2016 American Control Conference (ACC)
Boston Marriott Copley Place
July 6-8, 2016. Boston, MA, USA

Fault Diagnosis Based on Deep Learning

Feiya Lv[a], Chenglin Wen[b], Zejing Bao[a], Meiqin Liu[a]

***Abstract*— As representation scheme can severely limit the window by which the system observes its world, deep learning for fault diagnosis is put forward in this paper. It is a real time online scheme that can enhance the accuracy of detection, classification and prediction, and efficient for incipient faults that cannot be detected by traditional statistic technology. A stacked sparse auto encoder is used to learn the deep architectures of fault data to minimize the loss of information. Experiment results show that the proposed method not only improves the divisibility between faults and normal process, but also exhibits a better performance on the accuracy of fault classification for the chemical benchmark, Tennessee Eastman Process (TEP) data.**

Keywords: fault detection, fault classification, deep learning, sparse auto encoding

I. INTRODUCTION

As accidents caused by the system's performance dissipation and external disturbance result in huge property losses and casualties, intelligent techniques are required to detect and identify faults in complex industrial process. Fault diagnosis is essential and widely applied in large and complicated modern industrial systems, especially in chemical production systems due to its particularity and potentially dangerous.

Distributed control system (DCS) is widely used in many industrial systems, however, its complex structure in data collection and storage may cause data overload, but information poor. So data-driven process monitoring methods is needed to process those massive data, statistical analysis methods have been successfully applied, such as principle component analysis (PCA) [1, 2, 3, 4], independent component analysis (ICA) [5, 6, 7]. However, information will be lost in linear dimension reduction, the detailed information about incipient fault maybe contained. Furthermore, as statistical methods always using different kinds of thresholds as detection standard, it may ignore the impact of incipient fault. Moreover, these linear discriminative is not suitable for complex process, some approaches have been developed to solve nonlinear problem, such as pattern classification on kernel fisher discrimination analysis subspace [8], non-negative matrix factorization with sparseness constrained (NMFSC) [9]. Data used in NMFSC is described in terms of basis W and encoding H with sparseness constrains, which is a parts-based representation and cannot make full use of the correlation between all variables.

Most recently, machine learning technology has been successfully used in fault diagnose to treat it as a discriminative problem. Support vector machine (SVM) [9, 10, 11] is the main research direction contributing by its excellent classification performance. Structural SVM can address the complementary issue of designing classification algorithms [9], and train a multi-class predictor directly. As a supervised learning method, SVM requires more labeled samples and suffers from data imbalance, this lead to higher cost and may be unrealistic in real industrial process. Besides, random forests [12] can be applied to the data set with unknown samples. As a promotion of decision tree, it is not affected by variable missing. However, it cannot give a full play to the superiority of the baggings when only few samples are available for model establishment. After all, improvements in machine learning technologies still cannot capture a flexible and powerful relation among data in industrial process due to its limitations of learning ability, according to the data's dynamic, large-scale, multi-scale, autocorrelation features etc.. Feature representation is a key step for fault diagnosis, sparse representation classification [13] collects training samples to build a dictionary, which can be used to calculate the sparsely reconstruction. However, it relies on the high differences between variables. In order to represent high-level abstractions by a kind of complicated functions, deep architectures are need, which is a difficult task until deep learning is proposed by Hinton [14].

Deep learning had formed a hot topic mostly in image application and object recognition, beat the state-of-the-art methods in certain areas due to its strong learning ability [15, 16]. It is suitable for large system with multiple variables, fault diagnosis, of course. However, deep learning for fault diagnosis had been paid less attention. This appears to be a challenging task, of three essential difficulties: (1) for images, the characteristics of recognition objects are relatively fixed, but faults are changeable, such as patterns variability and shape variability; (2) as fault has no fixed pattern, whether deep learning can capture a useful "hierarchical grouping" or "part-whole decomposition" of the fault data is unknown; (3) the detection mechanism and ability based on deep learning is not yet well explored, especially for the incipient faults not any observable changes, which is a bottleneck that traditional methods suffering.

