Preprints, 10th IFAC International Symposium on
Advanced Control of Chemical Processes
Shenyang, Liaoning, China, July 25-27, 2018

Fault detection and classification using artificial neural networks

Seongmin Heo and Jay H. Lee

Department of Chemical and Biomolecular Engineering
Korea Advanced Institute of Science and Technology
291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea
(email: smheo@kaist.ac.kr, jayhlee@kaist.ac.kr)

Abstract: Process monitoring is considered to be one of the most important problems in process systems engineering, which can be benefited significantly from deep learning techniques. In this paper, deep neural networks are applied to the problem of fault detection and classification to illustrate their capability. First, the fault detection and classification problems are formulated as neural network based classification problems. Then, neural networks are trained to perform fault detection, and the effects of two hyperparameters (number of hidden layers and number of neurons in the last hidden layer) and data augmentation on the performance of neural networks are examined. Fault classification problem is also tackled using neural networks with data augmentation. Finally, the results obtained from deep neural networks are compared with other data-driven methods to illustrate the advantages of deep neural networks.

Keywords: artificial neural network, deep learning, fault detection, fault classification

1. INTRODUCTION

Fault detection and diagnosis has been an active area of research for the last few decades, which is an essential part of modern industries to ensure safety and product quality. Numerous fault diagnosis methods have been proposed, and they can be classified into three categories (Venkatasubramanian et al., 2003b): quantitative model based methods, qualitative model based methods and data driven methods. As the complexity of modern processes increases, it becomes more challenging to build a mathematical model which effectively captures the system's dynamic behavior. As a result, data driven methods, which rely only on the data acquired from processes, are getting more and more attention (Venkatasubramanian et al., 2003a; Ge et al., 2013).

The key step in data driven methods is the feature extraction step, where the process data is transformed into more informative, lower dimension data. Multivariate statistical techniques, such as principal component analysis (PCA) (e.g. Wold et al., 1987) and partial least squares (PLS) (e.g. Wold et al., 1984), have been traditional methods to perform such transformation. In order to overcome inherent limitations of traditional approaches (e.g. assumption of Gaussian distribution), the variants of PCA and PLS, such as dynamic PCA (e.g. Ku et al., 1995) and modified PLS (e.g. Yin et al., 2011), as well as other methods, such as independent component analysis (ICA) (e.g. Kano et al., 2003) and fisher discriminant analysis (FDA) (e.g. Chiang et al., 2004), have been developed.

Artificial neural network based approach is another possibility, which receives significant amount of interest in recent years. Artificial neural network is a network of neurons, which learns very complex functions through a series of nonlinear transformation, and with the advent of deep learning techniques (see e.g. Schmidhuber, 2015, for an overview), it has been successfully applied to complex classification tasks such as image recognition (e.g. Simonyan and Zisserman, 2014) and speech recognition (e.g. Hinton et al., 2012). Artificial neural networks have been also adopted to address fault diagnosis problem (e.g. Eslamloueyan, 2011; Zhang and Zhao, 2017). However, most of the works utilized shallow neural networks or neural networks with hierarchical structures. Thus, the full potential of deep neural networks for addressing fault diagnosis is yet to be explored.

To this end, in this work, we apply deep artificial neural networks to address the problems of fault detection and classification. First, we formulate fault detection and classification problems as neural network based classification problems. Then, we first address fault detection problem using deep neural networks with different hyperparameter values. Specifically, we investigate the effects of two hyperparameters, number of hidden layers and number of neurons in the last hidden layer, on the performance of neural networks. We also evaluate how data augmentation affects the network performance. Finally, we examine the capability of deep neural networks for fault classification. A benchmark chemical process, the Tennessee Eastman process, is considered as an illustrative example, and the results are compared with other data-driven methods and neural network models.

2. FAULT DIAGNOSIS USING ARTIFICIAL NEURAL NETWORKS

In this section, we briefly review the concept of artificial neural networks, and how they can be used to perform classification problems. Then, we formulate fault detection and classification problems as neural network based classification problems.

