Journal of Dynamics, Monitoring and Diagnostics, 2022, 1, 127-138
https://doi.org/10.37965/jdmd.2022.67

RESEARCH ARTICLE

An Intelligent Process Fault Diagnosis System based on Andrews Plot and Convolutional Neural Network

Shengkai Wang and Jie Zhang
School of Engineering, Merz Court, Newcastle University, Newcastle upon Tyne, NE1 7RU, UK

(Received 05 December 2021; Revised 24 April 2022; Accepted 13 May 2022; Published online 19 May 2022)

Abstract: This paper proposes an intelligent process fault diagnosis system through integrating the techniques of Andrews plot and convolutional neural network. The proposed fault diagnosis method extracts features from the on-line process measurements using Andrews function. To address the uncertainty of setting the proper dimension of extracted features in Andrews function, a convolutional neural network is used to further extract diagnostic information from the Andrews function outputs. The outputs of the convolutional neural network are then fed to a single hidden layer neural network to obtain the final fault diagnosis result. The proposed fault diagnosis system is compared with a conventional neural network based fault diagnosis system and integrating Andrews function with neural network and manual selection of features in Andrews function outputs. Applications to a simulated CSTR process show that the proposed fault diagnosis system gives much better performance than the conventional neural network based fault diagnosis system and manual selection of features in Andrews function outputs. It reveals that the use of Andrews function and convolutional neural network can improve the diagnosis performance.

Keywords: Andrews plot; convolutional neural network; fault diagnosis; neural network

I. INTRODUCTION

Industrial chemical processes have become more productive and automated. Consequently, they become more complicated, making the failure-related risks easily to hide in the production processes. Undetected process faults can lead to serious consequences, such as inferior products, environmental contamination, or even causing casualties in a plant accident. These quintessentially consequences can reduce profitability and affect the reputation of enterprise. Since process faults are hard to completely eliminate in industry processes, modern industries demand some advantageous initiatives to acquire better process monitoring. Numerous researches in process monitoring focus on development of fault diagnosis systems to against this conundrum.

The various fault diagnosis approaches for industrial processes proposed in the past can be roughly divided into knowledge based approaches, model based approaches, and data based approaches [1–4]. Industrial process operations are composed by known principles and unknown uncertainties. The unpredictable disturbances in processes are omnipresence and time-varying. As more and more process operation data are monitored and archived, the widely accepted methods in process fault diagnosis are based on multivariate statistical analysis [5–9] and neural networks [10–15]. With the advent of the era of big data, neural networks have been evolved from shallow networks to deep networks. A myriad of studies in the research area of fault diagnosis evidenced that a fault diagnosis system based on deep neural network can achieve encouraging performance improvement [16–19]. A normalised sparse autoencoder is used to extract features from mechanical signals for fault diagnosis in mechanical systems [16]. Convolutional neural network (CNN) and domain adaption are used in [17] to address the inconsistence between the training data and testing data in mechanical system fault diagnosis. A fault diagnosis method based on federated learning and CNN is proposed in [18] for rolling bearing fault diagnosis. A CNN-long short-term memory (LSTM) approach for forecasting the system parameters in future sampling windows is proposed in [19] for fault prognosis. To address the difficulty of recurrent neural networks in directly extracting degradation features from original monitoring data, a residual convolution layer is stacked to a LSTM network for the prediction of remaining useful life [20].

In chemical process monitoring, the on-line measured process data generally forms the on-line monitoring information sources. Key features reflecting the process operation states are hidden in the multivariate process measurements. Discovering these features is important for successful process fault diagnosis. This research work proposes a method for extracting these features using Andrews plot and CNN. The matrix dimensions of the Andrews function outputs can be flexibly changed to an appropriate size according to the specific processes. This gives the potential that the fault diagnosis system development can easily be integrated with CNN, since the matrix size can be adjusted flexibly and CNN can handle large input matrix sizes.

The techniques utilized in the proposed fault diagnosis method include principal component analysis (PCA), Andrews plot, and convolutional neural network. PCA, as a multivariate statistical data analysis method, is widely adopted in various fields for data dimension reduction. In the monitoring of an industrial chemical process, a PCA model is developed from the historical process operation data when the process is under normal operation. Due to the correlation between the monitored process, the number of retained principal components is usually much less that the number of monitored variables. In the method developed in this paper, the purpose of PCA is to eliminate the influence of variable ordering on the results of Andrews plot and,

Corresponding author: Jie Zhang (email: jie.zhang@newcastle.ac.uk) hence, all the principal components are retained. Andrews plot is a technique for the visualization of high-dimensional data with applications in many fields including analysing the UK general election data in 2001 [21,22]. The algorithm is advantageous in setting the dimension of the extracted features. Since the number of monitoring variables in an industrial process is fixed, traditional process data analysis methods are generally utilized for dimension reduction. Andrews plot method can adjust the dimensions of monitoring dataset while maintaining the main feature of monitoring information. However, the different selection of parameter is generally accompanied by hard-predictable and ineliminable uncertainties to affect the results of data analysis. Despite this limitation, Andrews plot gives a potential to expand possibilities of data processing in fault diagnosis [12]. Convolutional neural network is a deep neural network with convolutional structure, which is one of the most popularly deep neural network structures in the past decade. The study of process fault diagnosis has also progressively accepted CNN in recently years. Numerous prior studies evidenced that CNN has strong adaptability, exceptional local feature data mining capability, global training feature extraction and classification [23–26]. Our previous work [12] shows that integrating Andrews plot with neural networks can enhance fault diagnosis performance and proposes a method for determining the features in Andrews function outputs. However, implementing this approach for determining the features in Andrews function outputs still needs some subjective decision. To overcome the uncertainty in determining the dimension in the Andrews function outputs, the paper proposes using a CNN to further process the Andrews function outputs as CNN can efficiently handle high dimensional inputs. The final extracted features by Andrews plot and CNN are then classified by a feedforward neural network to give the final diagnosis results.

In order to demonstrate the feasibility and superiority of the proposed method, it is compared with a traditional neural network based diagnosis scheme and our previous method with manual selection of features in Andrews function outputs with applications to a simulated continuous stirred tank reactor (CSTR) system. The contributions of this paper are summarised as follows: 1). Andrews plot and CNN are used to extract features in the monitored process data and this would ease the task of fault classification in the neural network. The use of CNN after Andrews plot is to ease the task of determining the appropriate numbers of features in Andrews function. 2). The proposed method is comprehensively evaluated through mechanistic model based simulation for both abrupt and incipient faults and compared with a conventional neural network based fault diagnosis approach and the previous approach of manual selection of features in Andrews function outputs.

The paper is structured as follows. A CSTR system used as a case study is given in Section 2. Section 3 presents the proposed diagnosis system and details of parameter selection. Fault diagnosis results are given in Section 4 where they are also compared with those from a conventional neural network based fault diagnosis system and integrating Andrews plot with neural networks and manual selection of features in Andrews function outputs. Some concluding remarks are given in Section 5.

II. A CSTR PROCESS WITH RECYCLE

The CSTR process with recycle, shown in Fig. 1, is taken from [27] and is used as a case study. The operation of this CSTR system is simulated using a dynamic mechanistic model, which is based on material and energy balances, as well as reaction kinetics. The historical process operation data under normal operation and different considered faults are generated from simulation. In this process, the on-line monitoring information sources include 10 on-line measured process variables and 3 controller outputs. In this study, 11 faults shown in Table I are considered and they represent typical faults in industrial processes. The simulated process operation data are corrupted with typical measurement noise to represent the practical situations