Motivated by these, this paper aims at enhancing the accuracy of detection, classification and prediction by capturing the deep architectures of the fault data, and obtaining its distributed representations. The main contributions of this paper are outlined as follows: (1) a deep learning based fault diagnosis algorithm is put forward, which has the deep structure of multiple nonlinear mapping, and can complete complex function approximation; (2) the proposed frame not only suitable for common fault diagnosis, but also covers the detection of incipient fault; (3) the proposed method is a real time detection and classification, unlike

*Research supported by National Natural Science Foundation of China (U1509203, 61333005, 61490701, 61273170), and Zhejiang Provincial Natural Science Foundation of China under Grant LZ15F030001.

*C. H. Wen is with the Institute of Systems Science and Control Engineering, School of Automation, Hangzhou Dianzi University, Hangzhou, 310000, People's Republic of China. (corresponding author to provide phone: 13819461626, e-mail: wencl@hdu.edu.cn).

F. Y. Lv, Z. J. Bao, M. Q. Liu, all are with the Institute of Control Theory and Control Engineering, Zhejiang University, Hangzhou. (e-mail: lvfeiya0215@126.com).

traditional methods, it can do the diagnosis online without fault's history sample information; (4) a stacked auto encoder [15] is employed to learn robust fault features that cannot be detected by statistic techniques, which enjoys all the benefits of any deep network of greater expressive power.

To the author's best knowledge, researches of this on chemical production systems haven't been found. This paper makes up this blank and does experiments on TEP benchmark.

This paper is organized as follows. In section 2, the deep learning based fault diagnosis is put forward, consists of feature learning by stacked sparse auto encoder and fault classification by softmax classifier. Experiments on TEP are proposed in Section 3 to validate the effectiveness of our algorithm; finally, conclusions and future work are discussed in Section 4.

II. Fault Diagnosis Based on Deep Learning

Fault diagnosis consists of fault detection and fault classification. Before distinguishing which type the fault belongs to, we should detect whether there is a fault. To some extent, the fault detection can also be viewed as a binary classification problem. So we give a detail discussion about this classification problem use stacked sparse auto encoder neural network and softmax classifier.

A. Learning Fault Features by Stacked Sparse Auto Encoder

A stacked auto encoder is a neural network consisting of multiple layers of sparse auto encoders in which the outputs of each layer is wired to the inputs of the successive layer. By a stacked auto encoder with n layers, the deep architectures of the input fault data can be learned well. An auto encoder neural network is an unsupervised learning algorithm that transforms the input data into a different dimension code automatically by ones' personal need, which applies back-propagation to recover the data, setting the target values to be equal to the inputs.

Firstly, the encoding step for the stacked auto encoder is given by running the encoding step of each layer in forward order, and the decoding step is running the decoding stack of each auto encoder in reverse order [17]:

$$\begin{aligned} Encoding: \quad & a^{(l)} = f\left(z^{(l)}\right) \\ & z^{(l+1)} = W^{(l,1)} a^{(l)} + b^{(l,1)} \end{aligned},$$

$$\begin{aligned} Decoding: \quad & a^{(n+l)} = f\left(z^{(n+l)}\right) \\ & z^{(n+l+1)} = W^{(n-l,2)} a^{(n+l)} + b^{(n-l,2)} \end{aligned}.$$

The activation of the deepest layer $a^{(n)}$ is what we learned finally, which gives a representation of the input in terms of higher-order features and contains the information of interest. But this only happens when the parameters are good trained. As a deep network works, the greedy layer-wise approach is employed to train each layer in turn to pre-train all the net weights, which can train the parameters of each layer individually while freezing parameters for the remainder of the model.