2.1 Artificial neural network based classification

A typical neural network for classification problems consists of 4 different types of layers as shown in Figure 1: input layer, hidden layer, softmax layer and output layer. As it is typically the case for data-driven fault diagnosis methods, input data needs to be normalized before it is fed into the input layer, and one possible way is to apply the feature scaling of the following form so that all the values are in the range [0,1]:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{1}$$

In the hidden layers, the information contained in the input data is successively transformed into higher representations (i.e. features) through the following nonlinear transformations:

$$\begin{aligned} h_1 &= \sigma(W_1 x + b_1) \\ h_l &= \sigma(W_l h_{l-1} + b_l),\ l = \{2, \ldots, d\} \end{aligned} \tag{2}$$

where $x \in \mathbb{R}^{n_x}$, $h_l \in \mathbb{R}^{n_{h_l}}$ are the vectors of input and hidden representations, respectively, $W_l \in \mathbb{R}^{n_{h_l} \times n_{h_{l-1}}}$ and $b_l \in \mathbb{R}^{n_{h_l}}$ are the weight matrices and bias vectors, respectively, and d is the number of hidden layers. Note that n_{h_l} (i.e. number of neurons in each hidden layer) and d are hyperparameters whose values need to be determined prior to the training of neural networks. σ is a nonlinear activation function which makes the above transformation nonlinear, and in this work, we employ the rectified linear unit (ReLU) which is defined as:

$$\sigma(x) = \max(0, x) \tag{3}$$

The transformation shown in Eq.(2) without the activation function is applied to the output of the last hidden layer:

$$h_s = W_s h_d + b_s$$

and the softmax layer calculates the values of each output neuron using the softmax function of the following form:

$$y_j = \frac{\exp(h_{s,j})}{\sum_{j=1}^{n_{h_s}} \exp(h_{s,j})} \tag{4}$$

Then, the network assigns a predicted label to the input data by selecting the label with the largest output value.

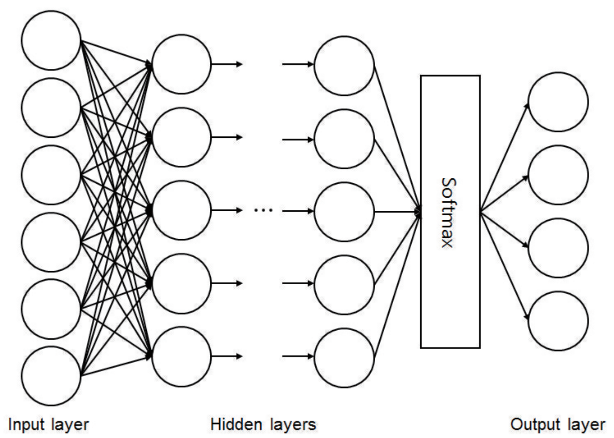


Fig. 1. Typical neural network structure for classification

The objective of the network training is to maximize the accuracy of the network, which is defined as follows:

$$\text{Accuracy} = \frac{\#\text{ of samples with correct label}}{\#\text{ of samples}} \tag{5}$$

Note that the training of neural network classifier is a supervised one since the calculation of the accuracy requires the true label of each data sample.

2.2 Fault diagnosis as a classification problem

Now, let us formulate fault detection and classification problems as neural network based classification problems. First, a fault detection problem can be formulated as a binary classification problem where the two labels are *normal* and *fault*. A neural network, in this case, can be trained using two different data sets: one with the normal operation data, and the other with the operation data with a specific type of fault. Along with the accuracy, two indices generally used for fault diagnosis, fault detection rate (FDR) and false alarm rate (FAR), can be defined as:

$$\text{FDR} = \frac{\#\text{ of faulty samples with fault label}}{\#\text{ of faulty samples}} \tag{6}$$

$$\text{FAR} = \frac{\#\text{ of normal samples with fault label}}{\#\text{ of normal samples}} \tag{7}$$

Then, a fault classification problem can be formulated similarly as a multiclass classification problem. Now, we can train a neural network using multiple data sets each of which contains normal and various faulty operation data. In this case, the performance of neural networks can be evaluated using the accuracy and confusion matrix, which is a square matrix whose i, j-th element is defined as the number of samples whose true and predicted labels are i and j, respectively. Note that the accuracy can be directly calculated from the diagonal elements of the confusion matrix.

In what follows, via a case study of a benchmark chemical process, we analyze the effects of network structures on the fault detection, and the effects of data structure on the fault detection and classification.

