Intelligent Process Fault Diagnosis Using Feature Extraction and Neural Network Techniques

Shengkai Wang

A Thesis submitted for the degree of Doctor of Philosophy

School of Engineering

Newcastle University

United Kingdom

October 2021

Abstract

Fault diagnosis has been an active research subject in the field of process systems engineering due to the advances in industrial technology. However, reliability and speed of fault diagnosis is still a common issue with the current fault diagnosis methods. This research work aims to propose some novel hybrid fault diagnosis systems by integrating multiple techniques to achieve a positive promotion in industrial process monitoring. This thesis proposes six hybrid intelligent fault diagnosis systems through integrating several techniques including Andrews function, artificial neural network, principal component analysis, qualitative trend analysis, autoencoder, stacked autoencoder, convolutional neural network, and multiple neural networks with information fusion. The works are mainly focusing on data pre-processing to extract the features from measured information and then using the extracted features for fault diagnosis. The final extracted features are used as inputs to a neural network to obtain the diagnosis results. Applications to a simulated continuous stirred tank reactor (CSTR) process evidence that the diagnosis speed and reliability can be improved by using the proposed diagnosis schemes, which are compared with conventional neural network based fault diagnosis schemes.

The main contributions include the following. (1), it is the first time that Andrews plot has been exploited and integrated with neural networks for fault detection and diagnosis. The features extracted by Andrews plot would help the subsequent fault diagnosis by neural networks. (2), a method for determining the important features in Andrews plot is proposed. (3), reducing uncertainty associated with parameter selection in Andrews plot by integrating with principal component analysis, qualitative trend analysis, convolutional neural network, and autoencoder has been proposed.

The proposed schemes include the following: (1) use the Andrews function to pre-process the measured process information and feed the extracted features into a neural network, as a classifier, to obtain the diagnosis outputs.

(2) Use the Andrews function to pre-process the measured process information and then convert the extracted features into qualitative trend data before feeding into a neural network to obtain the diagnosis outputs. (3) Use the Andrews function to pre-process the measured process information and then use principal component analysis to reduce the dimensions of extracted features. The retained principal components are fed into a neural network to obtain the diagnosis outputs.

(4) Use the Andrews function to pre-process the measured process information and then reduce the dimension of the extracted features via an autoencoder. The low dimensional features from the autoencoder are fed into a neural network.

(5) Use the Andrews function to pre-process the measured process information and feed the extracted features into a stacked neural network combining multiple neural networks.

(6) Use the Andrews function method to process each sample of the monitored information and then use convolutional neural network to extract features which are fed to a neural network for fault diagnosis

Acknowledgement

First appreciation to Dr. Jie Zhang. His guidance, suggestions and assistance in my PhD career is indelible.

Special thanks to my family and Qibing Jin for supporting my doctoral study.

Many thanks to Dr. Tengfei Wan, Chen Li, Rong Zhang.

Many thanks to Xinyi Zhang, Minxuan Cai, Siqi Su.

Many thanks to Haoxu Lee, Yongling Dai.

Extend thanks to the staffs and students at the School of Engineering.

Content

List of Figures

List of Tables

Abbreviations and Acronyms

AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CReLU	Concatenated Rectified Linear Units
CSTR	Continuous Stirred Tank Reactor
ELU	Exponential Linear Unit
FDR	Fault Detection Rate
ICA	Independent Component Analysis
k-NN	k-Nearest Neighbours
KPCA	Kernel Principal Component Analysis
LDA	Linear Discriminant Analysis
LLE	Local Linear Embedding
Mag.	Relative Fault Magnitude
MDR	Missed Detection Rate
MSE	Mean Square Error
PCA	Principal Component Analysis
PLS	Partial Least Squares
PReLU	Parametric Rectified Linear Unit
ReLU	Rectified Linear Unit
SVD	Singular Value Decomposition
SVM	Support Vector Machine

Mathematical Operators

Σ	summation
$\lvert\ \rvert$	modulus
exp	exp exponential function
$\prod$	product over a set of terms
$\lVert\ \rVert$	norm

Chapter 1. Introduction

1.1. Background

The breakthroughs and advances in industrial technology have made high-yield modern industrial production processes with intricate functionalities. These improvements have enhanced the automation and expanded the production scale during the past decades. The increasing complexity of industrial processes also brings in associated risk of potential faults of various components. In particular, faults hidden in insignificance could possibly cumulate colossal damages. Faults are hard to completely eliminate in chemical industry. Undetected faults in an industrial production process are usually accompanied by large hidden risks, which will cause distressingly serious consequences, and the subsequent indirect impacts. Degradation in product quality, as a quintessentially consequence, can ruin the brand reputation and corporate identity. Particularly to small-and-medium-sized enterprises, this can perhaps lead to the capital chain rupture and disrupt the development of the enterprise. Environmental contamination is also a typical effect of chemical process accidents. Authoritativeness of local authorities may be challenged due to the duty of environmental conservation, and enterprises may also risk heavy penalties. Disastrous consequences, such as casualties, usually bring inestimable lost. With the rapid growing engineering technology, bygone faults solved, but faults also develop vigorously surrounding the more sophisticated functionalities and intricate processes, especially in enhanced automation and expanded high-yield production scale. Production process under these complicated circumstances is increasingly hard to achieve a satisfactory performance to against the hidden risks through traditional process monitoring approaches. In general, with the advancement of industrial technology, the improvement of social environmental awareness, and the increased requirement for high-quality products, chemical industrial process fault diagnosis is evolving into an imperative study. This challenging conundrum attracts attention from numerous researchers in different areas.

Prior relevant researches contributed a myriad of attempt to promote the fault diagnosis in terms of speed, robustness, and reliability. The past fault diagnosis approaches can approximately be divided into three categories: model based approaches, knowledge based approaches, and data analysis based approaches. A fault diagnosis system can achieve excellent performance by using model based approaches with known precise mathematical models. Model based approach requires the knowledge of the relationships between particular faults and parameters or states in the estimated model. Knowledge based approaches develop the fault diagnosis system by using the available knowledge of the process behaviour and the experience of the process operations. These diagnosis systems can be achieved without establishing an accurate mathematical model, by properly using logical functions and fuzzy rules to carry out fuzzy reasoning about the process state. Knowledge based approaches requires various mechanistic knowledge, process principle knowledge and empirical knowledge covering the system. The data analysis based approaches diagnose the process faults by analyzing and processing the process operational data, without knowing the precise model and knowledge of the system. A fault diagnosis system developed by data analysis based approaches mainly analyze offline historical data. Artificial intelligence (AI) techniques in the big data era have high value for industrial process fault diagnosis. The data analysis based approaches require the utilization of a large amount of historical data.

Fault diagnosis systems developed by model based approaches and knowledge based approaches are suitable for processes with adequate experiences or with underlying theories from physics or mathematics. However, in the chemical industries, both approaches are hard to implement due to the complicated processes with uncertainties and demanding requirement on monitoring capabilities. Currently, most companies produce and store large amounts of historical process data every day. The process information generated during operation under these normal conditions and under specific fault conditions provides a good prerequisite for the application of data analysis based approaches to industry. The improvement of information processing

capabilities of computers also provides technical support the fault diagnosis system based on offline data to process the online information. Note that neither the techniques of big data analysis nor artificial intelligence is omnipotent for process monitoring, but the historical and real-time process information are helpful to improve the safety of process operation. Therefore, a hybrid fault diagnosis system combination with multiple techniques has extremely high research value and significance in process industries.

1.2. Aims and Objectives

This research work aims to propose some novel hybrid fault diagnosis systems by integrating with multiple techniques to achieve a positive promotion in industrial process monitoring. In the prior works, several process monitoring techniques based on data analysis achieved varying degrees of success in the industrial process fault diagnosis. The hybrid intelligent fault diagnosis systems developed in this thesis generally contain two subsystems: the former subsystem pre-processes the measured process information by feature extraction algorithms into a feature vector as the inputs for easier comparison and classification; the later subsystem classifies the processed features by an intelligent classifier to obtain the outputs for process state analysis. Among a variety of data analysis approaches, two notable methods are adopted in this thesis to deal with the high-dimensional data of process information: neural network based fault diagnosis method and multivariate statistical analysis based fault diagnosis method.

In order to achieve this aim, the following objectives need to be accomplished.

1). Investigation of Andrews plot in information pre-processing to aid process fault diagnosis.

2). Investigation of parameter selection in Andrews plot for effective process fault diagnosis.

3). Investigation of the combination of Andrews plot, principal component analysis, qualitative trend analysis, and autoencoder for effective process fault diagnosis.

4). Investigation of using deep neural networks, such as convolutional neural network and stacked autoencoder, to overcome potential problems in Andrews function parameter selection.

5). Compare the developed hybrid intelligent fault diagnosis systems with conventional neural network based fault diagnosis system on a simulated continuous stirred tank reactor (CSTR) system.

The following 6 different novel fault diagnosis systems are developed in this thesis.

Scheme A: an enhanced intelligent neural network based online process fault diagnosis system developed by integrating Andrews plot and neural network techniques. A method for determining the important features in Andrews function is also proposed.

Scheme B: a neural network based fault diagnosis system developed through the fusion of Andrews function and qualitative trend analysis.

Scheme C: a neural network based fault diagnosis system developed through the fusion of Andrews function and dimensional reduction by principal component analysis.

Scheme D: an intelligent fault diagnosis system based on the techniques of Andrews plot, autoencoder, and neural network.

Scheme E: a neural network based fault diagnosis system integrating Andrews function with multiple neural networks combined through information fusion.

Scheme F: a fault diagnosis method for industrial processes by integrating Andrews plots, convolutional neural network, and neural networks.

1.3. Thesis Contributions

This thesis proposed 6 different schemes to build hybrid intelligent fault diagnosis system. All the proposed schemes have been applied into CSTR to prove their feasiblity and performance. The contributions as follow:

1. It is first time the Andrews plot has been exploited and integrated with neural networks for fault detection and diagnosis. The features extracted by Andrews plot would help the subsequent fault diagnosis by neural networks.

2. A method for determining the important features in Andrews plot is proposed.

3. reducing uncertainty associated with parameter selection in Andrews plot by integrating with principal component analysis, qualitative trend analysis, convolutional neural network, and autoencoder has been proposed.

The following journal and conference papers are based on the work presented in this thesis.

Wang, S. and Zhang, J., 2021. An Intelligent Process Fault Diagnosis System Based on Neural Networks and Andrews Plot. *Processes*, *9*(9), article 1659.

S. Wang and J. Zhang, 2022, An Intelligent Process Fault Diagnosis System based on Andrews Plot and Convolutional Neural Network, *Journal of Dynamics, Monitoring and Diagnostics*, in press.

Wang, S. and Zhang, J., 2019, August. Enhanced process fault diagnosis through integrating neural networks and Andrews plot. In *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR2019)* (pp. 36-41). IEEE.

Wang, S. and Zhang, J., 2019, September. An Intelligent Process Fault Diagnosis System Integrating Andrews Plot, PCA and Neural Networks. In *2019 25th International Conference on Automation and Computing (ICAC2019)* (pp. 1-6). IEEE.

Wang, S. and Zhang, J., 2020, July. Improved Process Fault Diagnosis by Using Neural Networks with Andrews Plot and Autoencoder. In *2020 IEEE 18th International Conference on Industrial Informatics (INDIN2020)* (Vol. 1, pp. 787-792). IEEE.

Wang, S. and Zhang, J., 2021, March. A Fault Diagnosis Strategy based on Qualitative Trend Analysis Integrating Andrews Plot for Industrial Processes. In *2021*

22nd IEEE International Conference on Industrial Technology (ICIT2021) (Vol. 1, pp. 973-978). IEEE.

1.4. Thesis Structure

The structure of this thesis is organised as follows. Chapter 2 gives the literature review. It reviews several related techniques including information pre-process methods and feature classification methods. In this thesis, the proposed intelligent fault diagnosis systems use the same classifier structure, i.e. a neural network with single hidden layer. This is because all of the works focus on the part of feature extraction and information pre-processing and same intelligent classifier can give clear results for comparison. Several other classification methods also have been briefly introduced. Major techniques in this thesis include backpropagation neural network, principal component analysis, Andrews plot, Autoencoder, and convolutional neural network. These techniques are introduced in Sections 2.2 – 2.5.

Chapter 3 presents a case study of a simulated continuous stirred tank reactor system. The CSTR system is used to generate the simulated process operational data under the normal operation and 11 different typical abnormal conditions. These 11 typical faults include two types, abrupt faults and incipient faults. Training and testing data for developing the fault diagnosis systems are produced from simulation. To test the developed fault diagnosis systems, three groups of abrupt faults with different relative magnitudes and three groups of incipient faults with different developing speeds are also produced. A baseline fault diagnosis system based on neural network is developed as a comparative study to reveal the superiorities of the proposed diagnosis scheme.

Chapter 4 focuses on the method of Andrews plot and how the Andrews plot is applied in process industries. An intelligent fault diagnosis system based on Andrews plot and neural network is proposed. The proposed scheme is applied to the simulated CSTR process and is compared with the performance of the baseline scheme.

Chapter 5 proposes an intelligent fault diagnosis system based on Andrews plot, neural network and qualitative trend analysis. This system is developed to against the uncertainties generated from the parameter selection in Andrews function processing. This proposed scheme is applied to the simulated CSTR process and compared with the baseline scheme.

Chapter 6 proposes an intelligent fault diagnosis system based on Andrews plot, neural network and principal component analysis. Chapter 7 proposes an intelligent fault diagnosis system based on Andrews plot, neural network and autoencoder. These two proposed schemes focus on using different dimensionality reduction methods to the features from Andrews plot instead of manual selection of *t*-values. These proposed schemes are applied to the simulated CSTR process and are compared with the baseline scheme.

Chapter 8 proposes an intelligent fault diagnosis system based on Andrews plot and multiple neural networks with information fusion. This system stacks several imperfect neural network models to overcome the non-robustness in individual neural network models. This proposed scheme is applied to the simulated CSTR process and compared with the baseline scheme.

Chapter 9 proposes an intelligent fault diagnosis system based on Andrews plot and convolutional neural network. Andrews plot method can extract features with a very flexible dimension. However, the proper selection of extracted feature is still a problem. Hence, Andrews plot method combination with convolutional layer could deal with this problem. The proposed scheme is applied to the simulated CSTR process and is compared with the baseline scheme.

Chapter 10 gives the main conclusions of the works in this thesis. Suggestions for further development about the proposed diagnosis systems in future works are also given.

Chapter 2. Literature Review

2.1. Fault Diagnosis Approaches

A process fault diagnosis system is to detect the occurrence of faults and diagnose the fault type, location, and occurrence time information. According to the fault diagnosis results, it is helpful for the process operators to judge the operating state and abnormal conditions of the process. Performance evaluation of fault diagnosis system is predominantly based on the indicators of fault diagnosis speed, accuracy and robustness. Multitudinous previous approaches have been proposed to against the impacts from industrial process fault. These can approximately be divided into three categories: model based approaches, knowledge based approaches, and data analysis based approaches (Patton, Frank, and Clark, 2013; Venkatasubramanian, Rengaswamy, Kavuri, and Yin, 2003).

2.1.1. Model based approaches

Model based approaches (Venkatasubramanian, Rengaswamy, Yin, and Kavuri, 2003; Simani, Fantuzzi, and Patton, 2003; Chen, and Patton, 2012; Patton, 1994; Frank, Ding, and Marcu, 2000; Gertler, 1995; Garcfa, and Frank, 1996; Gao, Cecati, and Ding, 2015; Magni, and Mouyon, 1994) analysis the particular fault in the process by monitoring the changes in estimated model parameters or states. Its core idea is that a fault occuring in a process will change certain physical parameters which can cause changes in parameters and states of estimated model. The model requires the knowledge of the relationships between particular faults and parameters or states in the estimated model. Model based approaches can be subdivided, according to the generation method of residual, into state estimation approach, parameter estimation approach and parity space approach. In general, the parameter estimation approach is more suitable for nonlinear systems, due to the state observer of nonlinear systems is hard to design, and the parity space approach is only suitable for linear systems.

Since the state of the controlled process directly reflects the operating state of the system, state estimation approach can perform fault diagnosis by estimating the state of

the system and combining with an appropriate model. The basic idea is to use the quantitative model of the system and the measurement signal to construct a measurable variable, and use the difference between the estimated value and the measured value as the residual to detect and separate system faults. A fault diagnosis system developing by this method can easily be built and relatively easy to obtain strong robustness.

Parameter estimation approach is to combine theoretical modeling with parameter identification, and to judge the fault condition according to the deviation between the estimated and the normal parameter values. Compared with the state estimation method, the parameter estimation method is more conducive to the separation of faults. However, the traditional parameter estimation method uses historical information to estimate parameters, and there is a significant lag in the real-time parameter changes when forming estimated values for fault diagnosis.

The parity space approach is used as a re-statement to unify residual generation method. Fault diagnosis is achieved by checking whether the actual model of the system is consistent with the nominal model. Its core idea is to check the equivalence of the mathematical model of the system to diagnosis faults using actual measurements of the input and output of the system. This method is equivalent to the observer-based state estimation and diagnosis method.

2.1.2. Knowledge based approaches

Knowledge based approaches (Isermann, 1997; Gao, Cecati, and Ding, 2015) diagnose process faults via the available knowledge of the process behaviour and the experience of the process operations. A fault diagnosis system implemented by knowledge based approaches commonly utilizes rule based approaches and the qualitative simulation based approaches. The diagnostic rules in rule based approaches can be devised methodically from process structure and unit functions. The development of a knowledge based diagnosis system requires a great deal of effective knowledge about process structure, process unit functions, and qualitative models of process units under various faulty conditions. Exact mathematical model in knowledge based approaches is unnecessary.