After this, fine-tuning is taken by using back propagation to improve the results, which is tuning the parameters of all layers at the same time. In order to compute the gradients for all the layers in each iteration process, let

$$\delta^{(n_l)} = -\left(\nabla_{a^{n_l}} J\right) f'\left(z^{(n_l)}\right), \ \delta^{(l)} = -\left(\left(W^{(l)}\right)^T \delta^{(l+1)}\right) f'\left(z^{(l)}\right)$$

for $l = n_l - 1, n_l - 2, n_l - 3, \cdots, 2$, then compute the desired partial derivatives,

$$\nabla_{W^{(l)}} J(W,b;x,y) = a^{(l)} \delta^{(l+1)}, \quad \nabla_{b^{(l)}} J(W,b;x,y) = \delta^{(l+1)},$$

where $J(W,b) = \left[\frac{1}{m}\sum_{i=1}^{m} J(W,b;x^{(i)},y^{(i)})\right]$ is the cost function [17].

The features obtained by this stacked auto encoder can be used for classification problems by feeding $a^{(n)}$ to a softmax classifier. It is worth mentioning, if fine-tuning for the purposes of classification only, the gradients from the classification error will then be back propagated into the encoding layers. Therefore, the derivatives are computed by $\delta^{(n_l)} = -\left(\nabla_{a^{n_l}} J\right) f'\left(z^{(n_l)}\right)$, $\nabla J = \theta^T (I - P)$, I is the input labels and P is the vector of conditional probabilities.

B. Fault Classification by Softmax Classifier

Softmax classification is a supervised learning algorithm with training set $\left\{\left(x^{(1)}, y^{(1)}\right), \cdots, \left(x^{(m)}, y^{(m)}\right) \mid y^{(i)} \in \{1, 2, \cdots, k\}\right\}$, which used in conjunction with our deep feature learning methods. To a given input, our hypothesis is to estimate the probability $p(y = j \mid x)$ for each $j = 1, \cdots, k$. Thus, the hypothesis function $h_\theta(x)$ takes the form:

$$h_\theta\left(x^{(i)}\right) = \begin{bmatrix} p(y^{(i)} = 1 \mid x^{(i)};\theta) \\ p(y^{(i)} = 2 \mid x^{(i)};\theta) \\ \vdots \\ p(y^{(i)} = k \mid x^{(i)};\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}.$$

Specifically, weight decay should be incorporated to penalize large values of the parameters, so the cost function and its partial derivative are,

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

$$\begin{aligned} \nabla_{\theta_j} J(\theta) = & -\frac{1}{m}\sum_{i=1}^{m}\left[x^{(i)}\left(1\{y^{(i)} = y\} - p(y^{(i)} = j \mid x^{(i)};\theta)\right)\right] \\ & + \lambda\theta_j, \qquad \lambda > 0 \end{aligned}.$$

Thus the cost function $J(\theta)$ is strictly convex and the Hessian matrix is invertible, which guaranteed that $J(\theta)$ converges to the global minimum [17]. It is important to emphasize that softmax classification is used to predict which fault the input data belongs to after the fault had been detected. While k equals 2, the softmax classification reduces to a logistic regression, which can be used to distinguish whether the state is normal or not.

C. The Frame of Deep Learning Based Fault Diagnosis

The basic idea of deep learning-based fault diagnosis consists of, for a given unlabeled industrial samples, first do the unsupervised learning with two cascaded sparse coding network to learn a better representation, and then fine-tuning the network according to the categories with supervision, finally predict the fault in the case of binary classification and diagnosis in the case of multiple classification.

TABLE I. THE FRAME OF THE PROPOSED ALGORITHM.