3. CASE STUDY - TENNESSEE EASTMAN PROCESS

In this section, we apply neural network classifiers to the Tennessee Eastman (TE) process, which is a benchmark process for various studies including fault diagnosis. First, we provide brief descriptions of the TE process, and how neural networks are trained. Then, the fault detection and classification results are analyzed.

3.1 Process description

The TE process was introduced in Downs and Vogel (1993) as a test problem for process control and monitoring techniques. It consists of five major process units, the reactor, condenser, compressor, separator and stripper, as shown in Figure 2, and produces two products, G and H, and a byproduct, F, from four reactants, A, C, D and E. There also exists an inert compound, B. There are 52 measurements available (41 measurements for process variables and 11 measurements for manipulated variables),

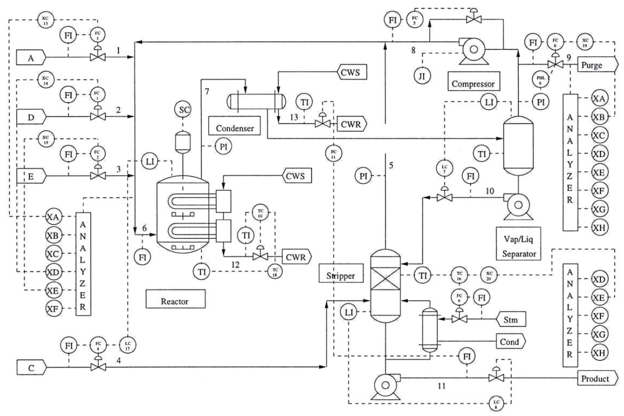


Fig. 2. Process flow diagram of the TE process

Fault ID	Process variable	Type
Fault 1	A/C feed ratio, B composition constant (stream 4)	Step
Fault 2	B composition, A/C ratio constant (stream 4)	Step
Fault 3	D feed temperature (stream 2)	Step
Fault 4	Reactor cooling water inlet temperature	Step
Fault 5	Condenser cooling water inlet temperature	Step
Fault 6	A feed loss (stream 1)	Step
Fault 7	C header pressure loss - reduced availability (stream 4)	Step
Fault 8	A, B, C feed composition (stream 4)	Random variation
Fault 9	D feed temperature (stream 2)	Random variation
Fault 10	C feed temperature (stream 4)	Random variation
Fault 11	Reactor cooling water inlet temperature	Random variation
Fault 12	Condenser cooling water inlet temperature	Random variation
Fault 13	Reaction kinetics	Slow drift
Fault 14	Reactor cooling water valve	Sticking
Fault 15	Condenser cooling water valve	Sticking
Fault 16	Unknown	Unknown
Fault 17	Unknown	Unknown
Fault 18	Unknown	Unknown
Fault 19	Unknown	Unknown
Fault 20	Unknown	Unknown

Table 1. Fault description of the TE process

Number of hidden layers	Network structure
0	52-2
1	52-25-2
2	52-25-12-2
3	52-52-25-2-2
4	52-52-25-12-2-2

Table 2. Network structure of neural networks with different number of layers

and 20 different fault types are defined in Downs and Vogel (1993) as summarized in Table 1.

In Chiang et al. (2000) and Zhang (2009), it is pointed out that it is especially difficult to detect Faults 3, 9 and 15 due to the absence of observable change in the mean, variance and the higher order variances. Thus, in this study, these faults are not considered in the following analysis.

3.2 Network training

The data sets, which are recently published online (Rieth et al., 2017), are used for the training and testing of neural networks. The data sets in Rieth et al. (2017) provide data of 500 simulation runs for each normal/fault state (10500 runs in total), and have basically the same structure as the data sets provided in Chiang et al. (2000). Each data set contains the results of a simulation run of 25 hours with a sampling time of 3 minutes, resulting in 500 data samples. In the case of faulty operation, a specific type of fault is introduced after 1 hour. For each state, 300 simulation runs are used for the training, and the remaining 200 simulation runs are used for the testing. The networks are initially designed to have 52 input nodes so that each data sample is directly used as the input to the network.