The rule based approaches (Zhang, and Roberts, 1991; Dash, Rengaswamy, and Venkatasubramanian, 2003; Isermann, 1998) diagnose the fault by causally tracing symptoms backward along their propagation paths. Fuzzy logic is frequently used in rule-based approach to deal with the information with uncertainties. The introduction of fuzzy logic into process fault diagnosis is mainly to overcome the difficulties caused by the uncertainty, inaccuracy and noise of the process itself.

The qualitative simulation based approaches (Venkatasubramanian, Rengaswamy, and Kavuri, 2003; Maurya, Rengaswamy, and Venkatasubramanian, 2007; Maurya, Paritosh, Rengaswamy, and Venkatasubramanian, 2010; Zhang, Roberts, and Ellis, 1990) diagnose the particular faults by comparing predicted behaviour with the actual observations. The predicted behaviour is estimated under the normal operating condition and various faulty conditions.

For a chemical engineering process, the mechanistic models and first principles are dependable under well-controlled operations. Fault diagnosis system developed by model based approaches and knowledge based approaches are suitable for the processes with adequate experiences or the processes with underlying theories from physics or mathematics. But in the chemical industries, both approaches are hard to acquire sufficient knowledge and first principles to implement a fault diagnosis system with favourable performance, due to the complicated processes with uncertainties and demanding requirement on monitoring capabilities.

2.1.3. Data analysis based approaches

The data analysis based approaches (Qin, 2003; Qin, 2012; Ali, Hussain, Tade, and Zhang, 2015; Ge, Song, and Gao, 2013; Famili, Shen, Weber, and Simoudis, 1997; Chiang, Russell, and Braatz, 2000; Russell, Chiang, and Braatz, 2012) are more suitable for the intricate large-scale industrial processes comparing to other approaches, according to its nature, and advances in technology of process information acquirement. As the process information utilized in data analysis based approaches is extracted from the industrial processes covering various normal and abnormal

operations, these information includes the interpretable process knowledge, and uncertainties which are time-varying and ubiquitous in the process operations.

A typical procedure for industrial process monitoring and analysis in practice is as follows: firstly, clean and pre-process the measured process information. Then develop the data analytic model by using the data collected under the normal operations and well-controlled circumstances. The indicators of fault detection and control limits need also to be established. Finally, perform fault diagnosis and troubleshooting. Note that faulty data are non-essential, but methods, such as linear discriminant analysis (LDA), will be applicable if plenty of faulty data are provided.

In the prior works, several techniques of data monitoring and analysis achieved varying degrees of success in the industrial process fault diagnosis. Among a variety of data analysis approaches, two notable methods are adopted in this thesis to deal with the high-dimensional data of process information: neural network based fault diagnosis method and multivariate statistical analysis based fault diagnosis method.

2.1.3.1 Multivariate statistical analysis

Multivariate statistical analysis based fault diagnosis methods (Kourti, and MacGregor, 1996; Chiang, Russell, and Braatz, 2000; Ge, and Song, 2007; Wang, Si, Huang, and Lou, 2018; Shah, Patwardhan, and Huang, 2002; Miletic, Quinn, Dudzic, Vaculik, and Champagne, 2004; Zhang, Martin, and Morris, 1996; Martin, Morris, and Zhang, 1996; Raich, and Çinar, 1997; Jiang, Yan, and Huang, 2019) have the core idea that the high-dimensional process data are projected via statistical algorithms onto a new set of latent variables to extract the feature and re-constructing the feature dataset, while maintain the main information. The widely accepted multivariate statistical analysis techniques including principal component analysis (PCA), partial least squares (PLS), and independent component analysis (ICA), etc. Wherein the principal component analysis is frequently used in this thesis, and its fundamental idea is that the high-dimensional process data transform all the available variables of inputs and outputs into a dataset of uncorrelated principal components, which are generally with a smaller size, i.e. dimensionality reduction.

In most works from chemical engineering, the purpose of hybrid techniques is improving the monitoring performance and enhancing the diagnostic applicability by pre-processing the measured industrial process data, especially dimensionality reduction. Dimensionality reduction plays a crucial role in the fault diagnosis system. Its advantages include saving the storage space, reducing the cost of computation, enhancing the availability of some algorithms which have poor performance in high-dimensional data, removing redundant features, and helping with data visualization.

This thesis implemented several hybrid fault diagnosis systems by integrating the techniques including PCA and Andrews plot. Andrews plot is an unusual but effective method in fault diagnosis system development, which is a key component in this thesis. The introduction of PCA and Andrews plot will be given in later independent sections. In addition to the methods mentioned above, past studies proposed various methods applied in the development of fault diagnosis systems. Some of the methods are briefly described in the following.

Partial least squares (PLS) methods (Li, Qin, and Zhou, 2010; Zhao, Zhang, and Xu, 2006; Zhao, Zhang, Xu, and Xiong, 2006; Zhang, 2001; Li, Ye, Wang, and Zhang, 2005) achieve the dimensionality reduction through decomposing the process data according to quality data and extracts the latent variables that are most relevant to the product quality. This method can enhance the diagnosis capability in quality relevant faults and reduce the false alarm.

Independent component analysis (ICA) (Kano, Tanaka, Hasebe, Hashimoto, and Ohno, 2003; Lee, Qin, and Lee, 2006) is a statistical technique used to find out the latent features that exist under random variables. ICA defines a generative model for observed data to extract the independent components of observed data. It is supposed that the variables of model are generated by latent features linearly mixed by an unknown system and assumed that latent features are non-Gaussian distribution and independent of each other. In contrast to PCA, it has a better performance in discover the latent features and can suppress Gaussian noise.

Forward feature selection and backward feature elimination (Ahmad, and Zhang, 2009; Mao, 2004; Sutter, and Kalivas, 1993) have similar key idea, and both of them are suitable for the dataset with fewer input variables, due to the high cost of computation and time consuming. Forward feature selection firstly uses each feature to train the model for *n* times, and *n* models are obtained. The variables in the model which has best performance are selected as initial variables. Continue training by adding one variable at a time and retaining the variables in the best performance model. Repeating previous step with adding and filtering, until the model performance cannot be significantly improved. Backward feature elimination firstly trains a model with all variables and calculates the model performance. Then compare the performance with the models trained by removing each variable one at a time. Identify the variable that has the least impact on model performance and remove it. Repeat the process until can no longer remove any variables.

Kernel principal component analysis (KPCA) (Huang, and Yan, 2017; Cho, Lee, Choi, Lee, and Lee, 2005; Ge, Yang, and Song, 2009) is a technique that uses the kernel function to reformulate the PCA in a high-dimensional space. The kernel method originated in the support vector machine and has been generalized into various algorithms. Difference with PCA, kernel PCA calculates the principal eigenvectors of the kernel matrix instead the covariance matrix so that it can deal with nonlinear problems.

Isomap (Balasubramanian, Schwartz, Tenenbaum, de Silva, and Langford, 2002; De Silva, and Tenenbaum, 2003) uses the concept of geodesics in differential geometry, where it is expected the data retained geodesic distances on the manifold after being mapped to a lower dimensional space. Geodesic distance is the distance between two points measured over the manifold. The algorithm calculates the geodesic distance between any two samples and constructs the distance matrix. Finally, the dimensionality reduction is achieved by classical scaling the data points in the low-dimensional space. After dimensionality reduction, the data of low-dimensional representation retains the distance information between the original data points.

Local linear embedding (LLE) (Sakthivel, Nair, Elangovan, Sugumaran, and Saravanmurugan, 2014; Roweis, and Saul, 2000) constructs a graph representation of the data points and attempts to preserve the local properties of the data. The local properties of high-dimensional data point are written as a linear combination of their nearest neighbours to the configuration. In the low-dimensional representation, LLE retains the reconstruction weights of the linear combinations.

2.1.3.2 Statistical learning classifiers

Neural network is an important part of artificial intelligence (AI). AI is defined that the capability of computers or other machines in performing activities that require human intelligence. Since the Dartmouth Conference in 1956 (Negnevitsky, 2005) to the deep learning in present, thousands study relevant to AI provide several breakthroughs. In the big data era of today, the application of AI is extended to various fields, including chemical industrial process monitoring. Since the introduction of AI techniques to the field of chemical engineering, several traditional techniques with their relative advantages, such as Luenberger observer, Extended Kalman Filter (EKF) and sliding mode observer (Aguilar-López, 2003; Damour, Benne, Boillereaux, Grondin-Perez, and Chabriat, 2010; Mesbah, Huesman, Kramer, and Van den Hof, 2011; Velardi, Hammouri, and Barresi, 2009; Zarei, and Poshtan, 2010), have been replaced by AI or fusion with AI, such as fuzzy Kalman filter (FKF) and differential neural network observer (DNNO) (Prakash, and Senthil, 2008; Senthil, Janarthanan, and Prakash, 2006; Chairez, Poznyak, and Poznyak, 2007; Porru, Aragonese, Baratti, and Servida, 2000). An ANN is composed of a number of neurons with various parameters and tries to mimic the function of human brain. Figure 2.1, taken from (Maltarollo, Honório, and da Silva, 2013), shows the neuron concept of artificial neural network and human. This thesis developed fault diagnosis systems by using the ANN as the classifier to output the fault diagnosis results, Hence the introduction of ANN will be given in a later section. Other techniques, e.g. auto-encoder and convolutional neural network, applied in this thesis will also be given in later sections.

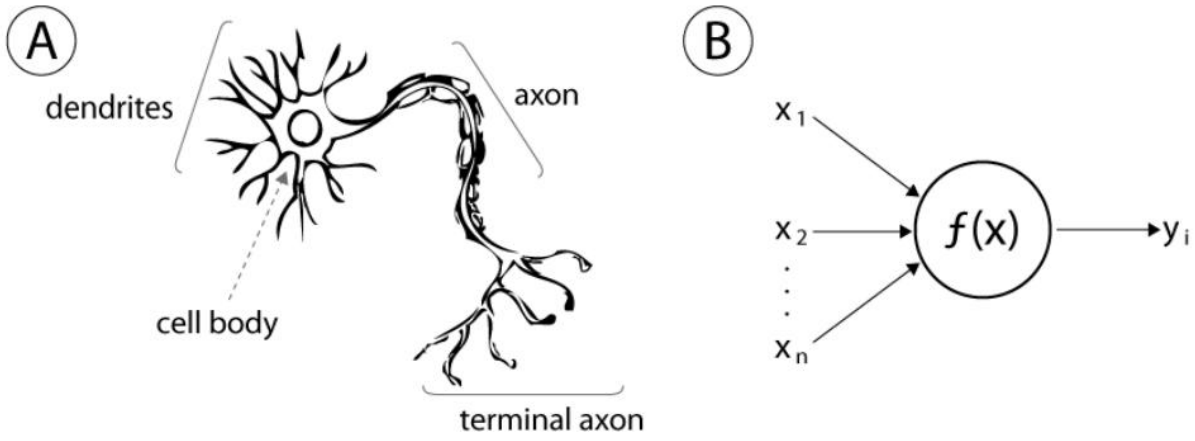

Figure 2.1 Neurons concept of artificial neural network and human (Maltarollo, Honório, and da Silva, 2013)

With the advances of statistical machine learning, the boundaries between the industrially adopted data driven applications, such as multivariate latent variable methods, and statistical machine learning are largely blunted. In a pragmatic view, all of them can be roughly classified into the class of statistical learning methods (Hastie, Tibshirani, and Friedman, 2009). The chemical engineering also provided imperative contributions to machine learning (Qin, 2014; Venkatasubramanian, 2019). For examples, the auto-associative neural networks (Kramer, 1991) developed for perform nonlinear PCA in process fault diagnosis is similar to the building blocks of deep neural networks for dimension reduction and feature extraction (Bengio, 2009). The integration of neural networks with statistical method to produce a strip net (Bhat, and McAvoy, 1992) has been contributed to remove zero weights after a regularization penalty. This work is similar to the sparse learning methods, such as Lasso (Zou and Hastie, 2005).

In addition to the ANN, prior works proposed other popular statistical learning classifiers specifically including k-nearest neighbours (k-NN) algorithms, naïve Bayes classifier, and support vector machine (SVM). Deep learning methods has also been increasingly applied in the field of fault diagnosis in recent years. Some of the techniques are briefly described in the following.

The k-NN algorithm (Wang, 2016, Kotsiantis, Zaharakis, and Pintelas, 2007) is based on the principle that instances within a dataset generally close to other instances with similar properties. It consists of the number of measured instances k, the distance metric and the decision rule for classification. The algorithm has obvious advantages including high accuracy, insensitive to outliers, and no data input assumptions. However, it also has disadvantages including high computational complexity and high space complexity. Assuming that a training set of classified instances $T = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, $y_i = c_1, c_2, ..., c_k; i = 1,2, ..., N$, where x_i and y_i are the feature vector of unlabelled instance and label, respectively. For each training sample in T, the k-NN searches for the k nearest instances to x based on a given distance metric. The decision rule for classification is as follow:

$$y = \arg max_{c_j} \sum_{x_i \in N_k(x)} I(y_i = c_j)$$

$$i = 1,2, ..., N; \quad j = 1,2, ..., K; \tag{2.1}$$

where I is the indicator function and $N_k(x)$ is the instances neighbourhood.

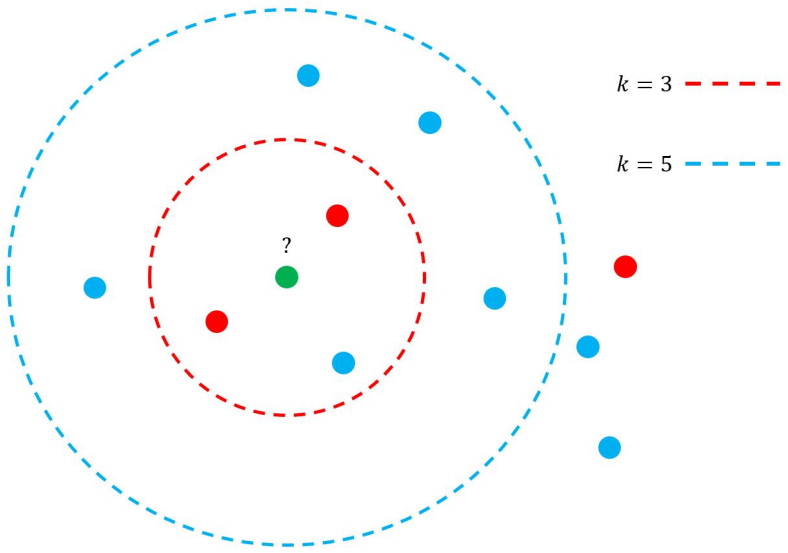

Figure 2.2 Example of k-NN

Figure 2.2 gives an example of k-NN. The green dot represents the test sample. If $k = 3$, the test sample will be classified into the class of red, due to the proportion of the red dot is 2/3. If $k = 5$, the test sample will be classified into the class of blue, due to the proportion of the blue dot is 5/7.

Naive Bayes classifier (Baraldi, Podofillini, Mkrtchyan, Zio, and Dang, 2015; Muralidharan, and Sugumaran, 2012) is a classification method based on Bayes' theorem and the strong independence assumptions between the features. The entire formalization processes only make the assumptions primitively and simply. Naïve Bayes is a part of Bayes decision theory, where the key idea is high probability thinking. The naïve Bayes classifier has a stable classification efficiency and can deal with multi-classification tasks. However, naïve Bayes classifier theoretically has the lowest error rate than other classification methods, but the fact is not the case, due to the data in practice are not always mutual independence between features. Moreover, the classifier is necessary to know the prior probability, and the prior probability generally based on the hypothesis. Hence, the hypothetical models can affect the prediction performance. Furthermore, the classification decision has a certain error rate, due to the classification determined by posterior probability, and the posterior probability determined by the prior and data. The basic Bayes equation is as follows.

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)} \tag{2.2}$$

It can be seen as:

$$P(Category|Feature) = \frac{P(Feature,Category)}{P(Feature)} = \frac{P(Category)P(Feature|Category)}{P(Feature)} \tag{2.3}$$

According to the naïve Bayes assumption that features are mutual independence, hence, the following equation can be obtained as while the category is y:

$$P(X|Y = y) = \prod_{i=1}^{d} P(x_i|Y = y) \tag{2.4}$$

Combining the above equations, the following can be obtained:

$$P_{post} = P(Y|X) = \frac{P(Y)\prod_{i=1}^{d} P(x_i|Y=y)}{P(X)} \tag{2.5}$$

Since $P(X)$ is a constant value, the function can be written as:

$$P(y_i|x_1, x_2, ..., x_d) = \frac{P(y_i)\prod_{j=1}^{d} P(x_i|y_i)}{\prod_{j=1}^{d} P(x_i)} \quad (2.6)$$

Support vector machine (SVM) (Cristianini, and Shawe-Taylor, 2000; Scholkopf, and Smola, 2018; Leong, Bahadori, Zhang, and Ahmad, 2019; Maldonado, Weber, and Famili, 2014) is a type of supervised learning algorithm for small sample classification. It establishes optimal separating hyperplane $f(x) = 0$ between data by dealing with the constrained quadratic optimization problem based on structural risk minimization (SRM), which is an inductive principle of use in machine learning.

$$y = f(x) = w^T x + b = \sum_{i=1}^{N} w_i x_i + b \quad (2.7)$$

where w is an *N*-dimensional vector and b is a scalar quantity.

The optimal separating hyperplane built by find out the maximum margin. The support vectors are obtained via convert the optimization problem with Kuhn-Tucker condition into LaGrange duality quadratic optimization. Figure 2.3 shows a basic classifier of SVM.

