J Adv Sci Technol* 50:121–140
28. Plakias S, Boutalis YS (2020) Fault detection and identification of rolling element bearings with attentive dense CNN. *Neurocomputing* 405:208–217

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.