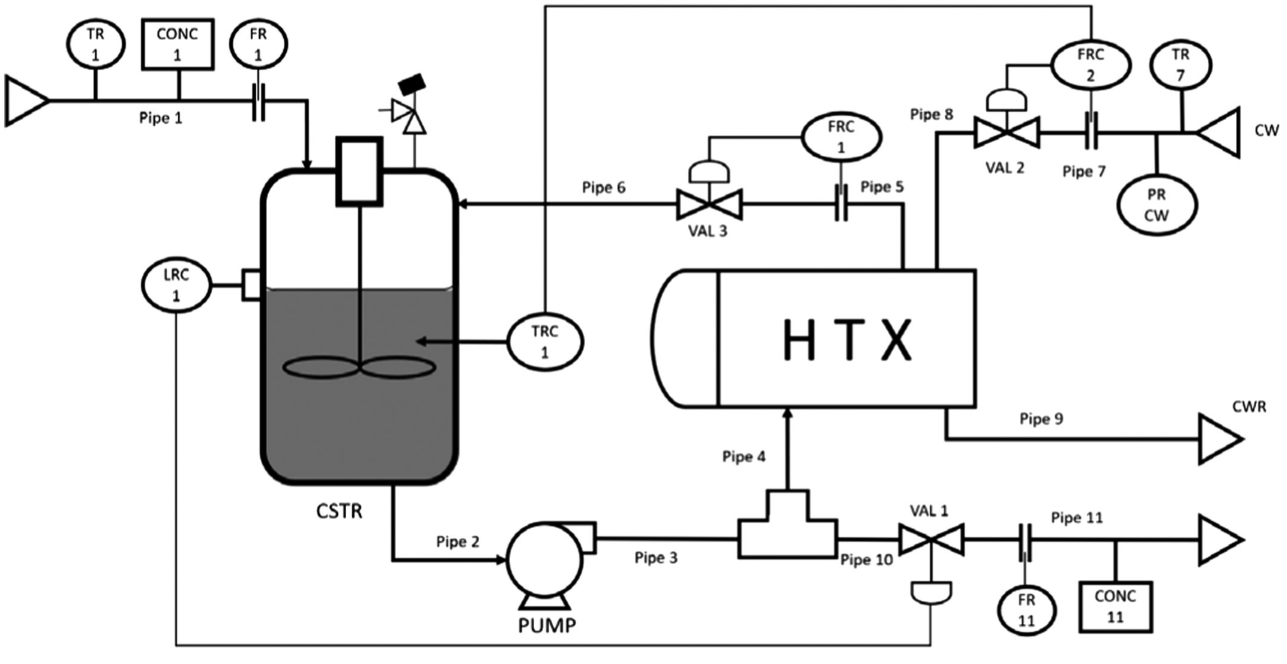

Fig. 1. A CSTR with a recycle.

Table I. List of considered faults

Fault No.	Descriptions of faults
1	Blockage of pipe 1
2	Too high flow rate in reactant feed rate
3	Blockage of pipe 2 or 3 or pump fails
4	Blockage of pipe 10 or 11 or control valve 1 fails low
5	Abnormal temperature in feed-reactant
6	Control valve 2 fails high
7	Blockage of pipe 7, 8, or 9 or control valve 2 fails low
8	Control valve 1 fails high
9	Blockage of pipe 4, 5, or 6 or control valve 3 fails low
10	Control valve 3 fails high
11	Too low concentration in external feed reactant

Table II. Ranges of measurement noise

Measurements	Noise Range
Flow	$-1 \sim 1$ cm^3/s
Temperature	$-0.25 \sim 0.25$ °C
Level	$-0.1 \sim 0.1$ cm
Concentration	-0.5 % ~ 0.5 %

where measurement noises always exist. The measurement noise ranges are given in Table II. The sampling time used in the control system is 4 s.

In industrial processes, a fault can appear in two forms: 1). an abrupt fault with a sudden variation of the process parameter corresponding to the fault; or 2). an incipient fault with the fault severity gradually growing with time. An abrupt fault is usually simulated as a step change in the process parameter corresponding to the fault. An incipient fault can be represented as the following [11]:

$$M_f(t) = M_n(1 + \gamma t) \tag{1}$$

where $M_f(t)$ is the faulty value of a process parameter at time t, M_n stands for the normal value of the same process parameter, γ represents the fault developing speed, and t is the time from the initiation of the fault. The above equation can generally describe the commonly encountered incipient faults in industrial processes.

III. FAULT DIAGNOSIS SYSTEM DEVELOPMENT

A. PRINCIPAL COMPONENT ANALYSIS

Principal component analysis is one of the most widely used data dimension reduction algorithms. Its main idea is to reduce the dimension of a correlated data set into a set of lower dimensional uncorrelated principal components with minimal loss of information. These principal components are linear combinations of the original variables [28,29]. In this study, the principal components instead of the original measured information are used to remove the impact of variable ordering in Andrews plot. Hence, the entire set of principal components are used without data dimension reduction. The principal components are not influenced by variable ordering. In practice, one of the most commonly used methods to obtain the principal components is singular value decomposition (SVD).

For a process dataset X_n collected under normal process operation, its SVD decomposition is as the following:

$$X_n = U \times \Sigma \times V^T \tag{2}$$

where X_n is a set of normal data with size $m \times n$, U is a matrix of $m \times m$, V is a matrix of $n \times n$, Σ is a matrix of $m \times n$ and the values of all elements except the main diagonal of the top sub-matrix are 0, and the elements on the main diagonal are called singular values, which are nonnegative and arranged in descending order.

Dimensionality reduction of the monitoring dataset, e.g. reducing to k principal components X'_k, can be achieved as:

$$X'_k = X_n V_k \tag{3}$$

where V_k is a matrix containing the first k columns of V.

B. ANDREWS PLOT

Andrews plot or Andrews curve is named after David F. Andrews [30], who proposed the curves to visualize high-dimensional data. Each sample of an a-dimensional data set, $X = (x_1,x_2,x_3, \cdots ,x_a)$, can be mapped into a curve using the following function:

$$f_x(t) = \frac{x_1}{\sqrt{2}} + x_2\mathrm{sint} + x_3\mathrm{cost} + x_4 \sin 2t + \cdots; \tag{4}$$

where $t \in [-\pi,\pi]$.

The dimension of the extracted features can be adjusted to an appropriate size by using different values of t in Andrews function. In applications to industrial process monitoring, each sample of the original monitored data is converted into a feature vector through the Andrews function by employing a certain number of t-values. The size of feature vector is equal to the number of t-values. Previous studies reveal that the outcome of Andrews function processing is sensitive to the data arrangement (i.e. variable ordering) and will produce uncertainties [31]. Hence, the principal components are generally used in place of the original process measurement as the inputs to Andrews function.

Figure 2 gives the details of Andrews function processing. The historical process data, $X = (x_1,x_2,x_3, \cdots ,x_a)$, are first scaled to zero mean and unit variance before its principal components are calculated. Then these principal components are used as the inputs to the Andrews function with n values of t. The final processed feature dataset has a dimension of n and contains the main information of the historical process data X.

Figure 3 shows an example of Andrews function application to the data from the CSTR process. The Andrews function is given by Eq. (6) and 50 samples from each class with 63 t values uniformly distributed in the range from $-\pi$ to π being used. Figure 3 shows the Andrews function outputs for the normal process operation data represented by the red solid curves, the process operation data under fault No. 3 represented by the blue dashed curves, and the process operation data under fault No. 5 represented by the black dotted curves. The curves shown in Fig. 3 reveal that the process operation data pre-processed by Andrews function can lead to clear separations between

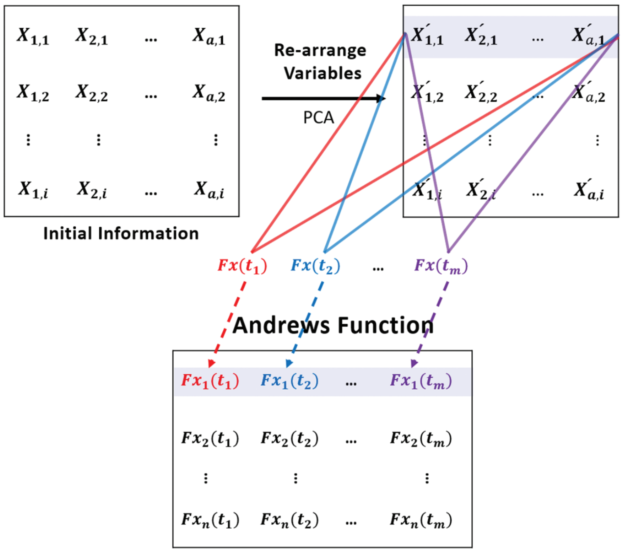
$X_{1,1}$ $X_{2,1}$ ... $X_{a,1}$
$X_{1,2}$ $X_{2,2}$ ... $X_{a,2}$
$X_{1,i}$ $X_{2,i}$ ... $X_{a,i}$
Initial Information
Re-arrange Variables
PCA
$X'_{1,1}$ $X'_{2,1}$ ... $X'_{a,1}$
$X'_{1,2}$ $X'_{2,2}$ ... $X'_{a,2}$
$X'_{1,i}$ $X'_{2,i}$ ... $X'_{a,i}$
$Fx(t_1)$ $Fx(t_2)$ ... $Fx(t_m)$
Andrews Function
$Fx_1(t_1)$ $Fx_1(t_2)$ ... $Fx_1(t_m)$
$Fx_2(t_1)$ $Fx_2(t_2)$ ... $Fx_2(t_m)$
$Fx_n(t_1)$ $Fx_n(t_2)$ ... $Fx_n(t_m)$

Fig. 2. Andrews function processing.