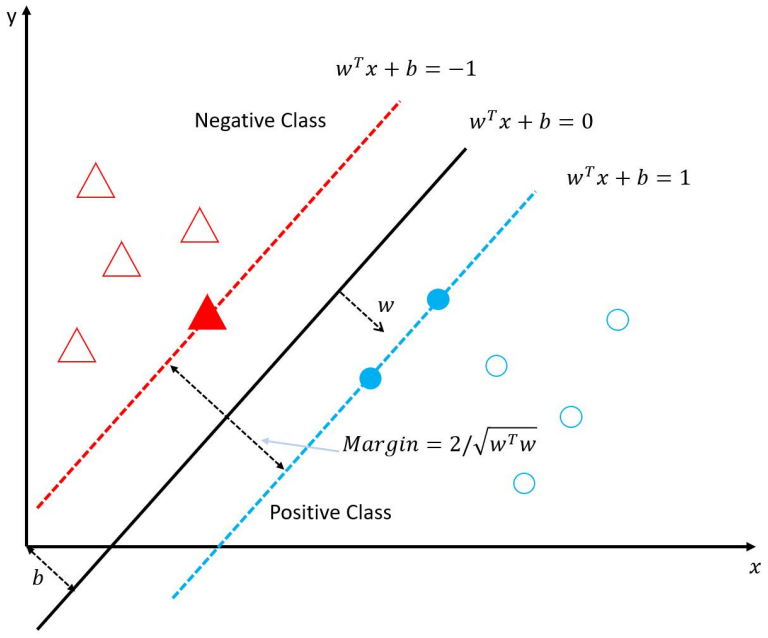

Figure 2.3 A basic classifier of SVM

For dealing with the linear classification, linear classifier based on SVM is a very effective method. For dealing with the non-linear classification, non-linear classifier based on SVM can be built via kernel trick. Kernel trick is a mapping function to map the low-dimensional data into high-dimensional representations. Thus, the SVM can process the high-dimensional separable data to deal with the inseparable problem in low-dimensional data. The kernel trick is also applied in other algorithms, such as KPCA. In general, SVM can achieve favourable results on the small sample set. Its generalization ability is excellent, due to its optimization objective is to minimize structural risk. Since the SVM acquires the structured description of data distribution, through the concept of margin, thus the requirements of data distribution and scale are reduced. However, in the SVM method it is hard to determine the kernel function in the high-dimensional space, which means SVM turned the difficulty on complexity of high-dimensional space into the difficulty of acquiring the kernel function. Moreover, SVM also demands large storage space, after the kernel function is determined. Hybrid fault diagnosis system combination with techniques of neural network and multivariate statistical analysis is commonly used. In addition to the previously mentioned methods, other techniques, such as rough sets (Pawlak, 1982) and slow feature analysis (Wiskott and Sejnowski, 2002), are also integrated in several hybrid fault diagnosis systems.

2.2. Introduction of artificial neural network

Artificial neural network (ANN) (Bishop, 1995; Sarle, 1994; Venkatasubramanian and Chan, 1989; Watanabe, Matsuura, Abe, Kubota, and Himmelblau, 1989; Ripley, 2007; Himmelblau, 2008; Horn, 2001; Du, del Villar, and Thibault, 1997; De Assis and Maciel Filho, 2000; Katare, Bhan, Caruthers, Delgass, and Venkatasubramanian, 2004; Stephanopoulos and Han, 1996; Kadlec, Gabrys, and Strandt, 2009; Kalogirou, 2003; Sharma, Singh, Singhal, and Ghosh, 2004; Lashkarbolooki, Vaferi, and Mowla, 2012; Priddy, and Keller, 2005; Zhang, and Friedrich, 2003; Zhang and Morris, 1998; Zhang, 2001; Zhang, 1999; Zhang, 1999; Zhang, Martin, Morris, and Kiparissides, 1997; Chen and Chang, 1996) is one of the most popularly used technique in

miscellaneous AI relevant fields including the process industries. The superiority of neural network based method is that the multiple independent and dependent variables can be handled synchronously by the network, without the requirement of first principle knowledge on the process.

2.2.1. Neural network structure

The most widely used ANN generally contains three layers: input layer, hidden layer and output layer. Figure 2.4 shows a basic structure of ANN with a single hidden layer. Each layer contains several disconnected neurons (also refer as perceptron). Each neuron in a layer, except for the neurons in input layer, is connected to all the neurons of previous layer, i.e. full connected, and the outputs of neuron in a layer are the inputs of the next layer, except for the output layer. Every connection between neurons has a weight. The output of a neuron is calculated by an activation function with the connection weights and values of inputs. The decisive work of developing a neural network is to design the hidden layer and train the weights between neurons. There can be multiple hidden layers. The network with a small number of hidden layers refers as shallow neural network, and with many hidden layers refer as deep neural network. The neural network works well in data classification, particularly applied in industry.

Deep learning (LeCun, Bengio, and Hinton, 2015; Venkatasubramanian, 2019; Chiang, Lu, and Castillo, 2017; Qin and Chiang, 2019; Abiodun, Jantan, Omolara, Dada, Mohamed, and Arshad, 2018; Ge, Song, Ding, and Huang, 2017) is a major breakthrough in AI and has been a hot subject unsurprisingly with large numbers of discussion and research in the field of process industries. Since the deep learning method has been applied to soft sensor modelling (Shang, Yang, Huang, and Lyu, 2014), more and more deep neural networks have been used in development of industrial process fault diagnosis system. This thesis also developed fault diagnosis system with the deep learning techniques of convolutional neural network and stacked autoencoder. The convolutional neural network and stacked autoencoder will be introduced in later chapters.

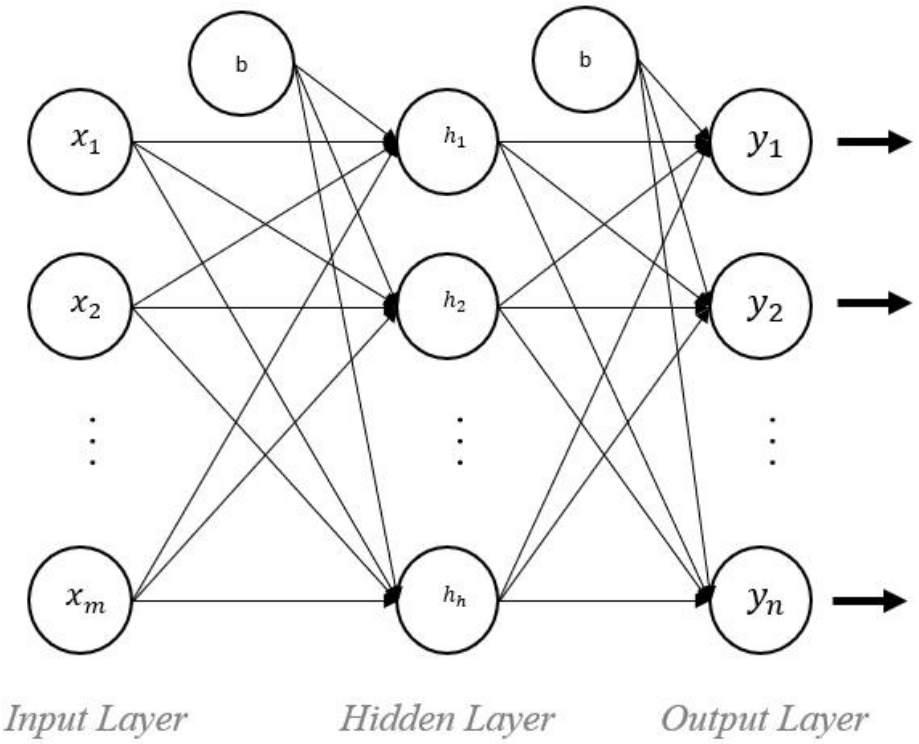

Figure 2.4 A basic structure of ANN

2.2.2. Neural networks for classification and regression

Neural network is applicable to data that are non-linearly separable. It can utilize the logical conjunction of multiple linear classifiers to implement the non-linear segmentation. Figure 2.5 shows an example of non-linear segmentation in neural network. The input layer generates two linear classifiers, and then the non-linear segmentation realized via combining the weights of two linear classifiers by logical conjunction.

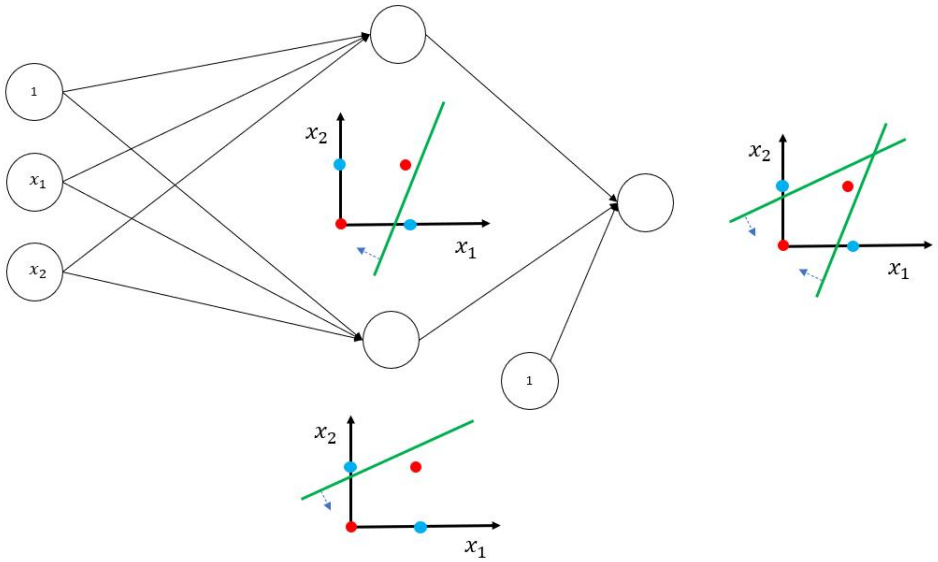

Figure 2.5 Example of non-linear segmentation in neural network

Theoretically, a single hidden layer neural network can approximate any continuous function, as long as the number of neurons in the hidden layer is sufficient. Multi-hidden-layer and single-hidden-layer neural networks mathematically have the same expression capabilities, but multiple-hidden-layer neural network can obtain a better engineering performance than single-hidden-layer neural networks in certain cases. However, it depends on the specific case, more layers or neurons in layers are not necessarily better.

2.2.3. Overfitting in neural network

Increasing the number of hidden layers or hidden layers, the expression ability of neural network will become stronger, but excessive hidden layers and neurons may cause overfitting. For against the problem of overfitting (Caruana, Lawrence, and Giles, 2001; Hagiwara and Kuno, 2000; Mc Loone and Irwin, 2001; Baldi and Sadowski, 2013; Hinton, Srivastava, Krizhevsky, Sutskever, and Salakhutdinov, 2012), it is suggested to utilize regularization instead reduction of neural network parameters. In the deep neural network, the method of dropout is generally used to against the overfitting. Regularization is an operation that imposes a priori restriction or constraint on network weights to achieve a certain purpose. In neural network, the purpose is generally to prevent overfitting.

2.2.4. Activation function

In a neural network, activation function is used in calculation after the neuron inputs are linearly transformed by $wx + b$. The activation function is mainly used to provide the non-linear modelling capability of the neural network. Neural network can be used to build non-linear models, due to neural network can approximate arbitrarily non-linear function via the activation function introduces non-linear factors to the neuron. It is most believed that an activation function should be differentiable almost everywhere (Gulcehre, Moczulski, Denil, and Bengio, 2016). There are some commonly used activation functions as following.

The sigmoid activation function is one of the most frequently used activation functions in neural networks. Its output is mapped in the interval from 0 to 1 and monotonous continuous. It works well when the feature difference is more complicated, or the difference is not particularly large. The limitation of its nature is that the function saturation makes the gradient disappearing, when the value is close to 0 or 1. In addition, most neurons will also be saturated, when the initialization weight is too large. The sigmoid activation function is given in the following.

$$f(x)_{sigm} = \frac{1}{1+e^{-x}} \tag{2.8}$$

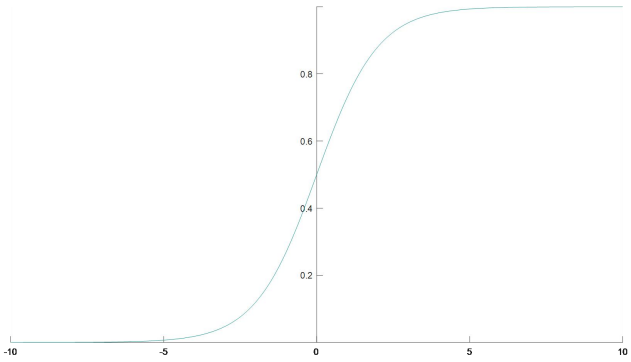

Figure 2.6 The sigmoid activation function

Hyperbolic tangent activation function (tanh) can be seen as an enlarged sigmoid function with a faster convergence speed and its output is mapped in the range of [-1, 1]. The function is given as the following.

$$f(x)_{tanh} = \tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}} = \frac{e^{2x}-1}{e^{2x}+1} = 2sigmoid(2x) - 1 \qquad (2.9)$$

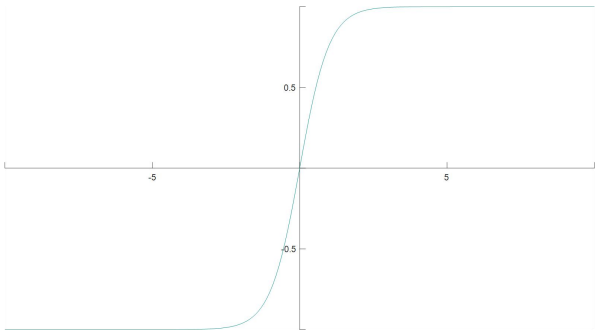

Figure 2.7 Activation function of hyperbolic tangent function (tanh)

The rectified linear unit (ReLU) (Krizhevsky, Sutskever, and Hinton, 2012) is the most popularly used activation function in recent years. Its superiorities enables it gradually replacing the sigmoid function as the most commonly used function, particularly in deep neural networks. The implementation of ReLU function is simpler and effectively alleviate the problem of gradient disappearance. It can also provide the sparse expression ability of neural networks. It is commonly believed that the sparsity is an imperative reason for its good performance (Xu, Wang, Chen, and Li, 2015). However, as the training progresses, some weights cannot be updated, and this 'neurons death' is irreversibly during training. The activation function of ReLU has various variants which have their own optimization. The variants including Leaky-ReLU (Maas, Hannun, and Ng, 2013), parametric rectified linear unit (PReLU) (He, Zhang, Ren, and Sun, 2015), concatenated Rectified Linear Units (CReLU) (Shang, Sohn, Almeida, and Lee, 2016), and exponential linear unit (ELU) (Clevert, Unterthiner, and Hochreiter, 2015), etc.

$$f(x)_{ReLU} = \max(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases} \tag{2.10}$$

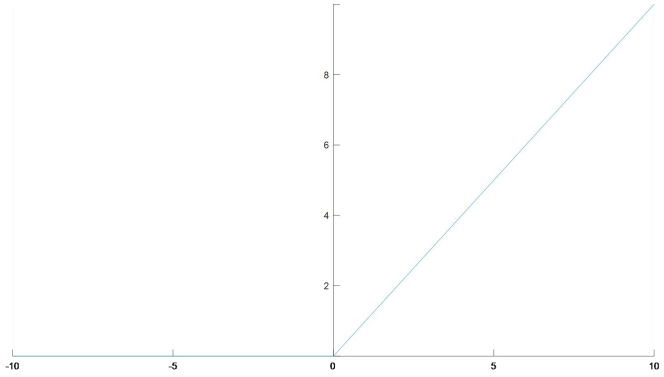

Figure 2.8 Activation function of rectified linear unit (ReLU)

2.2.5. Backpropagation algorithm

The neural network is generally trained by the best-known algorithm of backpropagation. The backpropagation deals with the instances in the training set via iteration. It calculates the difference between the predicted value and the true target value of the training inputs. The algorithm updates the weight of each connection through minimizing error computed by reverse transmission (output layer to hidden layer to input layer). The expression of backpropagation as following.

The activated value in each neuron can be expressed as:

$$a_j^l = f(\sum_k w_{jk}^l a_k^{l-1} + b_j^l) \tag{2.11}$$

where the a_j^l is activation output in the j neuron of l layer, f is the activation function, b_j^l is bias value in the j neuron of l layer, the w_{jk}^l is the weight in the connection between the kth neuron of $(l-1)$th layer to the jth neuron of the lth layer.

The propagation firstly calculates the partial derivative

$$e_j^l = \frac{\partial C}{\partial z_j^l} \tag{2.12}$$

where e_j^l is the partial derivative of the network error with respect to the j neuron of l layer, z_j^l is weighted input of the j neuron of l layer, which is defined that the $z^l = w^l a^{l-1} + b^l$, where the z^l is weighted input, i.e. $a^l = f(z^l)$. C is cost function which is usually the quadratic cost as the form:

$$C(w,b) = \frac{1}{2n}\sum_x \left\| y(x) - a^L(x) \right\|^2 \tag{2.13}$$

where n is the number of training samples, $y(x)$ is the corresponding desired output, L represent the maximum number of layers of the neural network, noted that in this case, $a^L = a^{OL}$, where OL represents the output layer, due to the output of a neural network is the output layer. Hence, the error gradient with respect to the output layer can be expressed as:

$$e_j^{OL} = \frac{\partial C}{\partial a_j^{OL}} f'(z_j^{OL}) \tag{2.14}$$

wherein $\frac{\partial C}{\partial a_j^{OL}}$ measures the change degree of loss function in j activation output, and $f'(z_j^{OL})$ measures the change degree of loss activation function at weighted input.