Step 1. Data preprocessing: normalize the unlabeled samples $\{x^{(1)}, x^{(2)}, x^{(3)}, \cdots\}$ to $\{X1^{(1)}, X1^{(2)}, X1^{(3)}, \cdots\}$;
Step 2. Train the stacked sparse auto encoder, and update the every layer weight parameters by $\frac{\partial}{\partial W_{ji}^{(l)}} J_{sparse}(W,b)$, $\frac{\partial}{\partial b_{ji}^{(l)}} J_{sparse}(W,b)$, then get the nonlinear transform features after feed forward;
Step 3. Train the softmax classifier by the feature set in step 2, with its labels, and update its parameters by $\nabla_{\theta_j} J(\theta)$;
Step 4. Fine-tuning the network parameters by $\nabla_{W^{(l)}} J(W,b;x,y)$, $\nabla_{b^{(l)}} J(W,b;x,y)$;
Step 5. Output.

III. EXPERIMENT ON TENNESSEE EASTMAN PROCESS

On the basis of the theory given above, experiments are presented and analyzed on a public benchmark, TEP, to evaluate its effectiveness in this section. TEP has been widely used in fault diagnosis research, whose data can be downloaded from http://web.mit.edu/braatzgroup/links.html. 21 types of identified faults with each consist of 52 variables that are generated at a sampling interval of 3 min. Each faulty condition consists of 480 samples as training dataset. Meanwhile, 960 samples are as testing dataset with faults being induced after 8 h which corresponds to 160 samples.

A. Fault Detection

To verify the detection performance of the proposed method, experiments are configured into 2 cases in this subsection: comparing with linear supervised classification, comparing with nonlinear detection methods. As the normal operating condition in database has only 500 samples, we combine it with the first 100 samples in testing dataset as normal samples, and the first 20 faults training sets together as fault samples in this new training dataset. The 860 samples remaining in testing dataset are still test samples to recognize whether the fault happens or not.

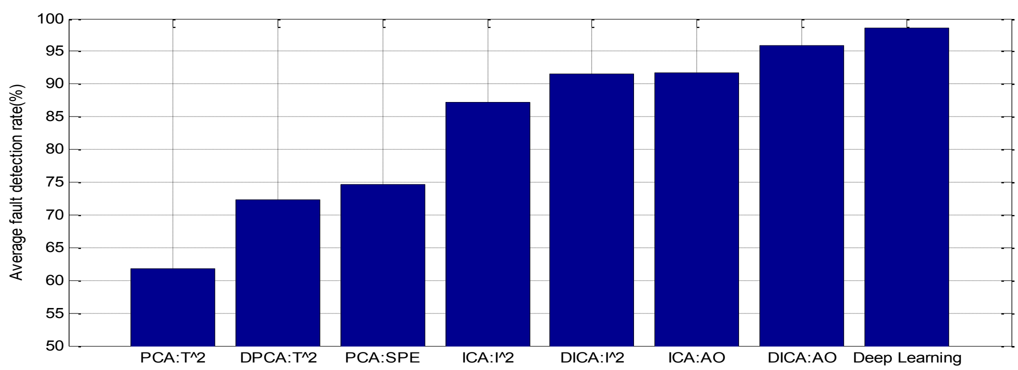

Figure 1. The average fault detection rates of several linear supervised binary classification methods.

a. comparing with linear supervised classification

Firstly, the performance of this algorithm will be compared with linear classification methods, PCA, ICA, DPCA, DICA [7]. The average fault detection rates on our test data are shown in Fig.1.

Obviously, average detection accuracies are improved based on deep learning method, 2.62% higher than DICA: AO [7]. The average rate of missed diagnosis is 1.5% over these 20 fault types, only 258 samples have not been detected. All detection results are shown in Table II. From it we can see, our method has the superior performance, especially for faults 10, 11, 16, 18, 20, the detection rate improved really evidently. And also the detection rate of deep learning method is consistently higher on all the faults, this mainly because of the deep architecture that learned can fully reveal the correlation between variables, not the important component of features only.