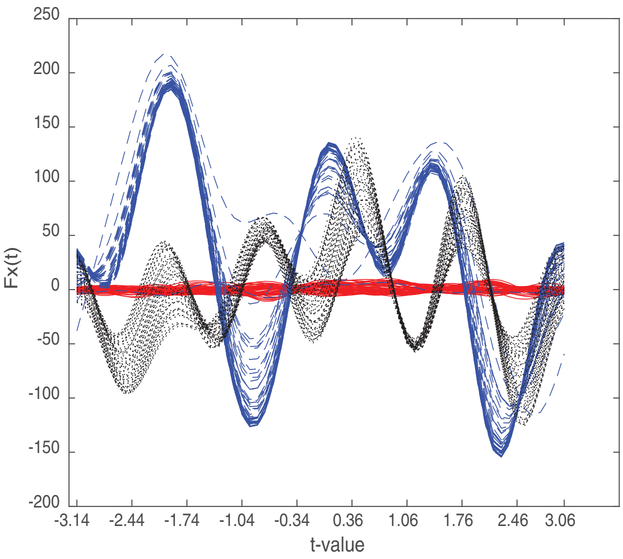

Fig. 3. Andrews plot on datasets from the CSTR system.

the normal process operation data and various faulty process operation data. Thus using Andrews function to preprocess data could enhance fault diagnosis.

C. CONVOLUTIONAL NEURAL NETWORK

A convolutional neural network is a type of deep neural network with convolutional structure. In recent years, CNN has become one of the most popular technique of artificial intelligence. Its development has gone through decades, with several great advances from LeNet to AlexNet, VGG, ResNet etc. [32–35]. CNN includes the non-linear trainable convolutional layers, sub-sampling layers (referred as pooling layer) and a fully connected layer. Figure 4 shows a basic structure of a convolutional neural network.

In a convolutional layer, the outputs represent feature maps. Each neuron in the output map is connected to a local patch in the input map via a weight kernel. Each convolutional operation shares the same weight kernel [36]. Assume that the size of original input data is $W_0 \times H_0 \times D_0$, where $W_0 \times H_0$ represents a sample of pre-processed features, and D_0 represents the number of samples. The number of weight kernel θ is N, the spatial extent of weight kernel is $F \times F$, the stride is S, the amount of zero padding is P. Then the output feature map can be calculated by the following equation [37]:

$$x_j^l = f\left(\sum_{i=1,\ldots,D_0} x_i^{l-1} * \theta_{ij}^l + b_j^l\right), \qquad j = 1,\ldots,N \qquad (5)$$

where x_i^{l-1} represents the ith input map, x_j^l represents the jth output map, θ_{ij}^l represents the jth kernel connected to the ith input map, b_j^l represents bias corresponding to the jth kernel, f represents activation function, and $*$ represents the convolutional operation.

The size of output map after convolution operation is $W \times H \times D$:

$$W = \frac{W_0 - F}{S} \qquad (6)$$

$$H = \frac{H_0 - F}{S} + 1 \qquad (7)$$

$$D = N \qquad (8)$$

The commonly used activation functions in CNN include sigmoid function $f(x)_{\text{sigm}}$, hyperbolic tangent function $f(x)_{\text{tanh}}$, and rectified linear unit (ReLU) function $f(x)_{\text{ReLU}}$.

Figure 5 gives an example of convolutional operation, where the size of input layer is $m \times m \times 1$, the size of weight kernel θ is $a \times a \times 1$, the stride is 1, the amount of zero

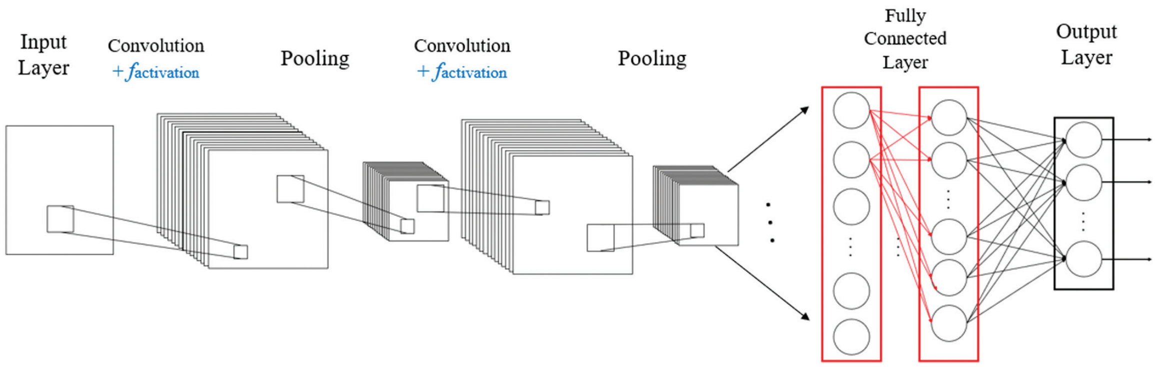

Fig. 4. The basic structure of a convolutional neural network.

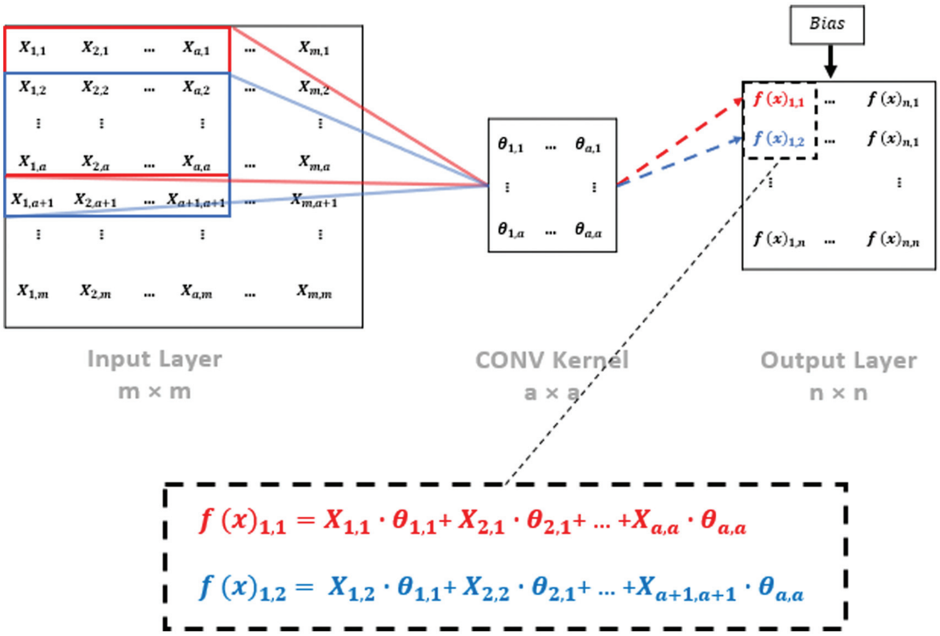
Bias
$X_{1,1}$ $X_{2,1}$ … $X_{a,1}$ … $X_{m,1}$
$X_{1,2}$ $X_{2,2}$ … $X_{a,2}$ … $X_{m,2}$
$X_{1,a}$ $X_{2,a}$ … $X_{a,a}$ … $X_{m,a}$
$X_{1,a+1}$ $X_{2,a+1}$ … $X_{a+1,a+1}$ … $X_{m,a+1}$
$X_{1,m}$ $X_{2,m}$ … $X_{a,m}$ … $X_{m,m}$
$\theta_{1,1}$ … $\theta_{a,1}$
$\theta_{1,a}$ … $\theta_{a,a}$
$f(x)_{1,1}$ … $f(x)_{n,1}$
$f(x)_{1,2}$ … $f(x)_{n,1}$
$f(x)_{1,n}$ … $f(x)_{n,n}$
Input Layer
m × m
CONV Kernel
a × a
Output Layer
n × n
$f(x)_{1,1} = X_{1,1} \cdot \theta_{1,1} + X_{2,1} \cdot \theta_{2,1} + \ldots + X_{a,a} \cdot \theta_{a,a}$
$f(x)_{1,2} = X_{1,2} \cdot \theta_{1,1} + X_{2,2} \cdot \theta_{2,1} + \ldots + X_{a+1,a+1} \cdot \theta_{a,a}$

Fig. 5. An example of convolutional operation.

padding is 0. The size of output map is $n \times n \times 1$, where $n = m - a + 1$.

The pooling layer (sub-sampling layer) is applied to merge the similar local features. The pooling operation can enhance the reliability of the CNN model and reduce the number of parameters sharply [38]. The pooling operation is similar to the convolution operation without the kernel weight. Input values in the operating range of pooling map are calculated for a patch of its output feature map base on selected operations, such as calculating the maximum value (max pooling) or average value (average pooling). The pooling operations in most cases are max pooling and average pooling. A frequently used size of pooling layer is 2×2, with stride as 2 and maintaining the size of depth. The zero-padding method is generally used before pooling operation in the case where the input size is not an integer multiple of two.