The error in layers except for output layer depend on the error of its next layer. Through the following equation, the error in the lth layer is calculated by the error in the $(l+1)$th layer.

$$e_j^l = \sum_k e_k^{l+1} w_{kj}^{l+1} f'(z_j^l) \tag{2.15}$$

The partial derivatives of the cost function with respect to weights and bias are utilized to update the parameters and can be expressed as follow.

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} e_j^l \tag{2.16}$$

$$\frac{\partial C}{\partial b_j^l} = e_j^l \tag{2.17}$$

2.3. Introduction of principal component analysis

Principal component analysis (Jiang, Yan, and Huang, 2015; Wise, Ricker, Veltkamp, and Kowalski, 1990; Kresta, MacGregor, and Marlin, 1991; Ringnér, 2008) is a technique of multivariate statistical data analysis and is popularly applied to process fault diagnosis. When applied to the online measured multi-variable data, it uses the correlation between variables under normal operating conditions to achieve an appropriate dimensionality reduction, while keeping the main information in the data. For each *n*-dimensional dataset, the PCA model can map the correlated variables to *k*-dimensional uncorrelated principal components, where *k* is generally much less than *n*. The measured process dataset can be transformed into the new coordinate system from the original coordinate system using PCA. The first new coordinate axis is selected as the direction of maximum variance in the original data and the second coordinate axis is selected as the direction that is orthogonal to the first axis and has the maximum variance. Followed by analogy, the required *k* coordinate axes can be obtained.

In practice, the algorithm of singular value decomposition (SVD) (Shlens, 2014; Abdi, 2007; Alter, Brown, and Botstein, 2000) is generally used to obtain the principal components. SVD is a widely used algorithm in the field of machine learning, especially used for feature decomposition in dimensionality reduction algorithms. Before using the algorithm of SVD, the process data should be scaled to zero mean and unit variance. For each scaled normal measured information dataset x_n, its SVD is as follow:

$$[U, \Sigma, V] = SVD(x_n) \tag{2.18}$$

where x_n is the nominal 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 value of all elements except the main diagonal are 0, and each element on the main diagonal is called a singular value, which is nonnegative and arranged in descending order. These matrixes satisfy the relationship as:

$$x_n = U \times \Sigma \times V^{'} \tag{2.19}$$

The dimensionality reduction on online monitoring dataset can be achieved as:

$$X^{'}_{d\times n} = U^{T}_{d\times m} X_{m\times n} \tag{2.20}$$

or

$$X^{'}_{m\times k} = X_{m\times n} V_{n\times k} \tag{2.21}$$

For fault diagnosis, an appropriate number of principal components should kept with minimal loss of information. Hence, a threshold is generally set for principal components selection. The follow threshold equation can help to make a decision on the numbers of principal components kept.

$$\frac{\sum_{i=1}^{k} \Sigma_{ii}}{\sum_{i=1}^{n} \Sigma_{ii}} \geq 1 - \text{threshold} \tag{2.22}$$

2.4. Introduction of Autoencoder

Autoencoder (Hinton and Salakhutdinov, 2006; Bengio, Courville and Vincent, 2013; Hinton and Zemel, 1994; Bourlard and Kamp, 1988) is an unsupervised learning algorithm, widely accepted for data dimensionality reduction or feature extraction. Its compression and decompression algorithms are implemented via neural networks.

2.4.1. Autoencoder structure

An autoencoder is a neural network with 3 layers: input layer, hidden layer and output layer. The neural network is trained by the backpropagation algorithm to make the output value equal to the input value: the input is compressed into a latent space representation, and then reconstructs this representation into the output. In other words, it learns an "identity function" to minimize the error $L(input, output)$. Wherein that the trained autoencoder can obtain a low-dimensional vector representing the input dataset in the hidden layer. It can achieve the dimensionality reduction on inputs. The process of input compression is called encoder, and the process of input reconstruction is called decoder. Figure 2.9 shows a basic structure of Autoencoder.

For each original input data, $= [x_1, x_2, ..., x_{var}]^T$, the encoder projects x from the input layer into the hidden layer, $h = [h_1, h_2, ..., h_k]^T$, through the mapping

function $f(x)$. And then the decoder maps h represented by the hidden layer to the output layer by the mapping function $\tilde{f}(h)$. Wherein the k is generally more less than m. The $f(x)$ function and $\tilde{f}(h)$ function can be respectively expressed as:

$$h = f(x) = s_f(W_x x + b) \tag{2.23}$$

$$\tilde{x} = \tilde{f}(h) = s_{\tilde{f}}(\tilde{W}_h h + \tilde{b}) \tag{2.24}$$

where W_x is a weight matrix of $k \times var$, $\tilde{W}_x$ is a weight matrix of $var \times k$, b and $\tilde{b}$ are bias vectors. The activation function s_f and $s_{\tilde{f}}$ can be the sigmoid function, the tanh function or the rectified linear unit function (ReLU). Hence the parameter set is $\theta = \{W_x, \tilde{W}_h, b, \tilde{b}\}$. In this study, the sigmoid function is used with the loss function as the mean square error (MSE):

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} L_{MSE}(x_i, \tilde{x}_i) = \frac{1}{2N}\sum_{i=1}^{N} \|\tilde{x}_i - x_i\|^2 \tag{2.25}$$

where N is the number of training samples. Each training sample x_i is projected to the implicit representation h_i, and then mapped to the reconstructed data $\tilde{x}_i$.

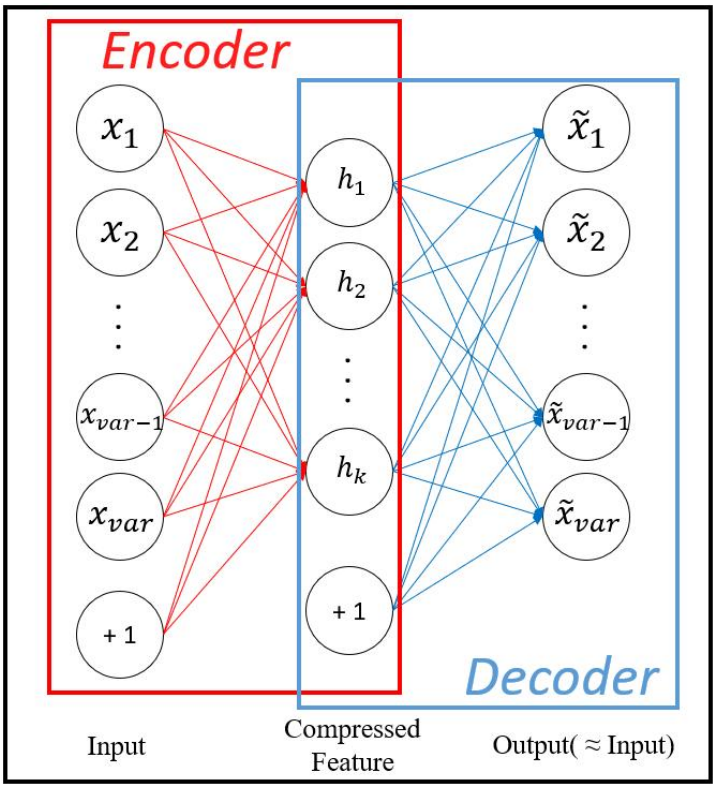

Figure 2.9 A basic structure of an autoencoder

2.4.2. Stacked autoencoder

Stacked autoencoder (Germain, Gregor, Murray, and Larochelle, 2015) is a hierarchical deep neural network structure that can be straightforward developed a deep autoencoder by inserting multiple hidden layers from autoencoder. The autoencoder described above is merely constituted by one encoder and one decoder. It can stack multiple hidden layers of encoders and decoders, thereby obtains a feature with more benefits comparison with shallow autoencoder. Paul and Venkatasubramanian (2014) gives a research on perspective of group theory for the mechanism of neural network with deep architecture. Although the deep learning requires more substantial studies to reveal the clearly theoretical and conceptual, the previous experiments give encourage performance on process data analysis.

Figures 2.10 and 2.11 show how a stacked autoencoder stacking two autoencoders connected layer by layer. It uses the original input, x_{var}, to train the first autoencoder, and learns a compressed latent representation $h^{(1)}$ with k-dimension which is much less than var. This step is shown as the left of Figure 2.10. Then the representation $h^{(1)}$ is used as an input to train the second autoencoder. This step is shown as the right of Figure 2.10. Finally, they can be combined as shown in Figure 2.11, and the eventual compressed feature $h^{(2)}$ can be fed into a fault classifier as inputs, which has the dimensionality of d, more less than k.

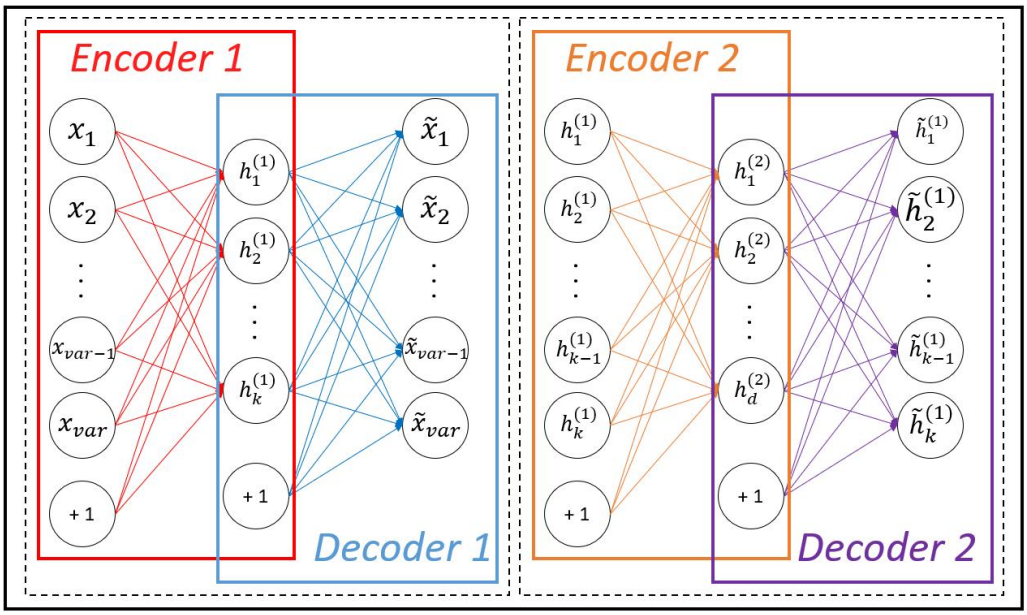

Figure 2.10 The encoders 1 and 2 training process

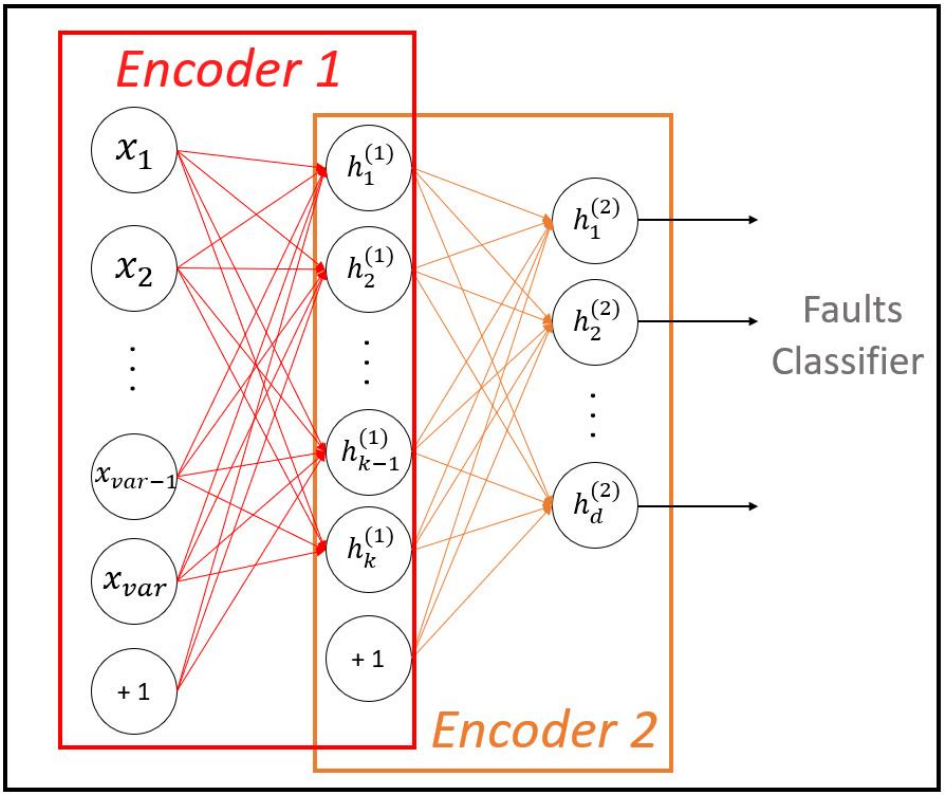

Figure 2.11 A basic structure of a stacked autoencoder

2.5. Deep learning applied in fault diagnosis system

Convolutional neural network (CNN) (Bouvrie, 2006; Wen, Li, Gao, and Zhang, 2017; Wu and Zhao, 2018) is a deep neural network with convolutional structure. In recent years, CNN has become one of the most popular techniques of artificial intelligence. Its development has gone through decades, with several great advances from LeNet-5 to AlexNet, VGG, ResNet etc. (LeCun, Bottou, Bengio, and Haffner, 1998; Krizhevsky, Sutskever, and Hinton, 2012; Simonyan and Zisserman, 2014; He, Zhang, Ren, and Sun, 2016). CNN includes the non-linear trainable convolutional layers, sub-sampling layers (referred as pooling layer), and a fully connected layer. Figure 2.12 shows a basic structure of convolutional neural network. Figure 2.13 is a classic CNN architecture of LeNet-5 and is taken from (LeCun, Bottou, Bengio, and Haffner, 1998).

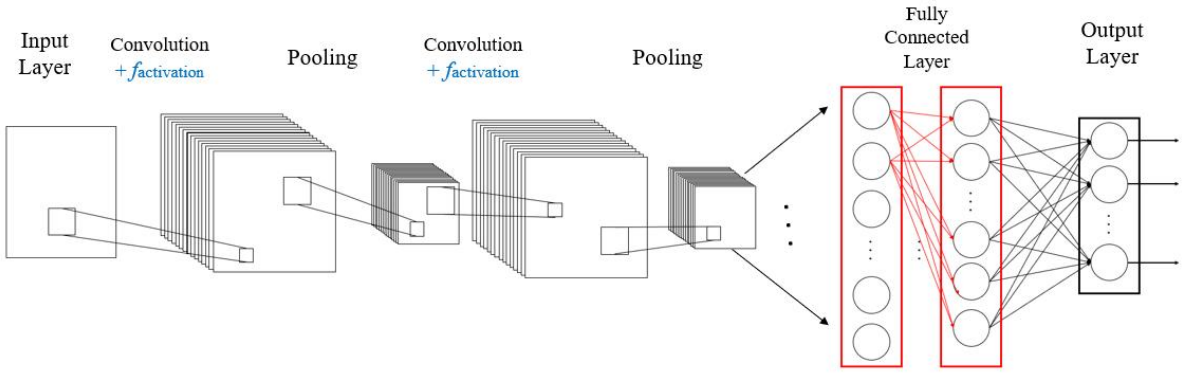

Figure 2.12 A basic structure of convolutional neural network

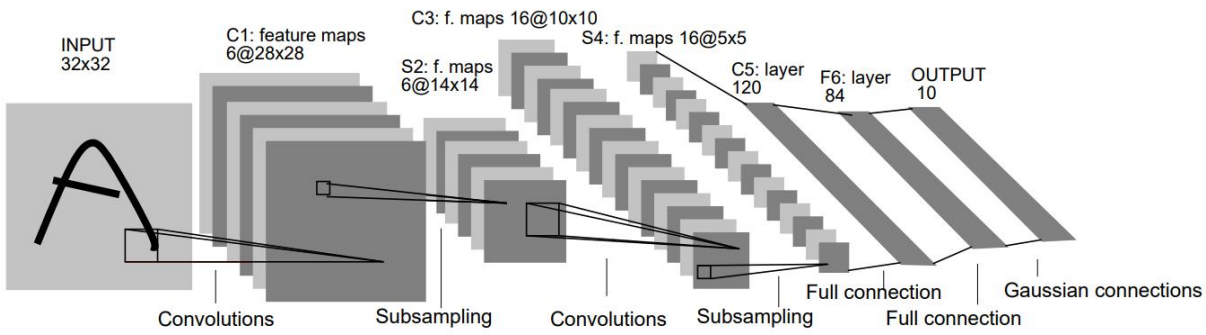

Figure 2.13 Architecture of LeNet-5 (LeCun, Bottou, Bengio, and Haffner, 1998)

2.5.1. Convolutional operation

In a convolutional layer, the output represents 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. The commonly used activation functions in CNN are mentioned previously. Assume that the size of the original input data is $W_0 \times H_0 \times D_0$, 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:

$$x_j^l = f\left(\sum_{i=1,\dots,D_0} x_i^{l-1} * \theta_{ij}^l + b_j^l\right),\ j = 1, \dots, N \tag{2.26}$$

where x_i^{l-1} represents the ith input map, x_j^l represents the jth output map, θ_{ij}^l represents the j th kernel connected to the i th 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 layer is $W \times H \times D$:

$$W = \frac{W_0 - F}{S} \tag{2.27}$$

$$H = \frac{H_0 - F}{S} + 1 \tag{2.28}$$

$$D = N \tag{2.29}$$

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

Figure 2.14 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 padding is 0. The size of output map is $n \times n \times 1$, where the $n = m - a + 1$.