Neural networks are initialized using the Xavier initialization (Glorot and Bengio, 2010) to make sure that, initially, the signals do not fade away or explode, and the ADAM optimizer (Kingma and Ba, 2014) is adopted for the training. The training data sets are divided into 50 batches for batch training, and the networks are trained for 400 training epochs.

3.3 Fault detection results

Number of hidden layers To examine the potential of deep learning, we first solve the fault detection problem using neural networks with different number of hidden layers. Neural networks with 0, 1, 2, 3, and 4 hidden layers are trained, and the network structure for each case is summarized in Table 2.

Figure 3 shows the overall accuracy (over all the fault IDs) of each network. From this figure, we can see that having a single hidden layer (i.e. a shallow neural network) is not very helpful, providing only a slight improvement over the case where we directly apply the softmax layer to the input layer (i.e. network with no hidden layer). The largest improvement is obtained when we add a second hidden layer to the network, and adding more hidden layers do not have significant impact on the accuracy of the network.

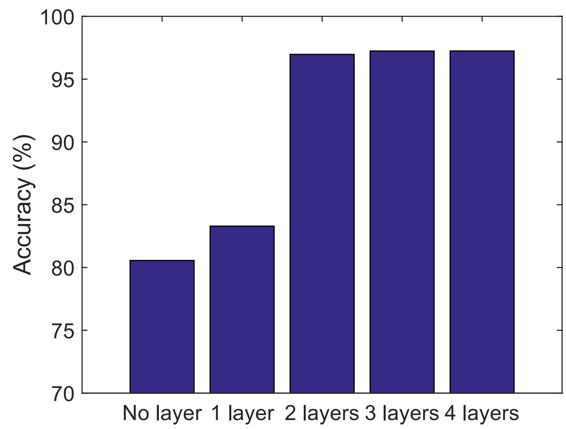


Fig. 3. Overall fault detection accuracy of neural networks with different number of hidden layers

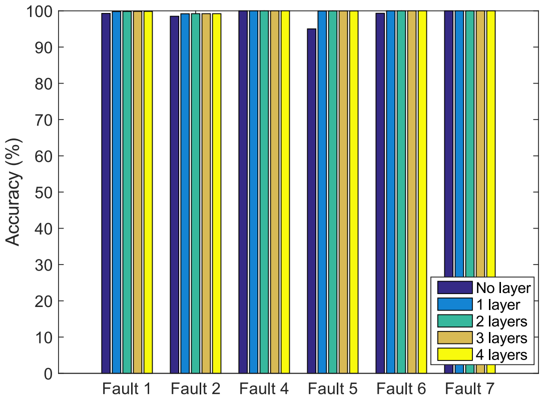


Fig. 4. Fault detection accuracy for Fault type 1 using different number of hidden layers

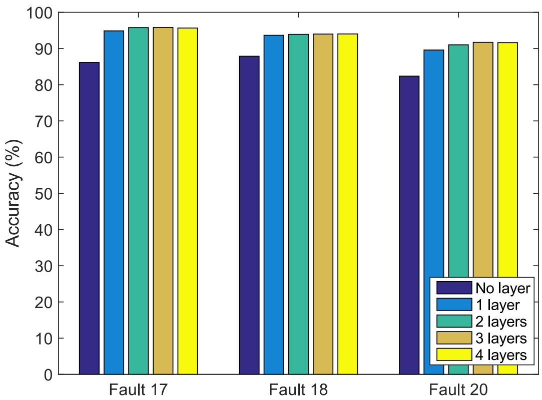


Fig. 5. Fault detection accuracy for Fault type 2 using different number of hidden layers

Also, different fault IDs can be classified into three categories based on the minimum number of hidden layers required to achieve an acceptable level of fault detection accuracy (here, defined as 90% of accuracy):

- Fault type 1: $\{1, 2, 4, 5, 6, 7\}$
- Fault type 2: $\{17, 18, 20\}$
- Fault type 3: $\{8, 10, 11, 12, 13, 14, 16, 19\}$

Fault type 1 is successfully detected even using the network with no hidden layer as shown in Figure 4. Note that all the faults of the *step* type are included in this fault type. Figure 5 shows the accuracy of fault detection for Fault type 2, and we can see that a shallow network is enough to detect this fault type. Note that some of the *unknown* type faults are classified as Fault type 2. Fault type 3 required at least 2 hidden layers to be effectively detected as it can be seen from Figure 6, and note that this fault type contains the faults which are not *step* type as well as some *unknown* faults. Figure 7 shows an example of how features are evolving in time and how they are distributed in two dimensional feature space.