D. THE PROPOSED FAULT DIAGNOSIS SYSTEM

The fault diagnosis system proposed in this paper first preprocesses the measured information by using PCA and then Andrews function. Then the Andrews function outputs are fed into a specific CNN. The length of the first convolution kernel would better be the same as the dimension of inputs. Its depth can be set to 1 or higher according to the need, e.g. 2-D CNN or 3-D CNN. Subsequent convolutional operations are the same as regular CNN. In order to show the good performance of the proposed fault diagnosis scheme (scheme A), this study also developed a fault diagnosis system based on a conventional neural network (scheme B). In scheme B, the monitored process variables are normalised to zero mean and unit variance and then fed to a single hidden layer neural network. The normalisation equation is as follows:

$$X_{i,p} = \frac{X_i - \bar{X}_{i,\text{normal}}}{X_{\text{std}}} \tag{9}$$

where X_i is the actual measured value for the ith on-line measurement, $X_{i,p}$ is its scaled value, $\bar{X}_{i,\text{normal}}$ and X_{std} are, respectively, the mean and the standard deviation of the normal data.

Scheme A uses Andrews function and CNN to preprocess the monitored data and the extracted features are used as the inputs of a classifier. The classifier of scheme A and scheme B use same method, i.e. a single hidden layer feedforward neural network.

The framework of the proposed fault diagnosis scheme A is shown in Figure 6. Scheme A can be divided into 3 main parts: Andrews function processing, CNN, and classifier. Andrews function and CNN are integrated as a complete process information pre-processing system. The dimension of the final processed feature dataset is determined by Andrews function outputs, i.e. the numbers of t-values, and the adopted convolutional operations.

As previously mentioned, the principal components $X^{'}$ are used in place of the original variables X to remove the impact of variable ordering on the results of Andrews function. The feature dataset $F_x(t)$ is then obtained by using Andrews function from $X^{'}$ by using a specific number of t-values (typically a large value). Then $F_x(t)$ is fed to the input layer of a trained CNN to acquire the final processed features which are then fed to a neural network classifier. The process data used in this study is generated from the simulated CSTR system. Each sample of the monitored data consists of 13 variables. Figure 7 gives the processing of each sample in scheme A. In this case, each sample $X = [x_1, x_2, \ldots, x_{13}]$ is first converted into principal components $X' = [x'_1, x'_2, \ldots, x'_{13}]$, which are then fed into Andrews function, Eq. (2), with 24 values of t uniformly distributed in $[-\pi, \pi]$. Then the offline trained CNN is used to process the Andrews function values. Convolution layer 1 converts the data into a feature matrix with size of 24×24, which are then processed by convolution layers 2 and 3. After the convolutional operations, the data are converted into a feature matrix with size of 4×4. Finally this matrix is transformed into a feature vector with

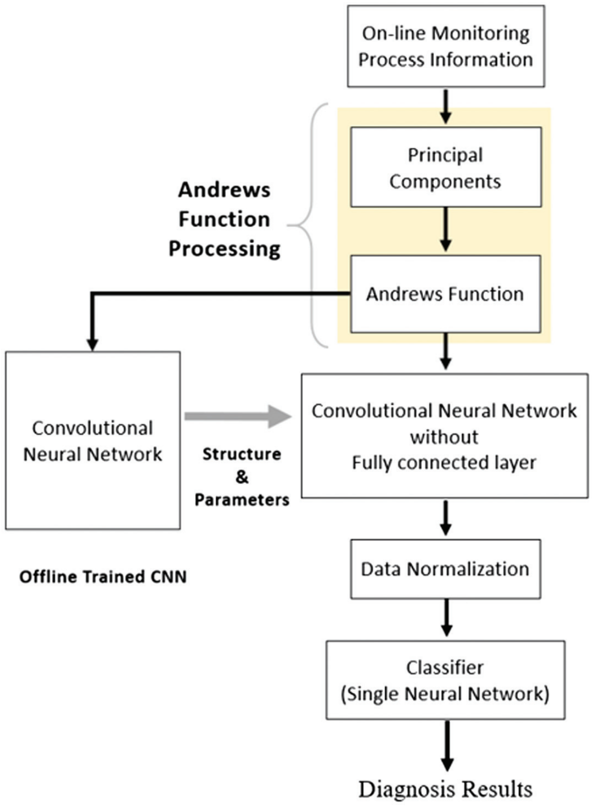
On-line Monitoring Process Information
Principal Components
Andrews Function
Andrews Function Processing
Convolutional Neural Network
Structure & Parameters
Convolutional Neural Network without Fully connected layer
Offline Trained CNN
Data Normalization
Classifier (Single Neural Network)
Diagnosis Results

Fig. 6. A block diagram of the proposed fault diagnosis system.

16 elements before feeding to the input layer of the classifier, i.e. a single hidden layer neural network. Figure 8 shows the baseline scheme (scheme B) for comparison. In this baseline scheme, the monitored process variables are scaled using the means and standard deviations of the normal data and then fed to a neural network. In addition, the proposed method is also compared with integrating Andrews plot with neural network proposed in our previous work [12] (referred to as scheme C here). In scheme C, the selected features in Andrews function are directly fed to a neural network for fault diagnosis [12]. In all the three schemes, the neural networks for final fault classification are single hidden layer neural networks with sigmoid function in both the hidden and output layers. All three networks have 11 output layer neurons with each corresponding to a particular fault. An output layer neuron with an output close to one indicates the corresponding fault occurred while an output close to zero indicates the corresponding fault did not occur. Thus, the normal operating condition is represented by all the 11 output layer neurons having close to zero outputs.

E. PARAMETER SELECTION

In this study, the selected 24 t-values used in Andrews function are uniformly distributed in $[-\pi, \pi]$. Spatial extent of kernel θ_1,θ_2 and θ_3 in convolution layer 1, convolution layer 2 and convolution layer 3 are 1×24, 5×5 and 3×3, respectively. The data for developing the fault diagnosis systems were generated using mechanistic model based process simulation. Simulated historical process operational data under the normal and faulty operation were generated. When a fault is initiated in the process, its impact may not appear immediately. Therefore, when one or more of the process variables exist three times of their normal standard deviations, it is considered that the process is under faulty operation. For the process under normal operation and under each of the 11 considered faults (abrupt faults), 80 samples were collected. Thus a data set of 960 samples are generated for building the fault diagnosis system. These 960 samples are partitioned into a training data set containing 600 samples with 50 samples randomly selected in each

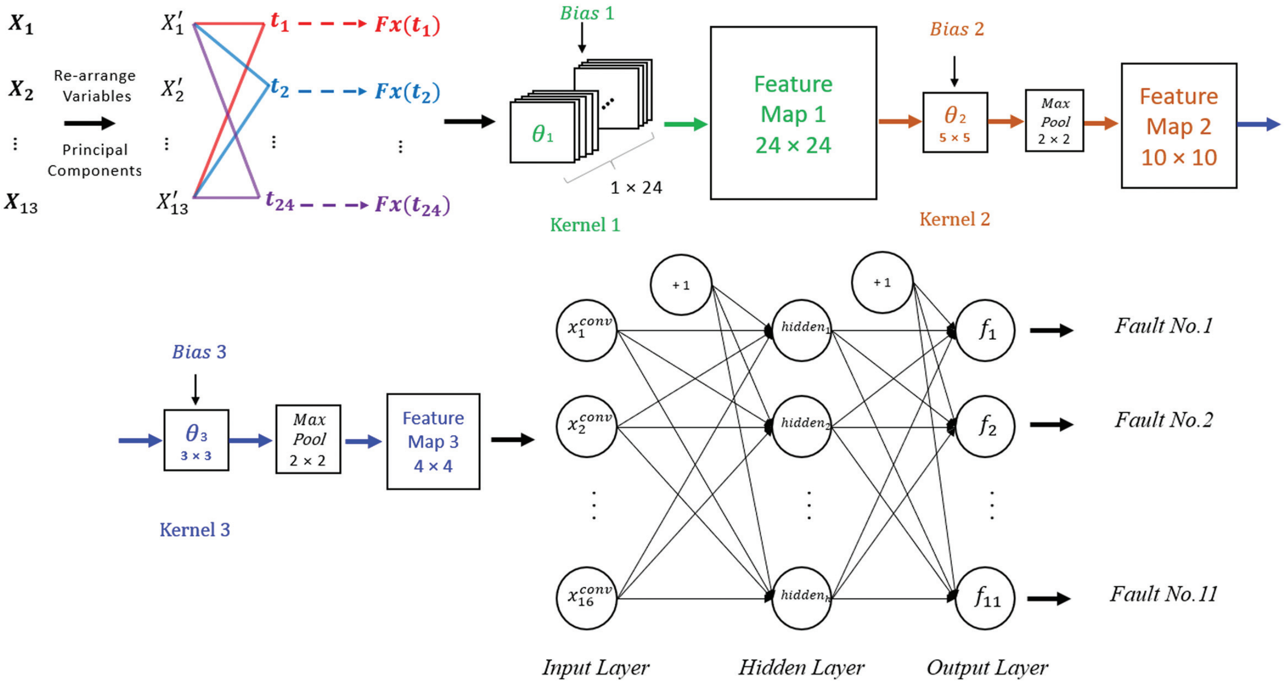

Fig. 7. Processing of each sample in scheme A.