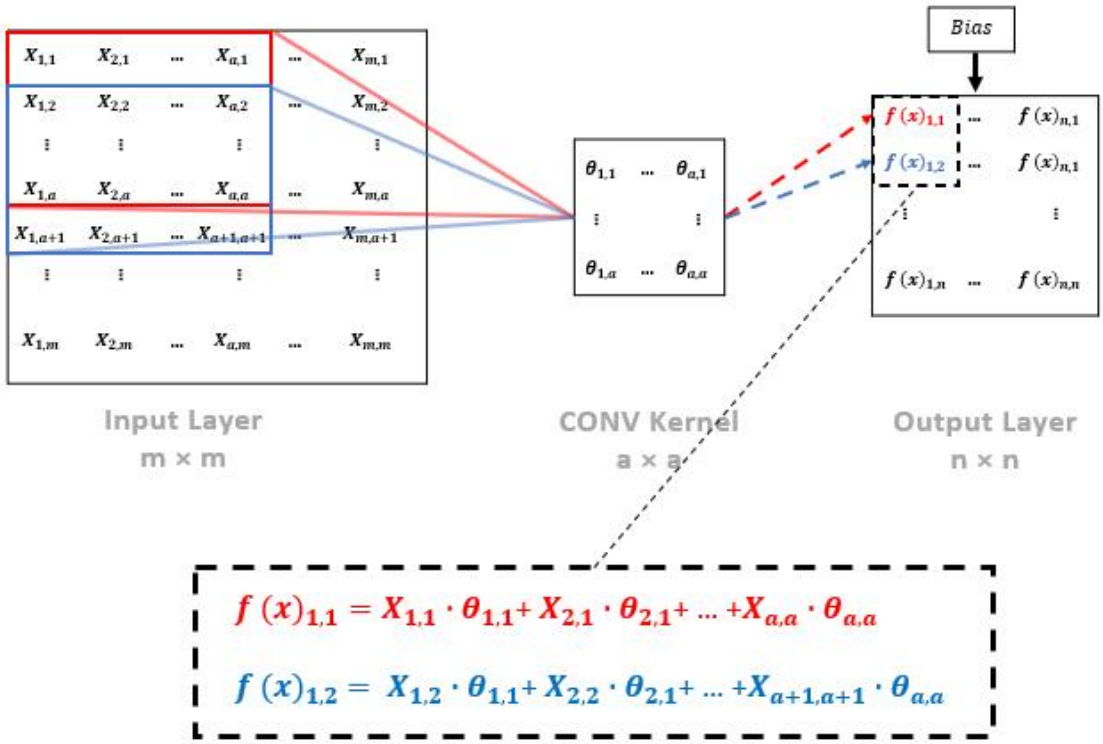

Figure 2.14 An example of convolutional operation

2.5.2. Pooling layer

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 (Hinton and Salakhutdinov, 2006). 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.

2.5.3. Fully connected layer

Convolution layers and pooling layers are used to extract the informative features. Fully connected layers are used to classify the features. Fully connected layers are

backpropagation neural networks with the inputs as 1-dimensional vectors. The fully connected layer generally utilizes the SoftMax as activation function to train the CNN model:

$$f(x_i)_{SoftMax} = \frac{e^{x_i}}{\sum_j^n e^{x_j}} \tag{2.30}$$

where e^{x_i} is the standard exponential function that is applied to each element of the input vector, the term of $\sum_j^n e^{x_j}$ is the normalization term, and n is number of classes in the classifier.

2.6. Summary

A process fault diagnosis system is to detect the occurrence of particular fault and diagnose the fault type, location, and time information. Performance evaluation of fault diagnosis system is predominantly based on the indicators of fault diagnosis speed, accuracy and robustness. Multitudinous previous proposed approaches can broadly be divided into three categories: model based approaches, knowledge based approaches, and data analysis based approaches. Among them, multivariate statistical analysis and machine learning in data-driven based approaches are the most popular research direction. In contemporary, deep learning shines brightly in the era of big data, especially the techniques of convolutional neural networks.

Chapter 3. Dataset Characteristics of a Typical Chemical Process and Benchmark of Fault Diagnostics

3.1. Introduction

This chapter gives a simulated continuous stirred tank reactor (CSTR) system, as a case study, to generate the simulated process dataset. The case study considers 11 typical faults and the on-line information sources include 10 measurement sources and 3 PI controller outputs. A traditional intelligent fault diagnosis system is developed as comparative study in this thesis. A set of abrupt faults are generated as training dataset to develop the fault diagnosis systems. The developed fault diagnosis schemes are tested on three different fault magnitudes and three diferent fault developing speeds for each of the 11 faults.

The chapter is organised as follows. Section 3.2 gives the principle of the CSTR system. Section 3.3 gives the introduction of abrupt fault and incipient fault. Section 3.4 gives the traditional neural network based fault diagnosis system. Section 3.5 concludes the chapter.

3.2. Principle in simulation of the CSTR system

The simulated CSTR system used in this study is taken from (Zhang, 1991), and is shown in Figure 3.1. This simulation system is developed based on mass and energy balance and reaction kinetics. An irreversible heterogeneous catalytic exothermic reaction takes place in the CSTR. The product concentration is indirectly maintained at a desired level by controlling temperature, residence time and mixing conditions in the CSTR. Part of the reactor outlet stream is recycled to the reactor through a heat exchanger to provide temperature control by manipulating the flow rate of the cold water fed to the heat exchanger via a cascade control system. The residence time is controlled through the reactor level controller and the mixing condition is controlled by maintaining the recycle flow rate. Constant physical properties and constant

boundary pressures of all input and output streams are assumed. Perfect mixing in the reactor is assumed. Under these assumptions, a mechanistic model is developed from mass balance and energy balance and is used to simulate the process. This mechanistic model generates the simulation process information of normal and abnormal operating with typical measurement noise.

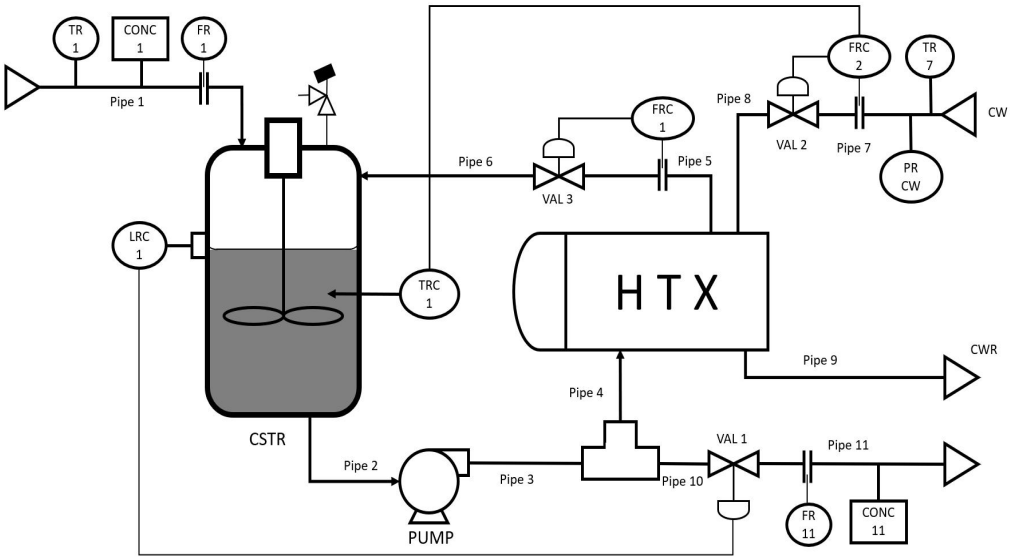

Figure 3.1 CSTR with a recycle

Principle of the CSTR system as following. First of all, assume an ideal condition of perfect mixing in the reactor; perfect heat exchange in heat exchanger; the reactant and product have same density; and the model is first order reaction.

The conservation of the mass (equivalently volume as density is constant) is:

$$A\frac{dH}{dt} = Q_1 + Q_2 - Q_3 \tag{3.1}$$

where A is the cross-sectional area in reactor, H is the tank level, Q_1, Q_2 and Q_3 represents the flowrates of input reactant, flowrate of recycled reactant and the liquid leaving the reactor, respectively.

The flowrate is related to the pressure drop across the valve as follows:

$$Q_2 = K_2 A_2 \sqrt{P} \tag{3.2}$$

$$Q_4 = K_4 A_4 \sqrt{P} \tag{3.3}$$

$$Q_5 = K_5 A_5 \sqrt{P_5} \tag{3.4}$$

$$Q_3 = Q_2 + Q_4 \tag{3.5}$$

where Q_4 is flowrate of product, Q_5 is the cold-water entering heat exchanger, P is the pressure of liquid leaving the pump, P_5 is the pressure of feed cold water to the heat exchanger, K is the restriction parameter of corresponding valve, A is the fractional opening of corresponding valve, and the indexes 2, 4 and 5 in K and A are associated to the valves 3, 1, and 2, respectively.

The concentration variation can be calculated by the equations:

$$AH\frac{dC_a}{dt} = Q_1(C_{a0} - C_a) - r_a AH \tag{3.6}$$

$$AH\frac{dC_b}{dt} = r_a AH - C_b Q_1 \tag{3.7}$$

$$r_a = K_r C_a \tag{3.8}$$

$$K_r = a_r e^{-\frac{b_r}{T}} \tag{3.9}$$

where C_a is the concentration of the reactant a in the reactor, C_{a0} is the concentration of reactant a in the input stream, $C_b$is the concentration of the product in the reactor, r_a is the reaction rate, K_r is the reaction rate constant, and a_r, b_r are constant peculiar to reaction.

The temperature variation can be calculated by the equations:

$$AHB_2\frac{dT}{dt} = B_1 Q_1(T_1 - T) - B_2 Q_2(T - T_2) + H_r r_a \tag{3.10}$$

$$T_2 = \frac{C_0\rho_0 Q_5 T_5 + Q_2 T[C\rho(C_a + C_b) + C_0\rho_0(1 - C_a - C_b)]}{C_0\rho_0 Q_5 + Q_2[C\rho(C_a + C_b) + C_0\rho_0(1 - C_a - C_b)} \tag{3.11}$$

$$B_1 = C_{a0}\rho C + (1 - C_{a0})\rho_0 C_0 \tag{3.12}$$

$$B_2 = C\rho(C_a + C_b) + C_0\rho_0(1 - C_a - C_b) \tag{3.13}$$

where T is the temperature in reactor, T_1 is the temperature of input reactant, T_2 is the temperature of the recycled reactant after heat exchange, H_r is the reaction heat

constant, C is the specific heat of reactant, C_0 is the specific hate of the solvent, ρ is the density of reactant, and ρ_0 is the density of the solvent.

The pressure variation can be calculated by the equations:

$$P = P_0 + \Delta P \tag{3.14}$$

$$P_0 = H[(C_a + C_b)\rho + (1 - C_a - C_b)\rho_0] \tag{3.15}$$

where P_0 is the pressure at the bottom of the reactor, and ΔP is the pressure increase caused by pump.

The system considers 11 process typical faults (Zhang and Roberts, 1992). The monitoring information consists of 3 controller outputs and 10 measured variables. The on-linear monitored variables are $[H, P, T, T_1, Q_1, Q_2, Q_4, Q_5, C_b, C_{a0}, uh, uq2, ut]$, where H represents the tank level, P represents the pressure of liquid leaving the pump, T and T_1 represents, respectively, the temperature in reactor, and temperature of input reactant, Q_1, Q_2, Q_4 and Q_5 represents, respectively, the flowrate of input reactant, the flowrate of recycled reactant, the flowrate of product, and the flowrate of cold-water entering heat exchanger, C_b and C_{a0} represents, respectively, the concentration of product in the reactor, and concentration of reactant in input stream, uh, $uq2$, and ut represents, respectively, the PI controller outputs for tank level, flowrate of recycled reactant, and temperature in reactor.

Figure 3.2 marks the locations of the 11 considered faults and the corresponding monitoring parameters or influencing factors (variation of valve opening), the locations of other monitoring parameters are also marked. Table 3.1 gives description of the considered faults. Table 3.2 gives 10 measured process variables with the simulated noise range.

Table 3.1 List of faults

Fault No.	Fault descriptions
1	Pipe 1 blockage
2	External feed-reactant flow rate too high
3	Pipe 2 or 3 is blocked or pump fails
4	Pipe 10 or 11 is blocked or control valve 1 fails low
5	External feed-reactant temperature abnormal
6	Control valve 2 fails high
7	Pipe 7, 8, or 9 is blocked or control valve 2 fails low
8	Control valve 1 fails high
9	Pipe 4, 5, or 6 is blocked or control valve 3 fails low
10	Control valve 3 fails high
11	External feed-reactant concentration too low

Table 3.2 Measurements Noise

Categories	Measurements	Noise Range
Level	Tank level (H)	-0.1 ~ 0.1 cm
Pressure	Pressure of liquid leaving the pump (P)	-1.5 ~ 1.5 g/cm^2
Flowrate	Flowrate of input reactant (Q_1) Flowrate of recycled reactant (Q_2) Flowrate of product (Q_4) Flowrate of cold-water entering heat exchanger (Q_5)	-1 ~ 1 cm^3/s
Concentration	Concentration of product in the reactor (C_b) Concentration of reactant in the input stream (C_{a0})	-0.5 % ~ 0.5 %
Temperature	Temperature in reactor (T) Temperature of input reactant (T_1)	-0.25 ~ 0.25 °C

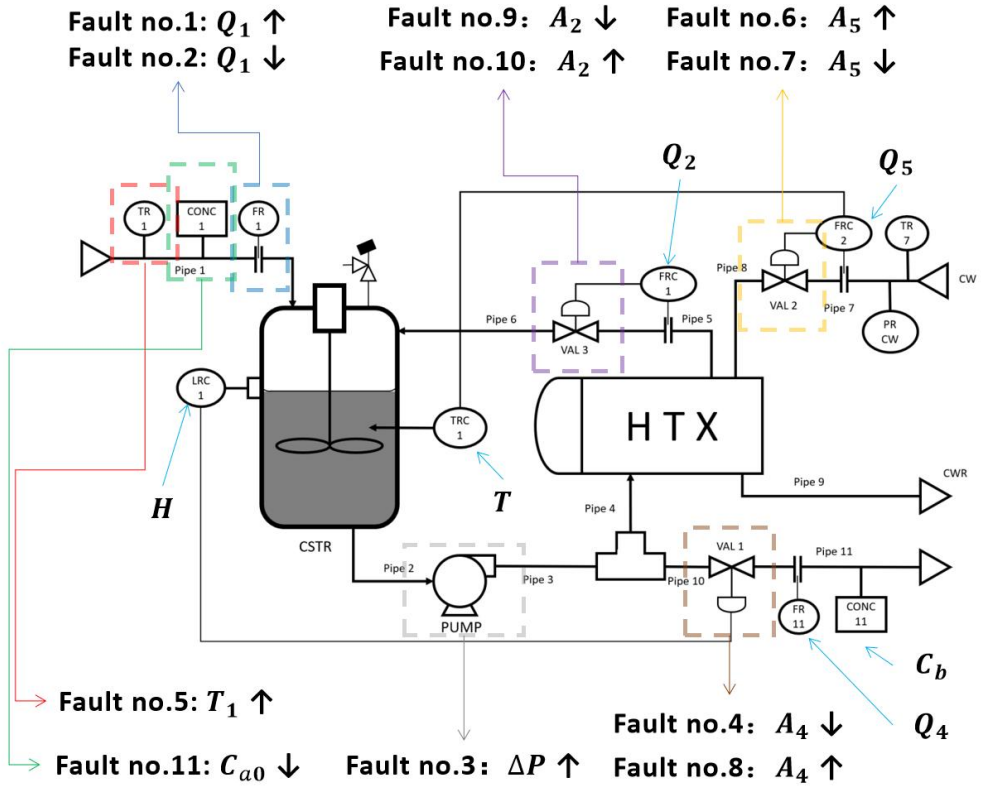

Figure 3.2 Location of faults and monitoring parameters

3.3. Abrupt faults and Incipient faults

The chemical industrial process faults can be broadly split to 2 types: abrupt faults and incipient faults. An abrupt fault occurring in a process indicates that the process variable(s) related to the fault has a suddenly stepwise change. An incipient fault occurring in a process indicates that fault magnitude slowly increase with time. As the incipient fault develops, without detection and management, the accumulation of damage increases in severity, which can result in the production processes suffer the seriously abnormal conditions.

An incipient fault can be expressed as the following equation:

$$M_f(t) = M_n(1 + \gamma t) \tag{3.16}$$

where M_n is the nominal value of a process parameter, $M_f(t)$ is the faulty value of the same parameter at time t, γ is the fault developing speed, and t is the time from the occurrence of a fault. This equation can represent common incipient faults.

3.4. Baseline fault diagnosis scheme in the CSTR system

A baseline fault diagnosis scheme is developed as comparative study in this thesis. It is a traditional shallow neural network based diagnosis scheme. Figure 3.3 gives the structure of the baseline fault diagnosis scheme. The scheme first directly scales the measured information by the following equation:

$$X_{i,p} = \frac{X_i - \bar{X}_{i,normal}}{X_{std}} \tag{3.17}$$

where X_i is the actual value, $X_{i,p}$ is pre-processed values for the i th on-line measurement, $\bar{X}_{i,normal}$ is the mean value of normal data, and X_{std} is the standard deviation of normal data.