Number of neurons in the last hidden layer Now, let us analyze the effects of number of neurons in the last hidden layer (i.e. number of features from which classification is performed) on the performance of fault detection. In this

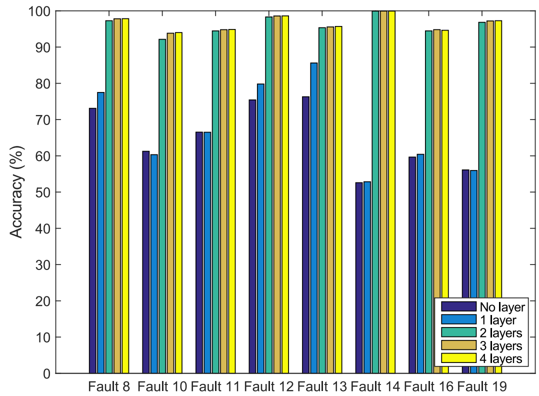
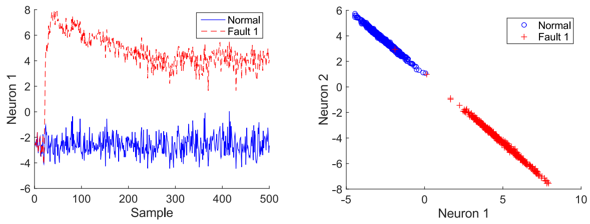


Fig. 6. Fault detection accuracy for Fault type 3 using different number of hidden layers



(a) (b)

Fig. 7. Fault detection result of Fault 1 (a) temporal evolution of features, (b) distribution of features

Number of neurons	1	2	3	12
Accuracy (%)	97.20	97.24	97.26	97.26

Table 3. Overall fault detection accuracy of neural networks with different number of neurons in the last hidden layer

Input data length	1	2	3
Accuracy (%)	97.24	97.65	97.73

Table 4. Overall fault detection accuracy with the augmented input

analysis, we use the neural network with three hidden layers whose structure up to the last hidden layer is 52-52-25. Neural networks with 1, 2, 3 and 12 neurons in the last hidden layer are trained, and the overall accuracy of fault detection is summarized in Table 3.

We can see that, although having more neurons in the last hidden layer improves the overall accuracy of fault detection, the rate of improvement is very small and it eventually diminishes. It may imply that the neural network has reached its maximum potential to discriminate faulty samples from the normal samples with the current input data. Thus, in what follows, the input data is augmented to test if the overall accuracy can be further improved.

Data augmentation In this analysis, we augment the input data by combining a few consecutive samples, mimicking dynamic principal component analysis. The neural network with three hidden layers, whose structure is 52-52-25-2-2, is used to obtain the results, and the augmented inputs are prepared by combining 2 and 3 consecutive samples. The overall fault detection accuracy with the data augmentation is tabulated in Table 4. Note that the overall accuracy is improved, and the data augmentation has stronger impact on the overall accuracy than the number of neurons in the last hidden layer.

Through the data augmentation, the fault detection accuracy for Faults 1, 4 and 5 has reached 100%. Also, the fault detection accuracy for Faults 11 (from 94.8% to 97.2%), 19 (from 97.22% to 99.18%) and 20 (from 91.71% to 93.62%) has been improved significantly.

Comparison with other data-driven methods Now, let us compare our results with other data-driven methods. The results obtained using dynamic principal component analysis (DPCA), modified partial least squares (MPLS), and independent component analysis (ICA), which are reported in Yin et al. (2012), and the results from deep belief network (DBN) with sigmoid and Gaussian activation functions (abbreviated as s and G later on, respectively), which are reported in Zhang and Zhao (2017), are used for the comparison.

Table 5 summarizes the results from different methods, and our deep neural network model shows the best overall fault detection rate. However, in the case of Faults 8, 12, 14 and 17, traditional methods resulted in better fault detection rates, and the DBN proposed in Zhang and Zhao (2017) performed better than our model in the case of Faults 10, 17 and 18. Note that our neural network model is not fully optimized, especially in terms of data augmentation, implying that there still exists a potential for our model to produce better results than other methods on the fault IDs mentioned above.