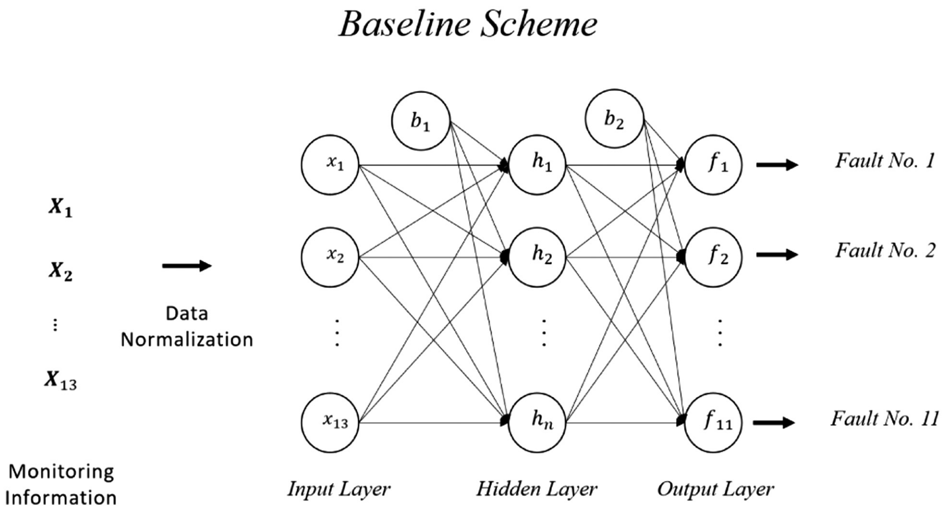

Fig. 8. The baseline scheme (scheme B).

class (normal and 11 faulty classes) and a testing data set containing the remaining 360 samples. The purpose of training data set is for network training, whereas that of the testing data set is for the determination of network structure and implementation of the "early stopping" mechanism in neural network training. To prevent over-fitting, the "early stopping" mechanism stops training when the network error on the testing data reaches minimum. In order to determine the appropriate network structure (e.g. number of hidden neurons), a number of networks with different structures are developed and tested on the testing data. The one giving the best performance is considered to have the appropriate network structure. Table III gives the classification accuracy on the testing data from networks with different numbers of hidden neurons in the classifier, i.e. a single hidden layer neural network in proposed diagnosis scheme A. Table IV gives the accuracy on the testing data of different numbers of hidden neurons in diagnosis scheme B. In Tables III and IV, the highest classification accuracy is marked with bold font. The numbers of neurons in different layers of scheme A are given in Table V.

Table III. Accuracy on testing data with different numbers of hidden neurons in scheme A

Numbers of HN	Accuracy	Numbers of HN	Accuracy
10	83.89%	16	92.78%
11	88.78%	**17**	**97.22%**
12	87.78%	18	96.67%
13	93.89%	19	92.78%
14	96.11%	20	85.83%
15	90.56%	21	89.17%

Table IV. Accuracy on testing data with different numbers of hidden neurons in scheme B

Numbers of HN	Accuracy	Numbers of HN	Accuracy
10	65.28%	16	62.78%
11	66.39%	**17**	**76.67%**
12	71.67%	18	71.67%
13	68.33%	19	68.33%
14	64.72%	20	58.06%
15	75%	21	61.67%

Table V. Numbers of neurons in scheme A

Layer	Kernel size	Neuron
Input of CNN		24
CONV 1	1 × 24	24 × 24
CONV 2	5 × 5	20 × 20
MAX POOL	2 × 2	10 × 10
CONV 3	3 × 3	8 × 8
MAX POOL	2 × 2	4 × 4
Input of Classifier		16
Hidden		17
Output		11

IV. RESULTS

The proposed fault diagnosis scheme A, the traditional diagnosis scheme B, and scheme C where the features of Andrews function outputs are manually selected [12] are applied to the simulated CSTR system and compared in terms of the diagnosis performance. Fault diagnosis is indicated by the neural network outputs. When the neural network output corresponding to a fault exceeds the diagnosis threshold (set to 0.8) while other neural network outputs remain lower than 0.2 (i.e. close to 0), then a fault is diagnosed. When the neural network output corresponding to a fault exceeds the advance warning threshold (set to 0.4) while other neural network outputs remain lower than 0.2, then an advance warning is issued for the corresponding fault. Setting the diagnosis threshold as 0.8 is based on the following considerations. The output layer neurons use the sigmoid activation function with output bounded between 0 and 1. From the plot of the sigmoid function output, it can be seen that the output flatten off when the output is greater than 0.8. Thus setting the threshold higher than 0.8 would delay fault diagnosis. On the other hand, lowering this threshold below 0.8, there could increase the chance of incorrect diagnosis. These threshold values were also used in [11]. Diagnosis performance is generally evaluated by the diagnosis speed and accuracy. The advance warning time and the diagnosis time are defined, respectively, as the time periods elapsed from a fault occurred to a correct advance warning being issued and to it being successfully diagnosed.

A. DIAGNOSTIC PERFORMANCE ON ABRUPT FAULTS

Table VI gives 3 sets of fault relative magnitudes (Mag. 1 to Mag. 3) for the 11 abrupt faults. These magnitudes are different from those in the training data. Thus, these faults are used as unseen validation data to evaluate the performance of fault diagnosis scheme in diagnosing abrupt faults with unseen magnitudes.

Table VII gives the diagnosis times of these abrupt faults shown in Table VI. The performance from scheme C is taken from [12]. The results indicate that all three fault diagnosis schemes successfully diagnosed all these 11 abrupt faults. Diagnosis speed comparison in abrupt faults shows that the proposed fault diagnosis system (scheme A) successful diagnosed the faults 8.6 s earlier on average than the conventional neural network based fault diagnosis system (scheme B) and 3.15 s earlier than scheme C. The difference between schemes A and C is that, instead of selecting 11 important features in Andrews function in scheme C [12], 24 t-values uniformly distributed in $[-\pi, \pi]$ are used in Andrews function and a CNN is then used to further extract features from Andrews function outputs in scheme A. Thus, the improvement of fault diagnosis speed in the proposed scheme A is mostly likely due to that Andrews function and CNN have extracted features in the process data and this helps the subsequent neural network in classifying (diagnosing) the faults.

Table VI. Relative magnitudes of abrupt faults

Faults	Mag. 1	Mag. 2	Mag. 3
1	1.67%	2.33%	3.33%
2	1.67%	2.00%	2.33%
3	6.50%	7.50%	10.00%
4	4.56%	6.78%	11.22%
5	9.09%	14.29%	19.49%
6	38.73%	49.83%	66.48%
7	16.76%	27.86%	33.41%
8	2.10%	3.65%	4.32%
9	2.46%	4.97%	7.47%
10	2.54%	3.79%	5.03%
11	6.25%	8.75%	12.50%

Figures 9 and 10 show, respectively, the diagnosis outputs from scheme A and scheme B in diagnosing abrupt fault no. 1, with a relative magnitude of 1.67%. In Figures 9 and 10, as well as the latter plots, F1 to F11 represent the 11 network outputs which correspond to Fault No. 1 to No. 11 respectively. In this study, all the abrupt faults are initiated at 40 s and the incipient faults are initiated at 0 s. In Figs. 9 and 10, as well as the subsequent plots, the upper dash–dotted straight lines indicate the diagnosis threshold and the lower dash–dotted lines represent the advance warning threshold. A fault diagnosis or advance warning result is issued only when the corresponding neural network output exceeds the corresponding thresholds for a consecutive number of samples. Any isolated samples exceeding the thresholds can be ignored as "noise". Scheme A successfully diagnosed the fault at 12 s after it occurred (at 40 s) as shown in Fig. 9. The overall responses of the output curves are quite stable. Figure 10 shows that the network output from scheme B representing fault no. 1 responded quickly, when the fault occurred, but then a period of oscillation occurred in the region of advance warning. Then it rose and remained over the diagnosis threshold (0.8). At the same time, the network outputs corresponding to other faults were close to 0. Therefore, scheme B successfully diagnosed the fault at 48 s after it occurred. The proposed scheme A diagnosed this particular fault 36 s earlier than scheme B.