The neural network is a shallow network, i.e., has only one hidden layer. It has 11 output layer neurons corresponding to the 11 faults. The network is trained using the backpropagation training method with the learning rate, momentum constant, and maximum training steps selected as 0.01, 0.9 and 1000 respectively. The training objective is to minimize the mean squared errors of the network. The neural network is trained on the training data set and the testing data set is used for network structure determination and implementing the "early stopping" mechanism during network training (Zhang, 1999). Activation in the neural network is sigmoid function.

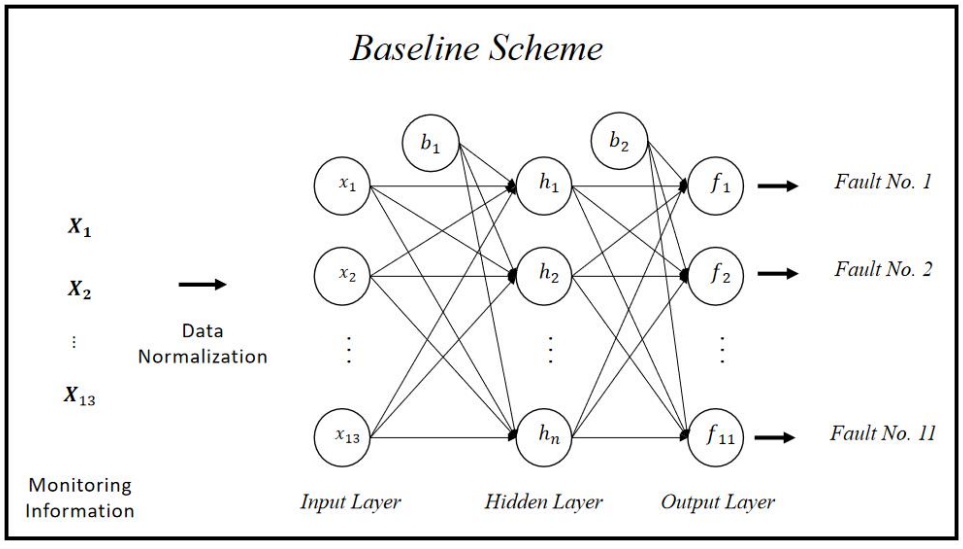

Figure 3.3 Structure of the baseline scheme

Chapter 4. Andrews Function based Intelligent Fault Diagnosis System

4.1. Introduction

This chapter proposes a fault diagnosis system based on Andrews plot and neural network. The core idea is utilizing the function of Andrews plot to extract the features from process information as inputs instead of the original information, before fed into the classifier. Andrews plot is a very useful visualization method for analysing multivariate data. Its unique advantage includes the convenience in setting the dimension of the extracted feature space. Andrews plot is efficient in dealing with the issues with large number of correlated variables. A difficulty with Andrews plot is that the proper selection of the number of features. Nevertheless, Andrews plot has a great potential for improving fault detection and diagnosis performance through extracting useful features from the original monitored process measurements. The original process information in this work is replaced by its principal components as the inputs to Andrews function, due to the Andrews function in practice is highly sensitive to data arrangement. This work presents a method for determining the important features in Andrews function which give good separations between classes. The classifier uses the same method as the baseline traditional scheme, i.e., a neural network with single hidden layer. The proposed fault diagnosis method is applied to a simulated continuous stirred tank reactor (CSTR) system. In order to demonstrate its superiority, it is compared with two traditional neural network based diagnosis schemes.

The chapter is organised as follows. Section 4.2 gives some Andrews functions. Section 4.3 gives the operation of Andrews function processing in process industries. Section 4.4 gives the proposed fault diagnosis scheme. Section 4.5 gives the simulated process data in this work, the development of the proposed scheme A and the baseline scheme, and the diagnosis results. Section 4.6 concludes the chapter.

4.2. Andrews plot

Andrews plot (Andrews, 1972) is a visualization method for high-dimensional data and is named after the statistician David F. Andrews. Andrews plot has been applied in many areas such as analysing data from the 2001 Parliamentary General Election in the United Kingdom (Spencer, 2003; Khattree and Naik, 2002). Its unique advantage lies in the convenience in adjusting the matrix dimension during the modelling process. Andrews plot is efficient in dealing with the issues with large number of variables. Andrews (1972) proposed the function that each a -dimensional data, $X = (x_1, x_2, x_3, \cdots, x_a)$, can be mapped into a curve using the following function:

$$f_x(t) = \frac{x_1}{\sqrt{2}} + x_2 sint + x_3 cost + x_4 sin2t + \cdots; \quad (4.1)$$

where t is in the range [-π, π]. Each data point can be viewed as a curve in the interval [-π, π]. Andrews also suggested other functions as the following to be used in the curves.

$$f_x(t) = x_1 sin\, n_1 t + x_2 cos\, n_2 t + x_3 sin\, n_3 t + \cdots; \quad (4.2)$$

$$f_x(t) = x_1 sin2t + x_2 cos2t + x_3 sin4t + \cdots; \quad (4.3)$$

where n_i are different integers and *t* is in the range [-π, π].

In prior works, several researchers also suggested different variations to the functions of Andrews plot. The following functions are suggested in (Khattree and Naik, 2002; Kulkarni and Paranjape, 1984; Wegman and Shen, 1993):

$$f_x(t) = x_1 sint + x_2 cost + x_3 sin2t + \cdots; \quad (4.4)$$

$$f_x(t) = x_1 sin\, \omega_1 t + x_2 cos\, \omega_2 t + x_3 sin\, \omega_3 t + \cdots; \quad (4.5)$$

$$f_x(t) = \frac{1}{\sqrt{2}}\{x_1 + x_2(sint + cost) + x_3(sint - cost) + x_4(sin2t + cos2t) + \cdots\}; \quad (4.6)$$

where t is in the range [-π, π] and the values of ω_i are mutually irrational and scaled between 0.5 and 1.

The following two curves are an example express how the Andrews plot convert the process data. The original data has 50 samples with 13 variables. After the computation by Andrews function (4.1) with 9 *t*-values, the data are changed to 9 dimensions.

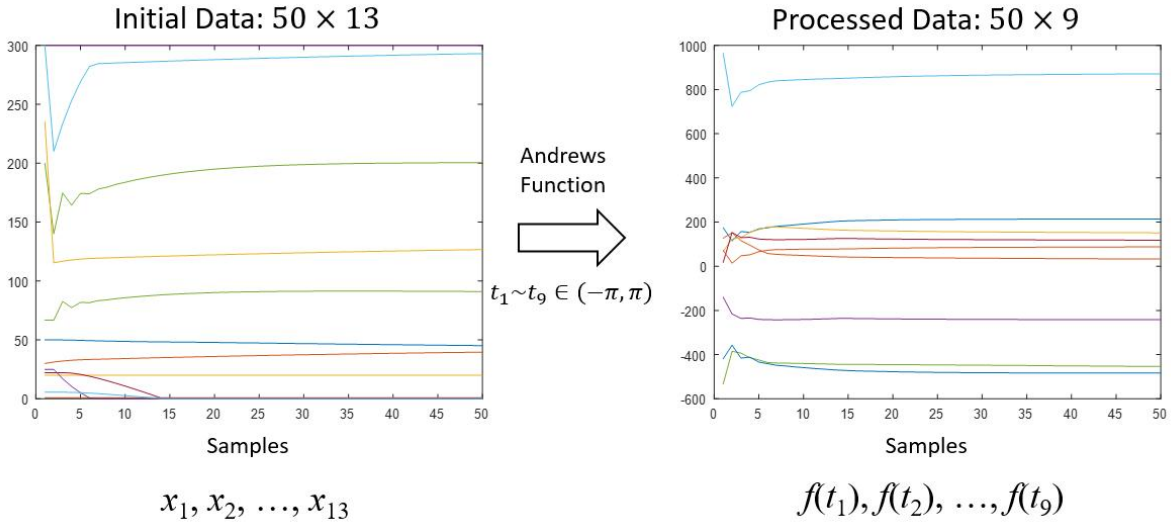

Figure 4.1 An example of Andrews function convert the input data

4.3. Andrews function processing in process industries

The feature extraction capability of Andrews plot has great potential in chemical industrial process fault diagnosis. Utilizing the Andrews function to process information, the dimension of data can be changed to an appropriate dimension, by selecting a different number of *t*-values. In chemical process monitoring applications, each sampling point, i.e. a multi-variables vector, using the Andrews function with several numbers of *t*-values, can convert into an information-contain vector with a new dimensionalities corresponding to numbers of *t*-values. In theory, the dimensions of feature vector can be changed unrestricted, due to the number of *t*-value selection is unlimited, if the final performance is disregarded. This superiority can beneficially use in data analysis.

Andrews function in practice is affected by the arrangement of multivariable data (Boonprong, Cao, Torteeka, and Chen, 2017). A general solution for this problem is to use the principal components instead the original information as the inputs to Andrews

funtion. Therefore, the operation of Andrews function processing the measurements as follows and the procedure of Andrews function processing of the original data is shown in Figure 4.2, which indicates that the follow steps are carried out.

Step 1. The original information is normalized by statistics methods, such as standard score, or Min-max feature scaling.

Step 2. The principal components instead of the scaled original information are used as the inputs for Andrews function.

Step 3. Several appropriate *t*-values are selected for Andrews function calculation.

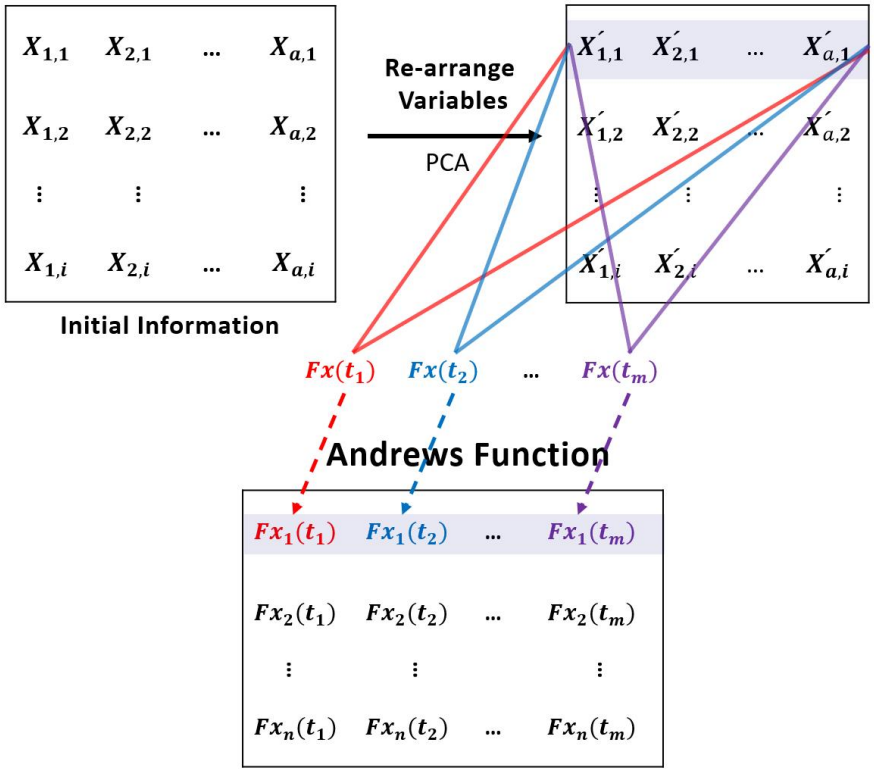

Figure 4.2 Data processing using Andrews function

The offline training stage in this work uses the simulated process operation dataset on CSTR system. As previously mentioned, the system uses the principal components X' instead the original dataset X as the inputs to Andrews function. Then Andrews

function with r t-value is used to obtain the information-contained feature dataset $F_x(t)$ from X'. Then $F_x(t)$ is fed into a neural network to train the fault classifier.

The training dataset X is given as:

$$X_{sample,var} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{s1,13} \\ x_{2,1} & x_{2,2} & & x_{s2,13} \\ \vdots & & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,13} \end{bmatrix} \tag{4.7}$$

The $U, \sum$, and V matrices from singular value decomposition are given as:

$$U = \begin{bmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,m} \\ u_{2,1} & u_{2,2} & & u_{2,m} \\ \vdots & & \ddots & \vdots \\ u_{m,1} & u_{m,2} & \cdots & u_{m,m} \end{bmatrix} \tag{4.8}$$

$$\sum = \begin{bmatrix} \sigma_{1,1} & 0 & 0 & 0 \\ 0 & \sigma_{2,2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & \vdots & 0 & \sigma_{13,13} \\ 0 & 0 & 0 & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.9}$$

$$V = \begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,13} \\ v_{2,1} & v_{2,2} & & v_{2,13} \\ \vdots & & \ddots & \vdots \\ v_{13,1} & v_{13,2} & \cdots & v_{13,13} \end{bmatrix} \tag{4.10}$$

The principal component dataset X' is calculated as:

$$X' = XV = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,13} \\ x_{2,1} & x_{2,2} & & x_{2,13} \\ \vdots & & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,13} \end{bmatrix} \cdot \begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,13} \\ v_{2,1} & v_{2,2} & & v_{2,13} \\ \vdots & & \ddots & \vdots \\ v_{13,1} & v_{13,2} & \cdots & v_{13,13} \end{bmatrix} =$$

$$\begin{bmatrix} x'_{1,1} & x'_{1,2} & \cdots & x'_{1,13} \\ x'_{2,1} & x'_{2,2} & & x'_{2,13} \\ \vdots & & \ddots & \vdots \\ x'_{m,1} & x'_{m,2} & \cdots & x'_{m,13} \end{bmatrix} \tag{4.11}$$

The Andrews function output $F_x(t)$ is given as:

$$F_x(t) = \begin{bmatrix} F_{x_1}(t_1) & F_{x_1}(t_2) & \cdots & F_{x_1}(t_r) \\ F_{x_2}(t_1) & F_{x_2}(t_2) & & F_{x_2}(t_r) \\ \vdots & & \ddots & \vdots \\ F_{x_m}(t_1) & F_{x_m}(t_2) & \cdots & F_{x_m}(t_r) \end{bmatrix} \tag{4.12}$$

$$\begin{cases} F_{x_1}(t_1) = \dfrac{x_{1,1}}{\sqrt{2}} + x_{1,2}sint_1 + x_{1,3}cost_1 + x_{1,4}sin2t_1 + x_{1,5}cos2t_1 + x_{1,6}sin3t_1 \\ \quad + x_{1,7}cos3t_1 + x_{1,8}sin4t_1 + x_{1,9}cos4t_1 + x_{1,10}sin5t_1 + x_{1,11}cos5t_1 \\ \quad + x_{1,12}sin6t_1 + x_{1,13}cos6t_1 \\ \vdots \\ F_{x_2}(t_2) = \dfrac{x_{2,1}}{\sqrt{2}} + x_{2,2}sint_2 + x_{2,3}cost_2 + x_{2,4}sin2t_2 + x_{2,5}cos2t_2 + x_{2,6}sin3t_2 \\ \quad + x_{2,7}cos3t_2 + x_{2,8}sin4t_2 + x_{2,9}cos4t_2 + x_{2,10}sin5t_2 + x_{2,11}cos5t_2 \\ \quad + x_{2,12}sin6t_2 + x_{2,13}cos6t_2 \\ \vdots \\ F_{x_m}(t_r) = \dfrac{x_{m,1}}{\sqrt{2}} + x_{m,2}sint_r + x_{m,3}cost_r + x_{m,4}sin2t_r + x_{m,5}cos2t_r + x_{m,6}sin3t_r \\ \quad + x_{m,7}cos3t_r + x_{m,8}sin4t_r + x_{m,9}cos4t_r + x_{m,10}sin5t_r + x_{m,11}cos5t_r \\ \quad + x_{m,12}sin6t_r + x_{m,13}cos6t_r \end{cases}$$

Figure 4.3 shows the comparisons in different variables ordering. Process information includes 10 monitored variables and 3 PI controller outputs. The first process variable arrangement is $[T, H, T_1, Q_1, Q_2, Q_4, Q_5, C_b, C_{a0}, P, uh, uq2, ut]$, where uh, $uq2$, and ut represents, respectively, the PI controller output for tank level, flowrate of recycled reactant, and temperature in reactor. The second process variable arrangement is $[T, H, T_1, uh, C_b, C_{a0}, P, Q_1, Q_2, Q_4, Q_5, uh, uq2]$. The third process variable arrangement is $[T, H, uq2, ut, C_{a0}, P, uh, T_1, Q_1, Q_2, Q_4, Q_5, C_b, C_{a0}]$. The process data in this case is generated from the simulated CSTR system under fault no.11 and contains 50 samples. The results shown in Figure 4.3 are produced using 100 *t*-values uniformly distributed in the range [-π, π]. The right column shows the results using original process information normalized by standard scores, and the left column shows the results using its principal components instead. The upper red curves represent the first variable ordering, the centre blue curves represent the second variable ordering, and the bottom dark cyan curves represent the third variable

ordering. Evidently, the results from using the principal components in Andrews function processing can guard against the impacts from process variables arrangement. Thus, fault diagnosis system can consequently maintain the output stability.

Figure 4.4 gives an example to illustrate why the data pre-processing by Andrews function can potentially work better than the baseline scheme. The process data used in this case are generated from the simulated CSTR system. The curves in the upper plot are data pre-processed by Andrews function method, and in the bottom plot are data pre-processed by the baseline scheme. In order to obtain a clear visualization, the numbers of *t*-values are equal to the number of variables in the baseline scheme, i.e., 13 *t*-values. In both plots, the black curves represent the nominal data, the red curves represent the data with fault no.1, the blue curves represent the data with fault no.3, and the orange curves represent the data with fault no.10. Each set of data consists of 50 samples. The results show that data pre-processed by Andrews function can enhance the degree of data separation between the nominal data and data containing faults, and more dimensions are involved in the fault diagnosis. Hence, the Andrews plot method participating in the development of diagnosis system can give a desirable expectation.