Then we discuss faults 3, 9, 15, 21 especially while the experiment analyzed above didn't conclude them. The references [18] and [19] have pointed out detecting faults 3, 9, 15, 21 are very difficult, the reason maybe that there are not any observable changes in the means, variance, or the peak time, i.e. they are incipient faults that cannot be detected by any statistics technique easily, but deep learning can do it. When the hidden units in the two hidden layers set as 200, 200, the dimensions of the new features learned are 200. Fig.2 shows the values of the 6th dimension. From this we can see the fault pattern is different with normal condition, so it's easy to identify the failure data with normal data through this dimension. And also we can get that different fault has different pattern, this can be used to diagnosis fault type.

TABLE II. THE DETECTION RATES OF DIFFERENT TYPES FAULT.

Fault	PCA (15 components)		DPCA (22 components)	ICA (9 components)		DICA (22 components)		DL
	T2	SPE	T2	I2	AO	I2	AO	
1	99.2	99.8	99	**100**	**100**	**100**	**100**	**100**
2	98.0	98.6	98	98	98	99	99	**99.75**
4	4.4	96.2	26	61	84	97	**100**	**100**
5	22.5	25.4	36	**100**	**100**	**100**	**100**	98.88
6	98.9	**100**	**100**	**100**	**100**	**100**	**100**	**100**
7	91.5	**100**	**100**	99	**100**	**100**	**100**	**100**
8	96.6	97.6	98	97	97	**98**	**98**	97.88
10	33.4	34.1	55	78	82	82	90	**99.25**
11	20.6	64.4	48	52	70	54	83	**89.25**
12	97.1	97.5	99	99	**100**	**100**	**100**	99.75
13	94.0	95.5	94	94	95	95	96	**99.75**
14	84.2	**100**	**100**	**100**	**100**	**100**	**100**	95.13
16	16.6	24.5	49	71	78	82	91	**99.5**
17	74.1	89.2	82	89	94	90	96	**99.75**
18	88.7	89.9	90	90	90	90	90	**99.5**
19	0.4	12.7	3	69	80	81	95	**96.75**
20	29.9	45.0	53	87	91	88	92	**99.38**

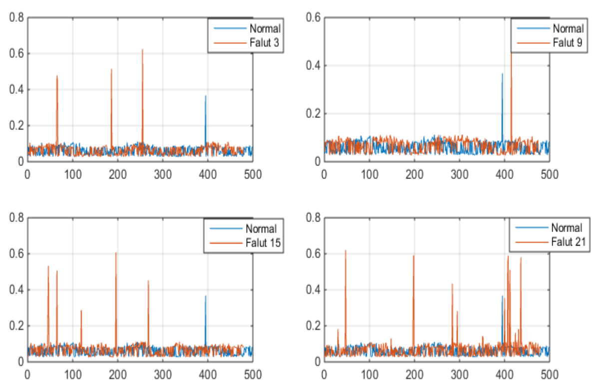

Figure 2. The 6th dimension values of faults 3, 9, 15, 21after learning.

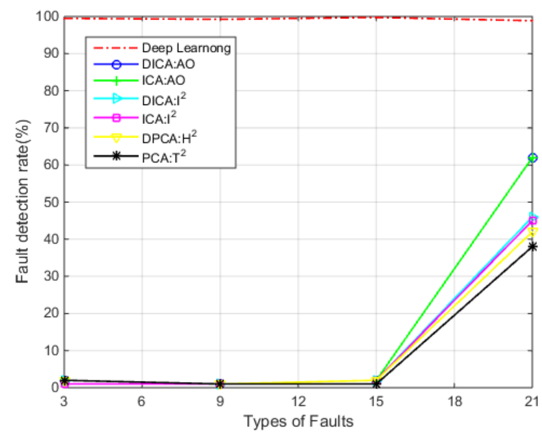

Figure 3. The rates of fault detection for faults 3, 9, 15, 21.

Fig.3 shows a revolutionary change as the performance improvement is very large, fundamentally solved the detection problems of these four kinds of faults.

b. comparing with nonlinear detection methods.