Figures 11 and 12 show, respectively, the neural network outputs of scheme A and scheme B in diagnosing abrupt fault no. 8 with the fault relative magnitude of 2.10%. It can be seen from Fig. 11 that the network output representing fault no. 8 from scheme A responded quickly and soon exceeded the upper threshold (0.8). Hence, the proposed fault diagnosis system successfully diagnosed this fault 16 s after it occurred. Figure 12 indicates that the output corresponding to fault no. 8 in scheme B has a relatively long period of oscillations before crossing over

Table VII. Fault diagnosis time (s) for abrupt faults

	Scheme A			Scheme B			Scheme C		
Fault No.	Mag. 1	Mag. 2	Mag. 3	Mag. 1	Mag. 2	Mag. 3	Mag. 1	Mag. 2	Mag. 3
1	12	4	4	48	24	12	24	12	4
2	16	8	4	32	12	12	20	8	4
3	16	8	8	44	32	28	36	28	12
4	32	8	8	32	28	8	32	12	4
5	20	12	4	36	24	8	32	16	8
6	36	32	16	12	8	8	60	28	12
7	32	16	8	52	36	16	36	16	8
8	16	8	8	52	20	4	24	8	8
9	24	12	12	40	8	8	20	8	8
10	20	12	8	36	20	4	20	8	4
11	8	8	8	12	8	8	16	8	8
Average	13.58			22.18			16.73		

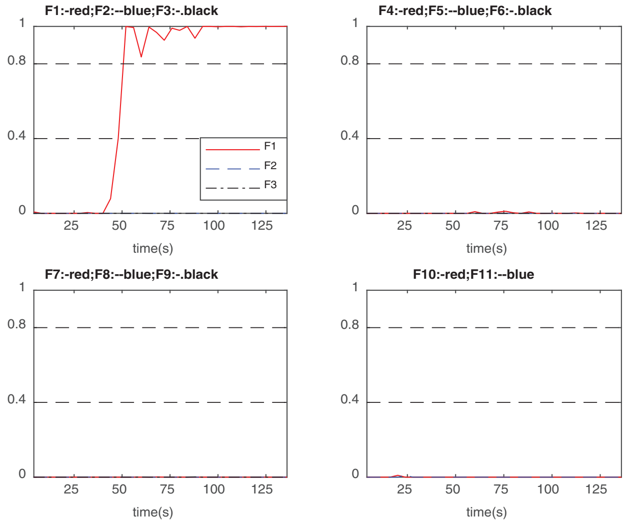

Fig. 9. Outputs of scheme A in diagnosing abrupt fault no.1 (relative magnitude: 1.67%).

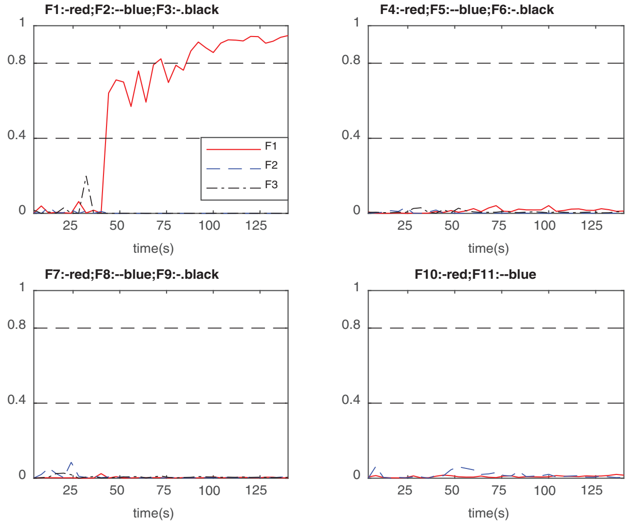

Fig. 10. Outputs of scheme B in diagnosing abrupt fault no.1 (relative magnitude: 1.67%).

the upper threshold. Scheme B successfully diagnosed the fault at 52 s after the fault occurred. For this fault, scheme A diagnosed the fault 36 s earlier than scheme B.

B. DIAGNOSTIC PERFORMANCE ON INCIPIENT FAULTS

Table VIII gives 3 sets of fault developing speeds for incipient faults. Table IX gives the diagnosis times of the three schemes for the incipient faults given in Table VIII. The results show that all three fault diagnosis schemes successfully diagnosed all these incipient faults. Diagnosis speed comparison in incipient faults shows that the average diagnosis time of the proposed scheme A is 18.43 s shorter than that of scheme B and 4.61 s shorter than that of scheme C.

Figures 13 and 14 show, respectively, the diagnosis outputs of scheme A and scheme B when incipient fault no. 3 with a fault developing speed of $\gamma = -1.29 \times 10^{-4}(\mathrm{s}^{-1})$ occurred. As can be seen from Fig. 13, the output

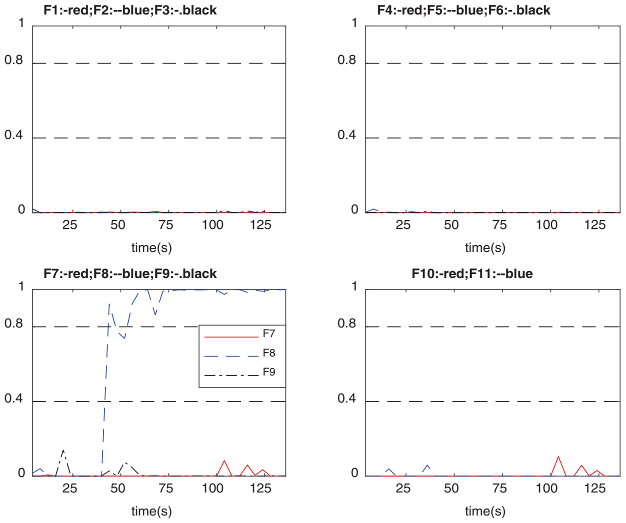

Fig. 11. Outputs of scheme A in diagnosing abrupt fault no. 8 (relative magnitude: 2.10%).

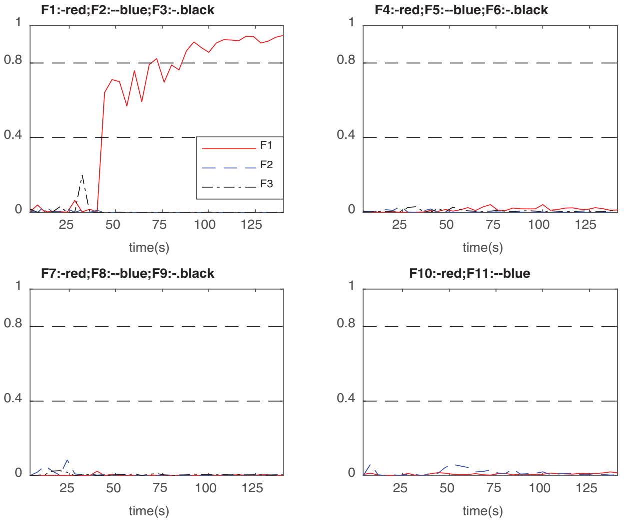

Fig. 12. Outputs of scheme B in diagnosing abrupt fault no. 8 (relative magnitude: 2.10%).

Table VIII. Fault developing speeds of incipient faults

Faults	γ_1 (s^{-1})	γ_2 (s^{-1})	γ_3 (s^{-1})
1	-6.67×10^{-5}	-1.67×10^{-4}	-3.67×10^{-4}
2	6.67×10^{-5}	1.67×10^{-4}	3.67×10^{-4}
3	-1.29×10^{-4}	-6.29×10^{-4}	-9.29×10^{-4}
4	-1.67×10^{-4}	-3.67×10^{-4}	-6.67×10^{-4}
5	1.12×10^{-4}	5.12×10^{-4}	9.12×10^{-4}
6	6.67×10^{-4}	1.67×10^{-3}	3.67×10^{-3}
7	-3.12×10^{-4}	-1.12×10^{-3}	-3.12×10^{-3}
8	7.13×10^{-5}	1.13×10^{-4}	6.13×10^{-4}
9	-1.23×10^{-4}	-5.23×10^{-4}	-9.23×10^{-4}
10	6.71×10^{-5}	1.71×10^{-4}	6.71×10^{-4}
11	-6.67×10^{-5}	-6.67×10^{-4}	-1.67×10^{-4}

Table IX. Fault diagnosis time (s) for incipient faults

	Scheme A			Scheme B			Scheme C		
Fault No.	γ_1 (s^{-1})	γ_2 (s^{-1})	γ_3 (s^{-1})	γ_1 (s^{-1})	γ_2 (s^{-1})	γ_3 (s^{-1})	γ_1 (s^{-1})	γ_2 (s^{-1})	γ_3 (s^{-1})
1	68	36	32	92	64	48	96	52	40
2	68	48	40	88	56	44	80	48	44
3	80	40	36	152	68	52	104	48	40
4	68	48	32	84	76	44	76	48	40
5	108	60	52	152	72	60	116	64	44
6	112	92	60	156	136	88	132	100	80
7	128	92	56	160	100	64	144	96	60
8	56	48	36	96	80	40	88	76	32
9	112	108	72	136	72	48	88	44	44
10	92	68	36	124	80	40	96	68	44
11	124	84	56	144	96	44	128	88	52
Average	68.12			86.55			72.73		