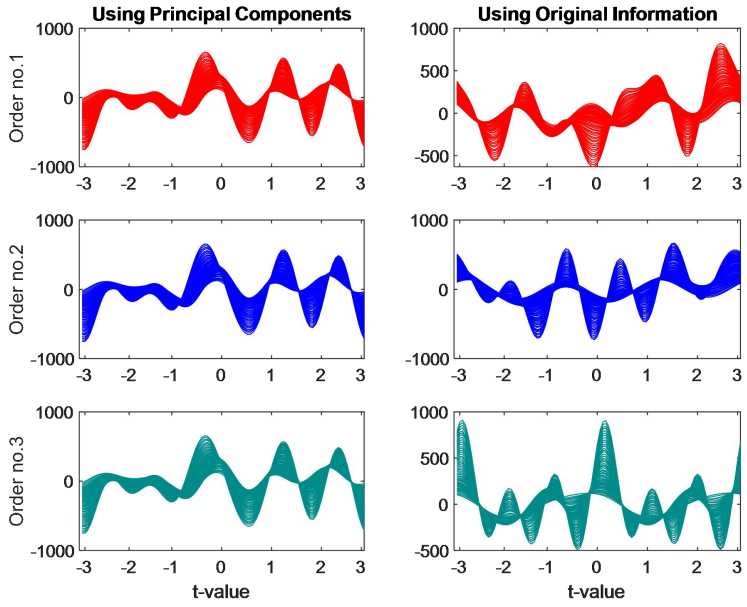

Figure 4.3 Results of Andrews function for the data under fault No. 11 with different variable orderings

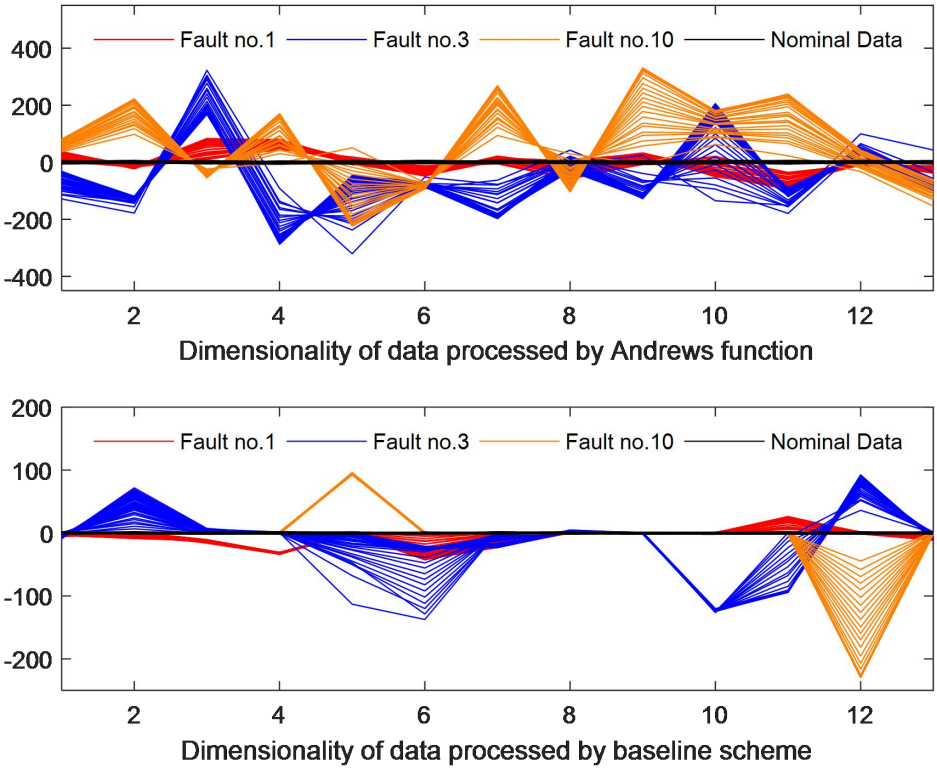

Figure 4.4 Comparison of data pre-processed by Andrews function and the baseline scheme

4.4. Proposed Fault Diagnosis System

It is expected that a fault diagnosis system combining neural network with Andrews plot can give enhanced fault diagnosis results. Figure 4.5 shows the framework of the proposed diagnosis scheme. In this work, the proposed fault diagnosis scheme (referred to as scheme A) consists of two subsystems according to their respective functions: data pre-processing system and process state analysis system. Data pre-processing system is used to pre-process the online measurements based on Andrews function, and the processed features are fed into process state analysis system, which is the classifier based on a neural network. The establishment of the proposed fault diagnosis system as the following steps:

Step 1. Data are re-arranged by using principal components.

Step 2. Information containing features are extracted from the principal components by using Andrews function with several selected *t*-values.

Step 3. The extracted features are normalized and fed into a neural network to obtain the diagnosis outputs.

In the system development, the values of t in Andrews function are of high importance. Different number of t and their values can affect the outputs of Andrews function processing and further exert influence on the final diagnosis results. Hence, the fault diagnosis system developing requires an appropriate number of t with appropriate values, even though this step is time-consuming. The determination scheme of number and value of t is as follow:

Step 1. Using the Andrews function to process the data with a relatively large number (e.g., 100) of *t*-values, which are uniformly distributed in the range $-\pi$ to π.

Step 2. The minimum distance between all the classes (normal, fault No. 1, …, fault No. 11) is calculated at each of these *t*-values.

Step 3. Then the appropriate *t*-values are selected according to the observation on these minimum distances.

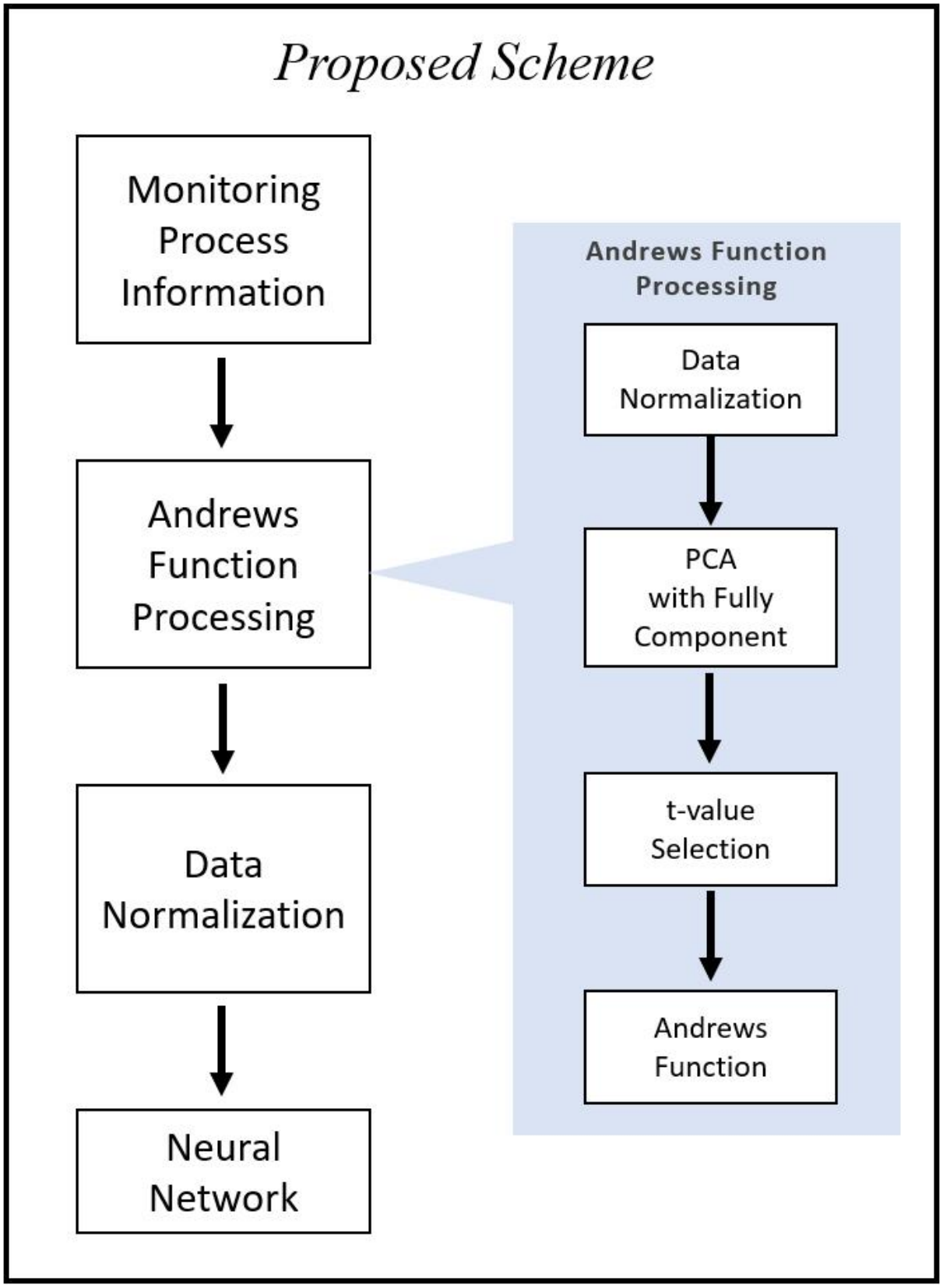

Figure 4.5 Framework of the fault diagnosis scheme

4.5. Fault Diagnosis Results

4.5.1. Simulated Process Data

Table 4.1 gives the relative magnitudes corresponding to the 11 faults. These values are utilized in the simulated CSTR system to generate the considered fault dataset for neural network training. For each fault, 80 samples were collected when one or more of the process variables exist 3 times of their normal standard deviations. The normal operation data also contains 80 samples. For the 80 samples corresponding to each category, 50 samples were randomly selected as training data while the remaining 30 samples were used as the testing data. Thus, the training data set contains 600 samples, and the testing data sets contains 360 samples. In this study, three groups of abrupt faults and three groups of incipient faults under different fault conditions were generated as unseen validation data for evaluating the developed fault diagnosis systems. Table 4.2 gives three groups of relative fault magnitudes (Mag. 1, Mag. 2, and Mag. 3) corresponding to the 11 abrupt faults. Table 4.3 gives three groups of faults developing speeds (γ1, γ2, and γ3) corresponding to the 11 incipient faults.

Table 4.1 Process parameter values for building the neural network models

Fault No.	**Related Variables**	**Relative Magnitudes**
1	Flow rate of the input reactant (Q_1)	6.67%
2	Flow rate of the input reactant (Q_1)	6.67%
3	Pressure increase caused by pump (ΔP)	60%
4	Fractional opening of valve 1 (A_4)	77.78%
5	Temperature of input reactant (T_1)	20%
6	Fractional opening of valve 2 (A_5)	54.54%
7	Fractional opening of valve 2 (A_5)	90.91%
8	Fractional opening of valve 1 (A_4)	11.11%
9	Fractional opening of valve 3 (A_2)	75%
10	Fractional opening of valve 3 (A_2)	25%
11	Concentration of the reactant in the input stream (C_{a0})	50%

Table 4.2 Relative magnitudes in abrupt faults

Fault No.	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%

Table 4.3 Fault developing speeds in incipient faults

Fault No.	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}

4.5.2. Fault Diagnosis System Development

The performance of the proposed diagnosis scheme A is compared with the baseline conventional scheme under abrupt faults and incipient faults. In the proposed scheme A, the selection of *t*-values (the number of features) in Andrews function is important and can affect the final diagnosis performance. In practical application, since the infinite number of *t*-values in Andrews function and large number of variables in a

chemical engineering process, the selection of appropriate *t*-values may be time consuming. The method for determining the important features (*t*-values) presented in section 4.4 is used here. Andrews function values for all the data at 100 t-values uniformly distributed in the range [– π, π] are calculated. The minimum distance between all the classes (normal, fault No. 1, …, fault No. 11) is calculated at each of these t-values. Figure 4.6 shows the top 30 minimal distance values in descending order. It can be seen that the minimal distance values drop quickly after about the first 11 *t*-values. Therefore, the first 11 *t*-values are selected, i.e., the final selected values of *t* are [-3.1414, -0.6846, -0.5586, -0.4956, -0.4326, -0.3696, 0.2604, 2.9064, 209694, 3.3024, 3.0954].

During network training, a target output of 1 is assigned to the network output corresponding to the fault while the targets for other network outputs are 0. During diagnosis, the neural network outputs should be lower than 0.2 when the samples are under the normal operating condition. A diagnosis result is issued when the neural network output corresponding to the fault is higher than 0.8 while other network outputs are below 0.2. An advance warning is issued when the neural network output corresponding to the fault is higher than 0.4. Evaluation of fault diagnosis systems is mainly based on accuracy, robustness, and speed of diagnosis. The diagnosis time is measured as the time between a fault being initiated and being successfully diagnosed. The advance warning time is de-fined as the time between the fault being initiated and a correct advance warning being issued.

The number of hidden neurons was determined through cross validation on the testing data. A number of neural networks with different numbers of hidden neurons were trained on the training data and tested on the testing data. The network gives the overall best performance on the testing data is considered as having the appropriate number of hidden neurons. Table 4.4 gives the accuracy of different numbers of hidden neurons in the proposed scheme on the testing data. The best performance is marked with bold font. Hence the number of neurons in hidden layer is 15. Left of the table gives the neuron numbers in different layer of the neural network. Table 4.5 gives the

neuron numbers and accuracy of different numbers of hidden neurons in the baseline traditional scheme.

The following performance table gives the fault diagnosis time and fault detection rate for each considered faults and their average value to express the system performance. In the subsequent diagnosis output plots, F1 to F11 represent the neural network outputs corresponding to Fault No. 1 to No. 11 listed in Table 3.1 respectively. The upper dash–doted straight lines indicate the diagnosis threshold (0.8) while the lower dash–dotted lines indicate the advance warning threshold (0.4).

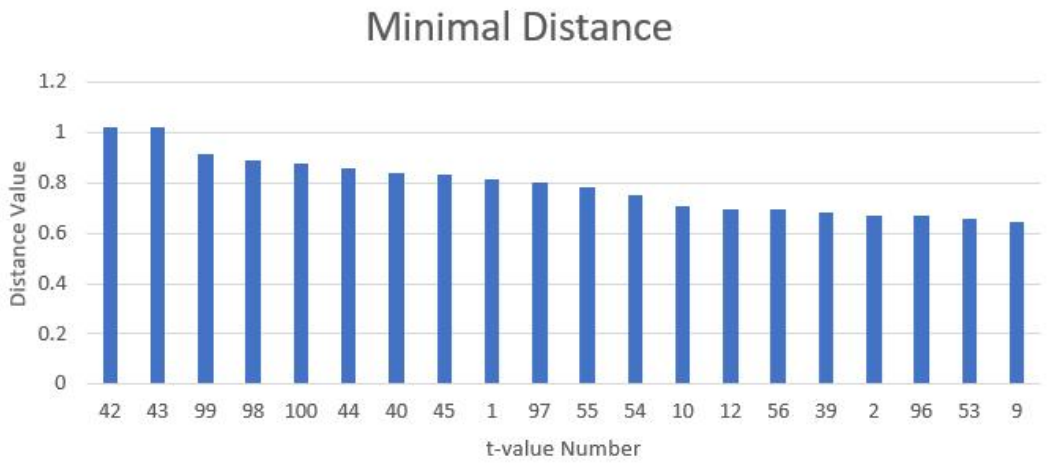

Figure 4.6 The top 30 minimal distance values in descending order

Table 4.4 Determination of the Numbers of Hidden Neurons in the Proposed Scheme

Proposed Scheme A							
Determined Neuron Number		**Accuracy of Different Numbers of Hidden Neuron**					
Layer	**Neuron**	Numbers of HN	Accuracy	Numbers of HN	Accuracy	Numbers of HN	Accuracy
Input	**11**	12	90.56%	**15**	**96.94%**	18	91.39%
Hidden	**15**	13	95.27%	16	92.22%	19	88.61%
Output	**11**	14	93.61%	17	90.83%	20	84.17%

Table 4.5 Determined Numbers of Hidden Neurons in the Baseline Scheme

Baseline Traditional Scheme							
Determined Neuron Number		**Accuracy of Different Numbers of Hidden Neurons**					
Layer	**Neuron**	Numbers of HN	Accuracy	Numbers of HN	Accuracy	Numbers of HN	Accuracy
Input	**13**	12	71.67%	15	75%	18	71.67%
Hidden	**17**	13	68.33%	16	62.78%	19	68.33%
Output	**11**	14	64.72%	**17**	**76.67%**	20	58.06%

4.5.3. Performance under Abrupt Faults

The proposed scheme A and the baseline conventional scheme are tested on three groups of abrupt faults given by Table 4.2 to demonstrate the superiorities of the proposed method. All of the abrupt faults were initiated at 40 s. Table 4.6 indicates that both diagnosis systems successfully diagnosed all the considered abrupt faults. In terms of diagnosis speed, the proposed diagnosis scheme A diagnosed the abrupt faults 5.45 s earlier on average than the baseline scheme.

Note that as the sampling time is given, traditional metrics like the fault detection rate (FDR) and the missed detection rate (MDR) can be easily obtained from the fault diagnosis times given in Table 4.6. Table 4.7 gives the FDR for abrupt faults. In all the considered faults here, there are no incorrect diagnosis cases and the undetected samples are during the early stages of the faults. Thus, for all the considered faults here, MDR can be simply obtained as 1 - FDR. It can be seen from Table 4.7 that the proposed scheme A gives overall higher FDR than the baseline scheme. Table 4.7 also indicates that the FDR is higher when the fault magnitude is higher as a fault with higher magnitude is generally easier to detect and diagnosis than the same fault with lower magnitude. Figures 4.7 to 4.10 show two sets of diagnosis outputs in abrupt faults as examples to show the robustness of the proposed scheme.