The performance of this algorithm will be compared with NMFSC, SVM respectively. The average fault detection rates on our test dataset are shown in Fig.4. As a binary classification, our algorithm has achieved ideal results. Unlike NMFSC, deep learning method can learn the data's structural and distributed characteristics, and fully tap the local features and global features. Though better detection rate SVM also achieved, it has no deep excavation for the potential characteristics of the data.

B. Fault Classification

After the fault is detected, identify its type is the next step. Formed as a multi-class classification problem, our model can train multiple faults data for only one uniform discriminative model, unlike the "one-against-one" or "one-against-all" strategies in traditional SVM.

TABLE III. TYPES OF FAULT 4, 9, 11AND 2, 10, 13, 14.

Fault	Description	Type
4	Reactor cooling water inlet temperature	Step
9	D feed temperature (stream 2)	Random variation
11	Reactor cooling water inlet temperature	Random variation
2	B composition, A/C ratio constant (stream 4)	Step
10	C feed temperature (stream 4)	Random variation
13	Reaction kinetics	Slow drift
14	Reactor cooling water valve	Sticking

As fault detection is only a binary classification, in order to verify the effectiveness of our algorithm on multi-class, we do experiments on faults 2, 4, 9, 10, 11, 13 and 14 in Table III first. For faults 4, 9 and 11, their faulty variables are overlapping when plot against each other, consequently are not easy to classify. Table IV gives their confusion matrixes. This clearly clarifies how the discrimination of the different fault is done. For example, 746 in the first row shows the number of right predict retrieval samples for fault 4, and 7 is the misclassification number that divided fault 4 as fault 9. Each column represents the instance in an actual class. For faults 2, 10, 13, 14, they cover all fault types that are difficult to distinguish. The diagnosis confusion matrixes obtained are listed in Table V, which show that our algorithm performs well. The rate of false alarms is 8.56%.

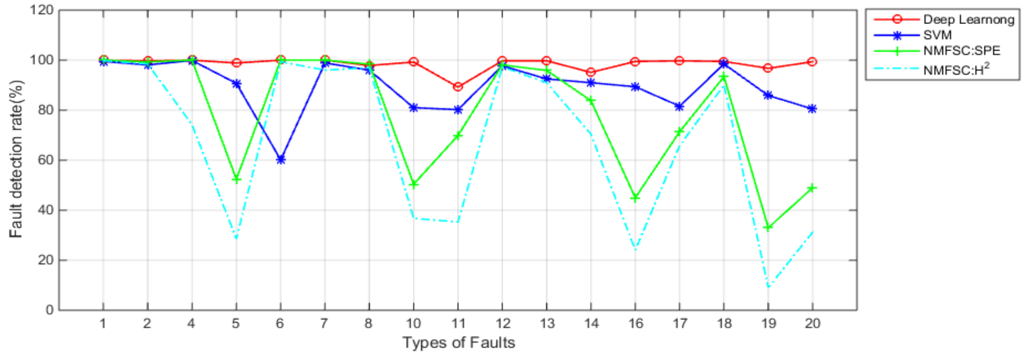

Figure 4. The average detection rates of different types with several nonlinear binary classification methods.

TABLE IV. THE CONFUSION MATRIXES FOR FAULT 4, 9, 11.

Fault	4	9	11
4	**746**	2	132
9	7	**652**	156
11	47	146	**512**
Total	800	800	800

TABLE V. THE CONFUSION MATRIXES FOR FAULT 2, 10, 13, 14.