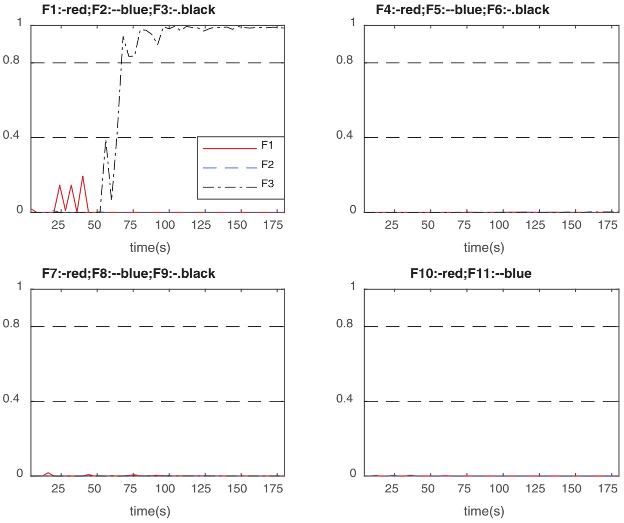

Fig. 13. Outputs of scheme A under incipient fault no. 3 ($\gamma = -1.29 \times 10^{-4}$ s^{-1}).

representing fault no. 3 rises rapidly after a period of damage accumulation of the fault, whereas all other network outputs are much lower than 0.2. The proposed scheme A successfully diagnosed this fault at 68 s after it occurred. As can be seen from Fig. 14, the network output corresponding to fault no. 3 responds quickly but has a long duration of fluctuations. Finally, scheme B gave the correct diagnosis result at 152 s after the fault occurred. For this incipient fault, the diagnosis time of scheme A is 84 s shorter than that of scheme B. The network outputs in scheme A appear to be more stable than those in scheme B.

Figures 15 and 16 show, respectively, the network outputs of schemes A and B when incipient fault no. 8 occurred with a fault developing speed of $\gamma = 7.13 \times 10^{-5}(\text{s}^{-1})$. As can be seen from Figs. 15 and 16, after the initial accumulation fault impact, the network output representing fault No. 8 in scheme A raises quickly to close to 1, but that in scheme B raises slowly. All other outputs are close to 0. The output curves in scheme A are very smooth. Scheme A and scheme B successfully diagnosed the fault at 56 s and 96 s, respectively, after the fault occurrence. For this particular fault, the diagnosis time of the proposed scheme A is 40 s shorter than that of the conventional scheme B.

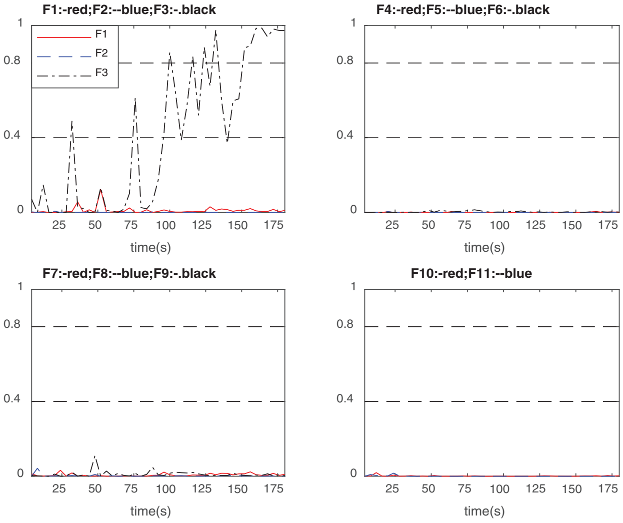

Fig. 14. Outputs of scheme B under incipient fault no. 3 ($\gamma = -1.29 \times 10^{-4}$ s^{-1}).

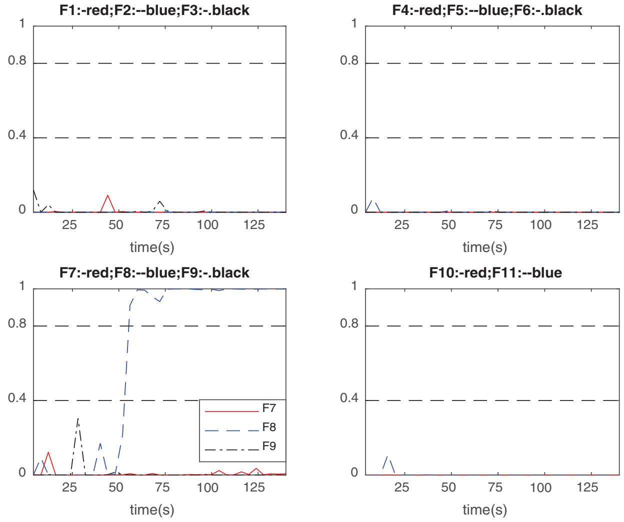

Fig. 15. Outputs of scheme A under incipient fault no. 8 ($\gamma = 7.13 \times 10^{-5}$ s^{-1}).

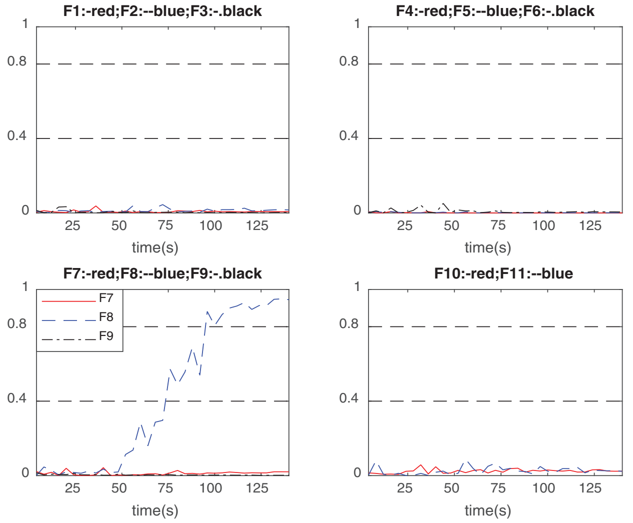

Fig. 16. Outputs of scheme B under incipient fault no. 8 ($\gamma = 7.13 \times 10^{-5}$ s^{-1}).

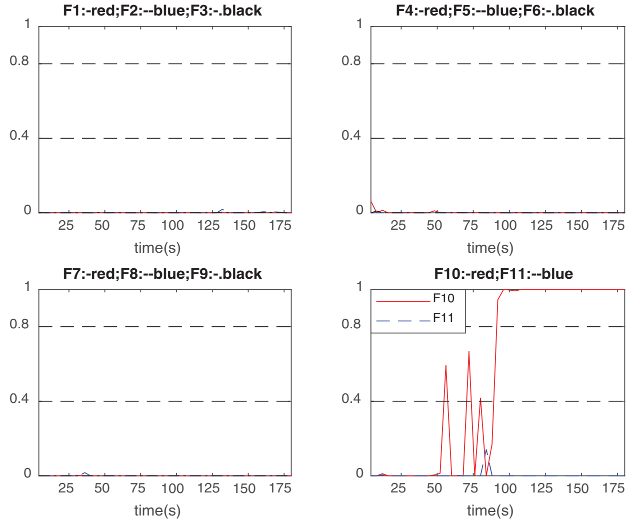

Fig. 17. Outputs of scheme A under incipient fault no. 10 ($\gamma = 6.71 \times 10^{-5}$ s^{-1}).

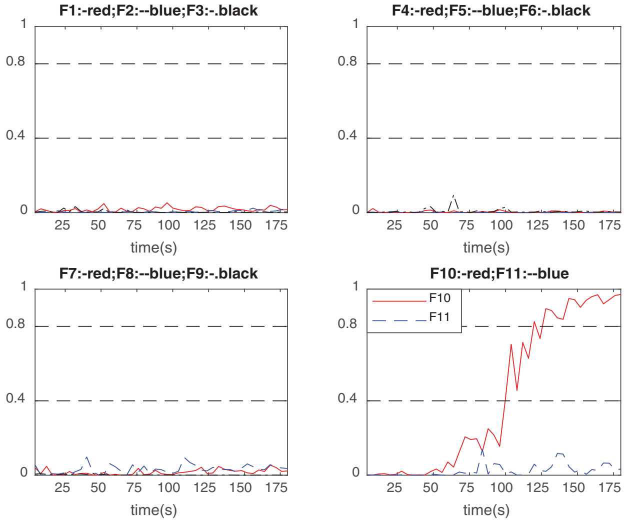

Fig. 18. Outputs of scheme B under incipient fault no. 10 ($\gamma = 6.71 \times 10^{-5}$ s^{-1}).

Figures 17 and 18 show, respectively, the network outputs of scheme A and scheme B when incipient fault no. 10 occurred with a fault developing speed of $\gamma = 6.71 \times 10^{-5}(s^{-1})$. Figures 17 and 18 show that, after a period of fault development, the network output representing fault No. 10 in scheme A raises quickly to close to 1, and that in scheme B raises relatively slowly. The network output representing fault No. 10 in scheme A has some fluctuations before reaching close to 1. Scheme A and scheme B successfully diagnosed the fault at 92 s and 124 s, respectively, after the fault occurred. For this fault, the diagnosis time of the proposed scheme A is 32 s shorter than that of scheme B.