False alarm rate is also compared with the results reported in the same references, and the values are shown in Table 6. Note that our deep neural network model outperforms the other methods, showing very low false alarm rate.

3.4 Fault classification results

Now, let us consider the fault classification problem of the TE process. Fault classification problem is solved using the original input data and the augmented input data prepared by combining 2 consecutive samples. The neural networks with the structure of 52-52-52-40-18 and 102-102-50-40-18 are trained using the original input and the augmented input, respectively.

Classification accuracy for each normal/fault state is shown in Figure 8, and the confusion matrices are not provided here for brevity. Note that, for the most of the normal/fault states, the classification accuracy is increased by augmenting the input data. Note also that, while the data augmentation improves the fault detection accuracy of Fault 19, it results in significantly higher misclassification rate of normal state as Fault 19 (from 3.27% to 10.56%), and of Fault 19 as normal state (from 6.12% to 11.61%), and in turn, lower classification accuracy of normal state and Fault 19.

DPCA	MPLS	ICA	DBN (s)	DBN (G)	Ours
15.13	10.75	2.63	7.03	9.14	0.25

Table 6. False alarm rates (%) of different data-driven methods

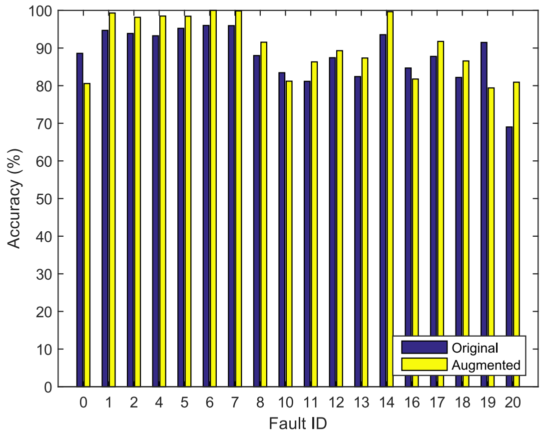


Fig. 8. Fault classification accuracy with the original and augmented data

Fault ID	DPCA	MPLS	ICA	DBN (s)	DBN (G)	Ours
1	99.88	100	99.88	100	98	100
2	99.38	98.88	98.75	97	95	99.51
4	100	100	100	100	100	100
5	43.25	100	100	87	79	100
6	100	100	100	100	100	100
7	100	100	100	100	100	100
8	98	98.63	97.88	77	89	98.06
10	72	92.75	89	0	98	93.96
11	91.5	83.25	79.75	12	91	97.20
12	99.25	99.88	99.88	1	72	98.69
13	95.38	95.5	95.38	60	91	95.78
14	100	100	100	5	91	99.97
16	67.38	94.38	80.13	0	0	95.41
17	97.25	97.13	96.88	100	100	95.93
18	90.88	91.25	90.5	100	78	94.15
19	87.25	94.25	93.13	13	98	99.18
20	73.75	91.5	90.88	93	93	93.62
Overall	89.13	96.32	94.83	61.47	86.65	97.73

Table 5. Fault detection rates (%) of different data-driven methods

Fault ID	SNN	HNN	SAE	Ours
1	81.19	97.51	98.75	99.31
2	81.97	98.26	98.33	98.16
4	80.02	95.89	93.10	98.50
5	73.33	97.13	99.79	98.46
6	83.31	99.38	97.70	100
7	81.49	100	99.37	99.86
8	46.65	60.15	53.98	91.55
10	10.96	46.33	63.39	81.19
11	17.30	47.32	62.97	86.32
12	24.12	45.21	66.74	89.30
13	16.93	31.51	29.29	87.35
14	29.11	66.75	94.56	99.66
16	15.83	36.61	66.53	81.76
17	51.15	70.74	88.70	91.76
18	75.76	94.89	88.29	86.57
19	14.50	51.18	82.64	79.40
20	46.16	66.25	77.20	80.92
Overall	48.83	70.89	80.08	91.18

Table 7. Fault classification rates (%) of different neural network models

Our classification results are compared with the results provided in two references. In Eslamloueyan (2011), a hierarchical neural network (HNN), where multiple neural networks are trained to classify subgroups of faults, is designed, and compared with shallow neural network (SNN). Stacked sparse auto encoder (SAE) is trained in Lv et al. (2016) through deep learning. The results from different methods are tabulated in Table 7. Note that, our network, which in principle is designed to perform fault detection and classification simultaneously (the normal state is also included in the classification problem), outperforms other networks which are designed only for the classification problem.