Fault	2	10	13	14
2	**786**	1	47	0
10	13	**779**	175	0
13	1	20	**561**	0
14	0	0	17	**800**
Total	800	800	800	800

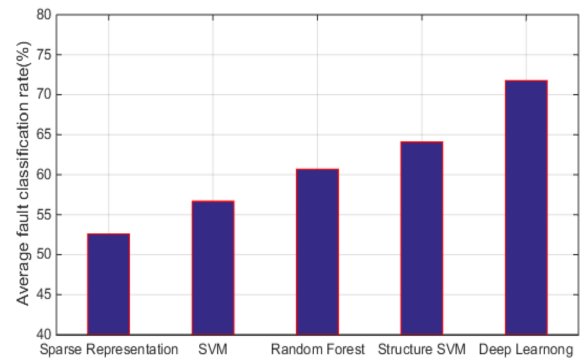

Figure 5. The average fault classification rates of 5 methods.

Remark: the reporting classification rates in Fig.5 are from references [14], [13], [12], [10] accordingly, but all these study are under the exact same case.

Fig.5 shows the average fault classification rates on test data based on stacked sparse auto encoder and other state-of-art approaches, sparse representation [14], random forests [13], SVM [12], structural SVM [10]. Average classification accuracies have improved 7.67% based on our method, obviously higher than the other methods.

Furthermore, the classification rates of all the faults are shown in Fig.6, all these methods' general trend of fault diagnosis on test set are roughly consistent. For faults 1, 2, their variables significantly deviated from the normal state, all these methods can have higher diagnostic rate, but for faults have no obvious change with normal state or variables are overlapping, our method has a relatively high diagnostic rate.

C. Computational Cost

The experiments in this paper have been implemented on an AMD Phenom Quad-Core 9750 PC (2.4GHz, 4GB RAM) 32Bit Windows 7 operating system in Matlab environment. The average execution time is 0.0035s, which is real time detection and classification. This mainly because the proposed method can do the diagnosis just by the current sample data after the parameters are trained offline, not needs to analyze the history data.

The advantage of deep learning is in processing large-scale database, for TEP has only 52 variables, 2 layers of sparse coder is enough, our simulation experiments show that 3 layers cannot significantly improve the accuracy, but the complexity increased. Our proposed methods are competitive considering both diagnosis accuracy and computational complexity.

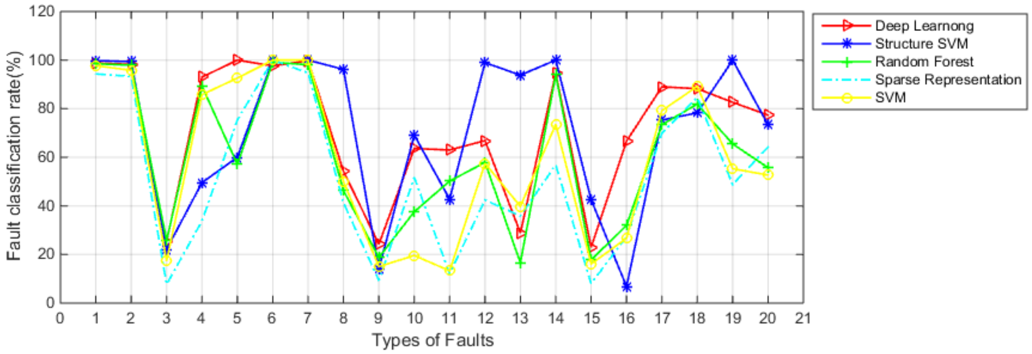

Figure 6. The average diagnosis rates of different types classification methods.

IV. CONCLUSION

In this paper, deep learning is employed to learn the deep architectures of fault data. Our method can improve the accuracy of fault detection and diagnosis due to its strong learning ability, especially for the incipient faults. A stacked auto encoder enjoys all the benefits of any deep networks' greater expressive power; it can capture the features what the statistic techniques cannot find. Another important advantage of this deep learning based fault diagnosis method is that it only needs the sample data of the current time when detecting and diagnosing online, not needs the history data. Experiments on TEP database showed that the proposed technique outperforms other machine learning approaches regarding its effectiveness and efficiency thanks to its better representation of nonlinear, complex feature data.