V. CONCLUSIONS

This paper proposes a fault diagnosis method for industrial processes by integrating Andrews plots, convolutional neural network, and neural networks. Features within the on-line monitored measurements are extracted by using Andrews plot. To address the uncertainty of setting the proper dimension of extracted features in Andrews plot, a convolutional neural network is used to further process the extracted features. The CNN outputs are then used as the inputs to a single hidden layer neural network to classify extracted features into various classes, i.e. the diagnosis results. The proposed fault diagnosis system is compared with a conventional neural network based fault diagnosis system, as well as integrating Andrews plot with neural network and manual selection of features in Andrews function outputs, through application to a simulated CSTR system. It is shown that the proposed fault diagnosis system performs better than the conventional neural network based fault diagnosis system and manual selection of features in Andrews function outputs in diagnosing both abrupt and incipient faults. Future works will consider integration of Andrews function with alternative deep neural networks such as light-weight CNN for on-line process fault diagnosis.

ACKNOWLEDGEMENTS

Partial financial supports from the European Commission (Project No.: PIRSES-GA-2013-612230) and National Natural Science Foundation of China (project No.: 61673236) are gratefully acknowledged.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

ETHICS OF HUMAN SUBJECT PARTICIPATION

Not applicable.

References

1. V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part I: Quantitative model-based methods", Comput. Chem. Eng., vol. 27, no. 3, pp. 293–311, 2003.

2. V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, “A review of process fault detection and diagnosis: part II: qualitative models and search strategies”, Comput. Chem. Eng., vol. 27, no. 3, pp. 313–326, 2003.
3. V. Venkatasubramanian, R. Rengaswamy, S.N. Kavuri, and K. Yin, “A review of process fault detection and diagnosis: part III: process history based methods”, Comput. Chem. Eng., vol. 27, no. 3, pp. 327–346, 2003.
4. J. Zhang, and P. D. Roberts, “Process fault diagnosis with diagnostic rules based on structural decomposition”, J. Process Control, vol. 1, no 5, pp. 259–269, 1991.
5. J. F. MacGregor and T. Kourti, “Statistical process control of multivariate processes”, Control Eng. Pract., vol. 3, no. 3, pp. 403–414, 1995.
6. S. J. Qin, “Survey on data-driven industrial process monitoring and diagnosis”, Annu. Rev. Control, vol. 36, no. 2, pp. 220–234, 2012.
7. J. MacGregor and A. Cinar, “Monitoring, fault diagnosis, fault-tolerant control and optimization: data driven methods”, Comput. Chem. Eng., vol. 47, pp. 111–120, 2012.
8. Z. Ge, “Review on data-driven modeling and monitoring for plant-wide industrial processes”, Chemometr Intell Lab Syst, vol. 171, pp. 16–25, 2017.
9. Y. Wang, Y. Si, B. Huang, and Z. Lou, “Survey on the theoretical research and engineering applications of multivariate statistics process monitoring algorithms: 2008–2017”, Can. J. Chem. Eng., vol. 96, no. 10, pp. 2073–2085, 2018.
10. S. Wang and J. Zhang, “An intelligent process fault diagnosis system integrating Andrews plot, PCA and neural networks”, in 2019 25th Int. Conf. Autom. Comput. (ICAC2019), Lancaster, UK, 5–7 September, IEEE, pp. 1–6, 2019.
11. J. Zhang, “Improved on-line process fault diagnosis through information fusion in multiple neural networks”, Comput. Chem. Eng., vol. 30, no. 3, pp. 558–571, 2006.
12. S. Wang and J. Zhang, “An intelligent process fault diagnosis system based on neural networks and Andrews plot”, Processes, vol. 9, no. 9, article 1659, 2021.
13. T. Mao, Y. Zhang, Y. Ruan, H. Gao, H. Zhou, and D. Li, “Feature learning and process monitoring of injection molding using convolution-deconvolution auto encoders”, Comput. Chem. Eng., vol. 118, pp. 77–90, 2018.
14. W. Sun, A. R. C. Paiva, P. Xu, A. Sundaram, and R. D. Braatz, “Fault detection and identification using Bayesian recurrent neural networks”, Comput. Chem. Eng., vol. 141, article 106991, 2020.
15. S. Wang and J. Zhang, “Improved process fault diagnosis by using neural networks with Andrews plot and autoencoder”, in 2020 IEEE 18th Int. Conf. Ind. Inform. (INDIN2020), University of Warwick, UK, 20–23 July 2020, 1, pp. 787–792, 2020.
16. F. Jia, Y. Lei, L. Guo, J. Lin, and S. Xing, “A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines”, Neurocomputing, vol. 272, pp.619–628, 2018.
17. X. Yu, Z. Zhao, X. Zhang, Q. Zhang, Y. Liu, C. Sun, and X. Chen, “Deep-learning-based open set fault diagnosis by extreme value theory”, IEEE Trans. Ind. Inform., vol. 18, no. 1, pp. 185–196, 2022.
18. Z. Zhang, X. Xu, W. Gong, Y. Chen, and H. Gao, “Efficient federated convolutional neural network with information fusion for rolling bearing fault diagnosis”, Control Eng. Pract., vol. 116, Article number 104913, 2021.
19. R. Arunthavanathan, F. Khan, S. Ahmed, and S. Imtiaz, “A deep learning model for process fault prognosis”, Process Saf. Environ. Protect., vol. 154, pp. 467–479, 2021.
20. W. Wang, Y. Lei, T. Yan, N. Li, and A. Nandi. “Residual convolution long short-term memory network for machines remaining useful life prediction and uncertainty quantification”, J. Dyn. Monit. Diagnost., vol. 1, no. 1, pp. 2–8, 2021. doi: 10.37965/jdmd.v2i2.43.
21. N. H. Spencer, “Investigating data with Andrews plots”, Soc. Sci. Comput. Rev., vol. 21, no. 2, pp. 244–249, 2003.
22. R. N. Khattree and D. N. Naik, “Andrews plots for multivariate data: Some new suggestions and applications”, J. Stat. Plan. Inference, vol. 100, no. 2, pp. 411–425, 2002.
23. H. Wu and J. Zhao, “Deep convolutional neural network model based chemical process fault diagnosis”, Comput. Chem. Eng., vol. 115, pp. 185–197, 2018.
24. K. Wang, C. Shang, L. Liu, Y. Jiang, D. Huang, and F. Yang, “Dynamic soft sensor development based on convolutional neural networks”, Ind. Eng. Chem. Res., vol. 58, no. 26, pp. 11521–11531, 2019.
25. L. Wen, X. Li, L. Gao, and Y. Zhang, “A new convolutional neural network-based data-driven fault diagnosis method”, IEEE Trans. Ind. Electron., vol. 65, no. 7, pp. 5990–5998, 2017.
26. X. Guo, L. Chen, and C. Shen, “Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis”, Measurement, vol. 93, pp. 490–502, 2016.
27. J. Zhang and P. D. Roberts, “On-line process fault diagnosis using neural network techniques”, Trans. Instit. Meas. Control, vol. 14, no. 4, pp. 179–188, 1992.
28. J. Zhang, E. B. Martin, and A. J. Morris, “Fault-detection and diagnosis using multivariate statistical techniques”, Chem. Eng. Res. Design, vol. 74, no. 1, pp.89–96, 1996.
29. M. Misra, H. H. Yue, S. J. Qin, and C. Ling, “Multivariate process monitoring and fault diagnosis by multi-scale PCA”, Comput. Chem. Eng., vol. 26, no. 9, pp. 1281–1293, 2002.
30. D. Andrews, “Plots of high-dimensional data”, Biometrics, vol. 28, no. 1, pp. 125–136, 1972.
31. S. Boonprong, C. Cao, P. Torteeka, and W. Chen, “A novel classification technique of Landsat-8 OLI image-based data visualization: the application of Andrews' plots and fuzzy evidential reasoning”, Remote Sens., vol. 9, no. 5, p. 427, 2017.
32. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition”, Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
33. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks”, Adv. Neural Inf. Process. Syst., vol. 25, pp. 1097–1105, 2012.
34. K. Simonyan, and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, arXiv preprint arXiv: 1409.1556, 2014.
35. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition”, in Proc. IEEE Conf. Comput. Vision Pattern Recogn., pp. 770–778, 2016.
36. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning”, Nature, 521(7553), pp.436–444, 2015.
37. J. Bouvrie, Notes on Convolutional Neural Networks. Tech. Rep., MIT, Cambridge, MA, USA, 2006.
38. G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks”, Science, vol. 313, no. 5786, pp. 504–507, 2006.