4. CONCLUSION

In this paper, we applied deep neural networks to the problem of fault detection and classification. In the case of fault detection, we investigated the effects of two hyperparameters (number of hidden layer, number of neurons in the last hidden layer) on the performance of networks, and concluded that increasing the network size does not improve the fault detection accuracy above certain level (approximately 97.26%). Then, we showed that the data augmentation can be a key to increase the fault detection accuracy further, and it also turned out to be beneficial for the fault classification case.

Although the results presented in this paper look promising, several points need to be addressed in the future work. First, the characteristics of the features and the faults need to be analyzed in detail to understand how neural network classifier works and to improve its performance. Second, the effects of data augmentation need to be investigated further. Lastly, different types of neural network (e.g. convolutional neural network) need to be tested to see if they fit better for the fault detection and classification problems.

ACKNOWLEDGEMENT

This work was supported by the Advanced Biomass R&D Center(ABC) of Global Frontier Project funded by the Ministry of Science and ICT (ABC-2011-0031354).

REFERENCES

Chiang, L., Kotanchek, M., and Kordon, A. (2004). Fault diagnosis based on fisher discriminant analysis and support vector machines. *Comput. Chem. Eng.*, 28(8), 1389–1401.

Chiang, L., Russell, E., and Braatz, R. (2000). *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media.

Downs, J. and Vogel, E. (1993). A plant-wide industrial process control problem. *Comput. Chem. Eng.*, 17(3), 245–255.

Eslamloueyan, R. (2011). Designing a hierarchical neural network based on fuzzy clustering for fault diagnosis of the Tennessee–Eastman process. *Appl. Soft Comput.*, 11(1), 1407–1415.

Ge, Z., Song, Z., and Gao, F. (2013). Review of recent research on data-based process monitoring. *Ind. Eng. Chem. Res.*, 52(10), 3543–3562.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 249–256.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Proc. Mag.*, 29(6), 82–97.

Kano, M., Tanaka, S., Hasebe, S., Hashimoto, I., and Ohno, H. (2003). Monitoring independent components for fault detection. *AIChE J.*, 49(4), 969–976.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ku, W., Storer, R., and Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometr. Intell. Lab.*, 30(1), 179–196.

Lv, F., Wen, C., Bao, Z., and Liu, M. (2016). Fault diagnosis based on deep learning. In *American Control Conference (ACC), 2016*, 6851–6856. IEEE.

Rieth, C., Amsel, B., Tran, R., and Cook, M. (2017). Additional Tennessee Eastman process simulation data for anomaly detection evaluation. doi: 10.7910/DVN/6C3JR1.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S., and Yin, K. (2003a). A review of process fault detection and diagnosis: Part III: Process history based methods. *Comput. Chem. Eng.*, 27(3), 327–346.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., and Kavuri, S. (2003b). A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Comput. Chem. Eng.*, 27(3), 293–311.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometr. Intell. Lab.*, 2(1-3), 37–52.

Wold, S., Ruhe, A., Wold, H., and Dunn, III, W. (1984). The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses. *SIAM J. Sci. Stat. Comp.*, 5(3), 735–743.

Yin, S., Ding, S., Haghani, A., Hao, H., and Zhang, P. (2012). A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. *J. Process Contr.*, 22(9), 1567–1581.

Yin, S., Ding, S., Zhang, P., Hagahni, A., and Naik, A. (2011). Study on modifications of PLS approach for process monitoring. *IFAC Proceedings Volumes*, 44(1), 12389–12394.

Zhang, Y. (2009). Enhanced statistical analysis of nonlinear processes using KPCA, KICA and SVM. *Chem. Eng. Sci.*, 64(5), 801–811.

Zhang, Z. and Zhao, J. (2017). A deep belief network based fault diagnosis model for complex chemical processes. *Comput. Chem. Eng.*, 107, 395–407.