At present, a softmax classier is simply used with auto encoder network, for more complex fault types, better performance is expected by adding the network's structure of different depths and combining with other optimal classifier. Most importantly, we just considered instantaneous faults in our experiments now, and multiple simultaneous faults will be taken into account in our future work. Moreover, incipient fault diagnosis will be our in-depth research direction using deep learning method.

REFERENCES

[1] Wise B.M., Ricker N.L., Veltkamp D.F., Kowalski B.R.. "A theoretical basis for the use of principle component models for monitoring multivariate processes". Process Control and Quality, 1990, 1(1), pp. 41-51.

[2] Kaistha N.. "Incipient fault detection and isolation in a PWR plant using principal component analysis". In: Proceedings of the American Control Conference, 2001, 3, pp. 12-24.

[3] Jinane Harmouche, ClaudeDelpha, DembaDiallo. "Incipient fault detection and diagnosis based on Kullback–Leibler divergence using Principal Component Analysis: Part II". Signal Processing, 2015,109, pp. 334-344.

[4] Zhang X, Polycarpou M M, Parisini T. "Fault diagnosis of a class of nonlinear uncertain systems withLipschitz nonlinearities using adaptive estimation". Automatica, 2014, 46(2), pp. 290-299.

[5] Kano M., Tanaka S., Hasebe S., Hashimoto I., Ohno H.. "Monitoring independent components for fault detection". AIChE Journal, 2003, 49(4), pp. 969-976.

[6] Ge Z, Xie L, Kruger U, Song, Z.. "Local ICA for multivariate statistical fault diagnosis in systems with unknown signal and error distributions". AIChE Journal, 2012, 58(8), pp. 2357-2372.

[7] Hsu C C, Chen M C, Chen L S. "A novel process monitoring approach with dynamic independent component analysis", in Control Engineering Practice, 2010, 18(3), pp. 242-253.

[8] Zhu Z B, Song Z H. "A novel fault diagnosis system using pattern classification on kernel FDA subspace", in Expert Systems with Applications, 2011, 38(6), pp. 6895-6905.

[9] Peng Y, Chen X, Ye Q, et al. "Fault detection and classification in chemical processes using NMFSC and structural SVMs", in The Canadian Journal of Chemical Engineering, 2014, 92(6), pp. 1016-1023.

[10] Yin S, Gao X, Karimi H R, et al. "Study on support vector machine-based fault detection in tennessee eastman process", in Abstract and Applied Analysis. Hindawi Publishing Corporation, 2014.

[11] Chiang L H, Kotanchek M E, Kordon A K. "Fault diagnosis based on Fisher discriminant analysis and support vector machines", in Computers and Chemical Engineering, 2004, 28(8), pp. 1389-1401.

[12] Dietterich T. "An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting and randomization", in Machine Learning, 2000, 40(2).

[13] Wu L J, Chen X J, Peng Yi, et al. "Fault detection and diagnosis based on sparse representation classification", in Proceedings of IEEE International Conference on Robotics and Biomimetics, 2012.

[14] Hinton G E, Salakhutdinov R R. "Reducing the dimensionality of data with neural networks", in Science, 2006, 313(5786), pp. 504-507.

[15] Bengio Y. "Learning deep architectures for AI", in Foundations and trends® in Machine Learning, 2009, 2(1), pp. 1-127.

[16] Raina R, Battle A, Lee H, et al. "Self-taught learning: transfer learning from unlabeled data", in Proceedings of the 24th international conference on Machine learning, 2007, pp. 759-766.

[17] Ngiam J, Khosla A, Kim M, et al. "Multimodal deep learning". Proceedings of the 28th international conference on machine learning (ICML-11). 2011: 689-696.

[18] Chiang L H, Russell E L, Braatz R D. "Fault detection and diagnosis in industrial systems", Springer, New York, USA, 2001.

[19] Zhang Y W. "Enhanced statistical analysis of nonlinear process using KPCA, KICA and SVM", in Chemical Engineering Science, 2009, 64, pp. 800-801.