Table 4.6 Fault Diagnosis Time in Abrupt Faults

Fault No.	Diagnosis time (s)					
	Proposed Scheme A			Baseline Scheme		
	Mag. 1	Mag. 2	Mag. 3	Mag. 1	Mag. 2	Mag. 3
1	24	12	4	48	24	12
2	20	8	4	32	12	12
3	36	28	12	44	32	28
4	32	12	4	32	28	8
5	32	16	8	36	24	8
6	60	28	12	12	8	8
7	36	16	8	52	36	16
8	24	8	8	52	20	4
9	20	8	8	40	8	8
10	20	8	4	36	20	4
11	16	8	8	12	8	8
Average		16.73			22.18	

Table 4.7 Fault Detection Rate in Abrupt Faults

Fault No.	Fault Detection Rate (FDR)					
	Scheme A			Baseline Scheme		
	Mag. 1	Mag. 2	Mag. 3	Mag. 1	Mag. 2	Mag. 3
1	85.0%	92.5%	97.5%	70.0%	85.0%	92.5%
2	87.5%	95.0%	97.5%	80.0%	92.5%	92.5%
3	77.5%	82.5%	92.5%	72.5%	80.0%	82.5%
4	80.0%	92.5%	97.5%	80.0%	82.5%	95.0%
5	80.0%	90.0%	95.0%	77.5%	85.0%	95.0%
6	62.5%	82.5%	92.5%	92.5%	95.0%	95.0%
7	77.5%	90.0%	95.0%	67.5%	77.5%	90.0%
8	85.0%	95.0%	95.0%	67.5%	87.5%	97.5%
9	87.5%	95.0%	95.0%	75.0%	95.0%	95.0%
10	87.5%	95.0%	97.5%	77.5%	87.5%	97.5%
11	90.0%	95.0%	95.0%	92.5%	95.0%	95.0%
Average		89.6%			86.1%	

Figure 4.7 and Figure 4.8 show the performance of scheme A and the baseline scheme in diagnosing abrupt fault No. 2, with the fault relative magnitude being 1.67%. In Figure 4.7 and the subsequent diagnosis output plots, F1 to F11 represent the neural network outputs corresponding to Fault No. 1 to No. 11 listed in Table 3.1 respectively. The upper dash-dotted lines indicate the diagnosis threshold (0.8) while the lower dash-dotted lines indicate the advance warning threshold (0.4). Figure 4.7 shows that scheme A successfully diagnosed the fault at 20 s after the fault being initiated. Figure 4.8 shows that the outputs from the baseline scheme responded quickly, when the abnormal condition occurred but with fluctuations. Eventually the fault was diagnosed at 32 s after the fault being initiated. The classifier output curve then still oscillates with a certain magnitude over the 0.8. It can be seen that the proposed scheme A diagnosed the fault 12 s earlier than the baseline scheme and the classifier output curve is steadier than that of the baseline scheme.

Figures 4.9 and 4.10 show the performance of scheme A and the baseline scheme in diagnosing abrupt fault No. 7, with the fault relative magnitude being 27.86%. As in the prior case, the outputs in scheme A have isolated points exceeding the diagnosis threshold (0.8) before the oscillation stabilized. Figure 4.9 shows that scheme A successfully diagnosed the fault at 16 s after the fault being initiated. Figure 4.10 shows that the baseline scheme successfully diagnosed the fault at 36 s after fault being initiated. The diagnosis time of scheme A for this fault is 20 s shorter than that of the baseline scheme.

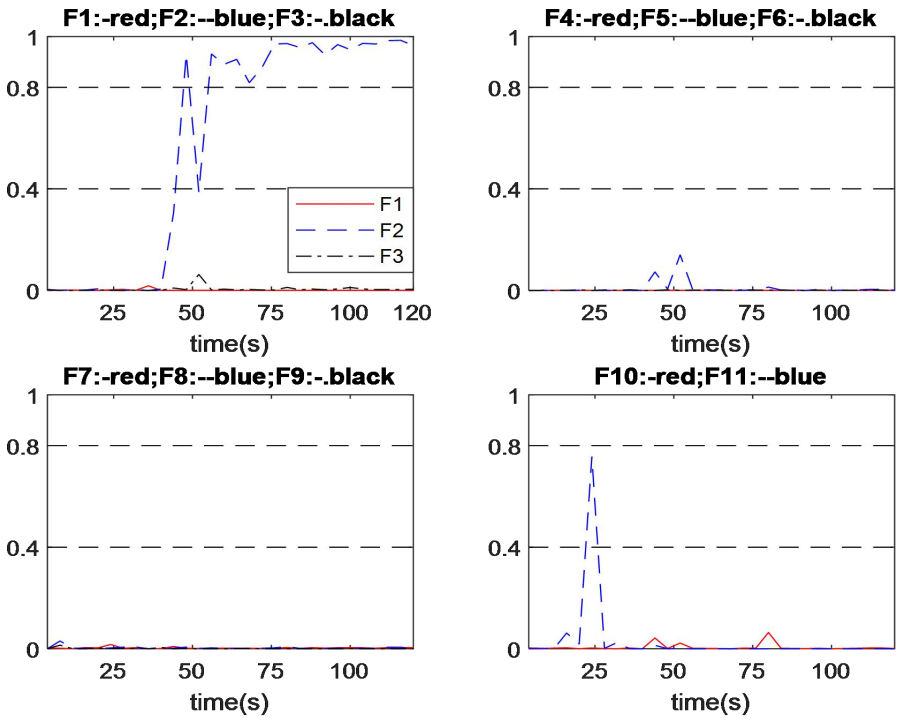

Figure 4.7 Output of fault no.2 diagnosis in abrupt fault with scheme A with relative magnitude of 1.67% (No.1)

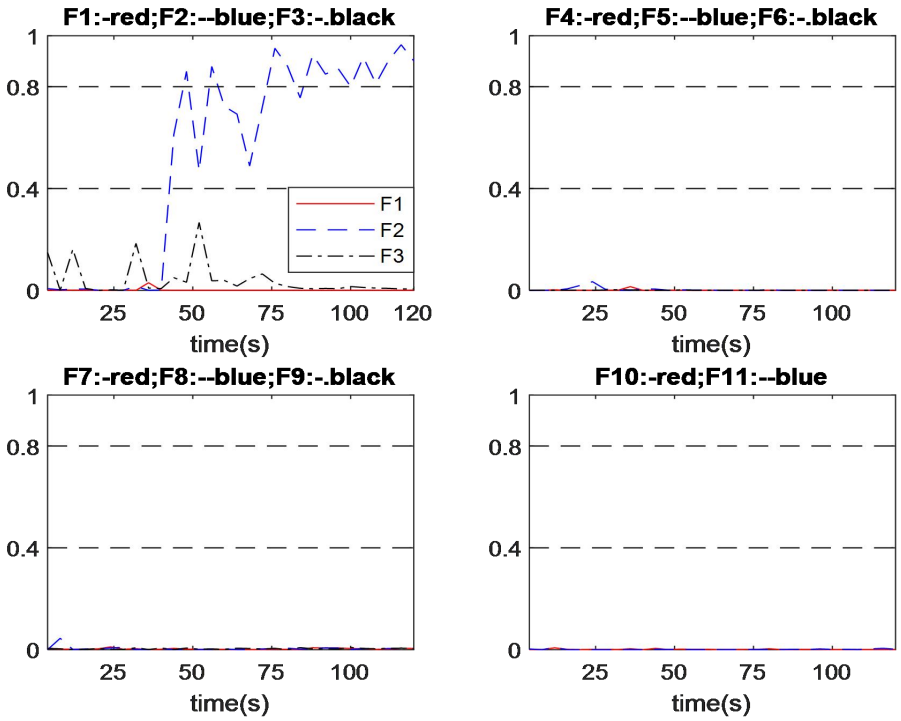

Figure 4.8 Output of fault no.2 diagnosis in abrupt fault with the baseline scheme with relative magnitude of 1.67% (No.1)

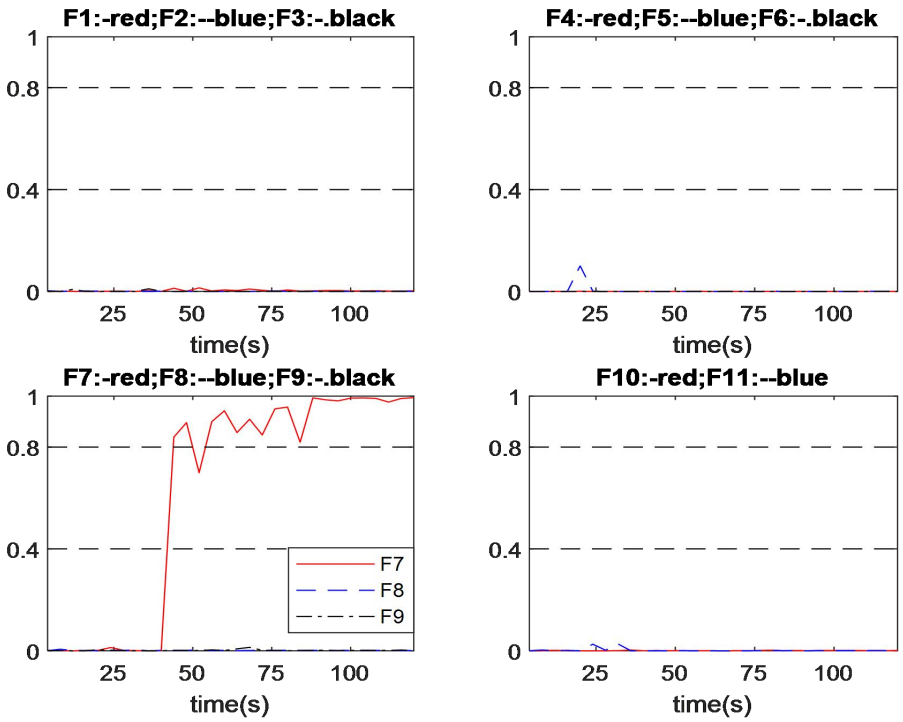

Figure 4.9 Output of fault no.7 diagnosis in abrupt fault with scheme A with relative magnitude of 27.86%
(No.2)

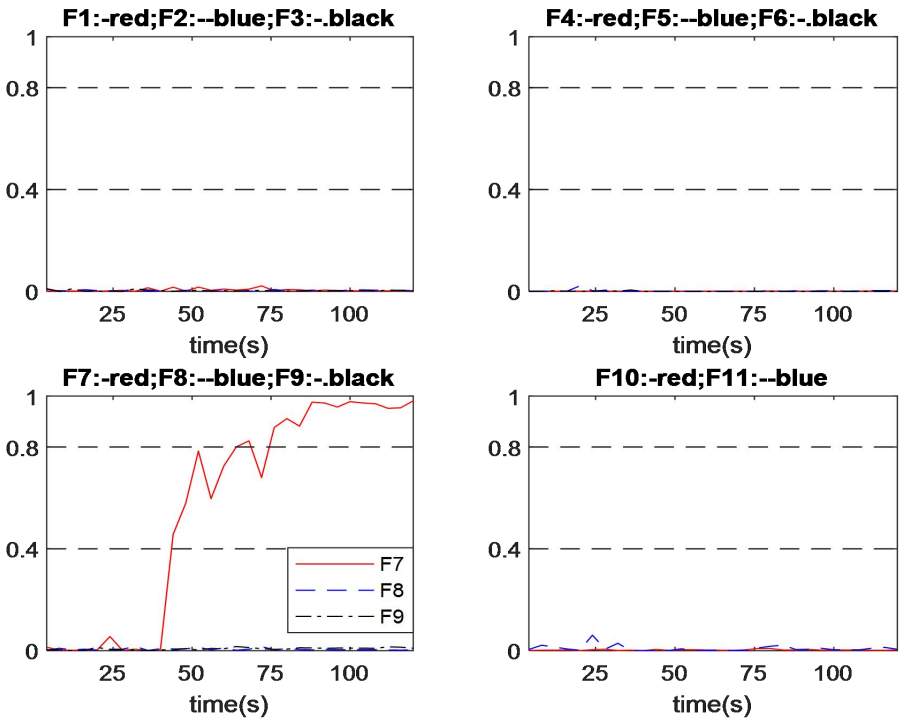

Figure 4.10 Output of fault no.7 diagnosis in abrupt fault with the baseline scheme with relative magnitude of 27.86% (No.2)

4.5.4. Performance under Incipient Faults

The proposed scheme A and the baseline scheme are applied to three groups of incipient faults given by Table 4.3 to compare fault diagnosis results. All of the incipient faults were initiated at 0 s. Table 4.8 indicates that both diagnosis systems successfully diagnosed all the considered incipient faults. In terms of the diagnosis speed, the proposed diagnosis scheme A diagnosed the faults 13.82 s earlier on average than the baseline scheme for incipient faults.

Table 4.9 gives the FDR values for the considered incipient faults of the three schemes. In can be seen that the proposed scheme A gives overall higher FDR than the baseline scheme. Table 4.9 also indicates that the FDR is higher when the fault developing speed is higher as a faster developing fault is generally easier to detect and diagnosis than a slower developing fault. As with the abrupt fault cases, there are no incorrect diagnosis cases and all the undetected samples are during the early stages of the faults when their magnitudes are low. Thus, MDR can be simply worked out as 1 - FDR and are not shown here. The following figures show three sets of diagnosis outputs in incipient faults as examples to show the robustness of the proposed scheme.

Table 4.8 Fault Diagnosis Time in Incipient Faults

Fault No.	**Diagnosis time (s)**					
	Proposed Scheme A			**Baseline Scheme**		
	γ_1	γ_2	γ_3	γ_1	γ_2	γ_3
1	96	52	40	92	64	48
2	80	48	44	88	56	44
3	104	48	40	152	68	52
4	76	48	40	84	76	44
5	116	64	44	152	72	60
6	132	100	80	156	136	88
7	144	96	60	160	100	64
8	88	76	32	96	80	40
9	88	44	44	136	72	48
10	96	68	44	124	80	40
11	128	88	52	144	96	44
Average		72.73			86.55	

Table 4.9 Fault Detection Rate in Incipient Faults

Fault No.	Fault Detection Rate (FDR)					
	Proposed Scheme A			Baseline Scheme		
	γ_1	γ_2	γ_3	γ_1	γ_2	γ_3
1	52%	74%	80%	54%	68%	76%
2	60%	76%	78%	56%	72%	78%
3	48%	76%	80%	24%	66%	74%
4	62%	76%	80%	58%	62%	78%
5	42%	68%	78%	24%	64%	70%
6	34%	50%	60%	22%	32%	56%
7	28%	52%	70%	20%	50%	68%
8	56%	62%	84%	52%	60%	80%
9	56%	78%	78%	32%	64%	76%
10	52%	66%	78%	38%	60%	80%
11	36%	56%	74%	28%	52%	78%
Average		63.6%			56.7%	

Figure 4.11 and Figure 4.12 show the performance of scheme A and the baseline scheme in diagnosing incipient fault No. 3, with the fault developing speed as $\gamma = -1.29 \times 10^{-4}(s^{-1})$. It can be seen from Figure 4.11 that, under scheme A, the network output corresponding to fault No. 3 raises to over 0.8 with some slight oscillations after a period fault developing, while all other network outputs remain close to 0. As shown in Figure 4.12, under the baseline scheme, the network output corresponding to fault No. 3 gradually raises accompanied with some large oscillations until across the diagnosis threshold, then it becomes steady, while all other network outputs remain close to 0. Hence, both schemes successful diagnosed fault No. 3 occurred in this process. Figure 4.11 shows that scheme A successfully diagnosed the fault at 104 s. Figure 4.12 shows that the baseline scheme successfully diagnosed the fault at 152 s.

For this particular incipient fault, scheme A diagnosed the fault 48 s earlier than the baseline scheme, and the output curves of scheme A also stabilize earlier than those in the baseline scheme.

Figure 4.13 and Figure 4.14 show the performance of scheme A and the baseline scheme in diagnosing incipient fault No. 5, with the fault developing speed being $\gamma = 1.12 \times 10^{-4}(s^{-1})$. As shown in Figure 4.13, after the period of fault progressing, the network output corresponding to fault No. 5 raises up and exceeds 0.8 rapidly, while all other network outputs remain close to 0. Hence, scheme A successfully diagnosed the fault at 116 s. As shown in Figure 4.14, the network output corresponding to fault No. 5 gradually increases across the diagnosis threshold 0.8, while all other network outputs remain close to 0. Hence, the baseline scheme successfully diagnosed the fault at 152 s. For this particular incipient fault, the proposed scheme A diagnosed the fault 36 s earlier than the baseline scheme, and the output curves of scheme A also stabilize earlier than those of the baseline scheme.

Figure 4.15 and Figure 4.16 show the performance of scheme A and the baseline scheme in diagnosing incipient fault No. 10, with the fault developing speed being $\gamma = 6.71 \times 10^{-5}(s^{-1})$. It can be seen from two plots that, after an initial period of fault progressing, the network output corresponding to fault No. 10 gradually increases close to 1 with some slight oscillations, while all other network outputs remain lower than 0.2. Figure 4.15 shows that scheme A successfully diagnosed the fault at 96 s. Figure 4.16 shows that the baseline scheme successfully diagnosed the fault at 124 s. It can be seen that the output curve corresponding to fault No. 10 from the baseline scheme is slower in reaching the diagnosis threshold when the abnormal condition occurred. In this case, the proposed scheme A diagnosed the fault 28 s earlier than the baseline scheme.

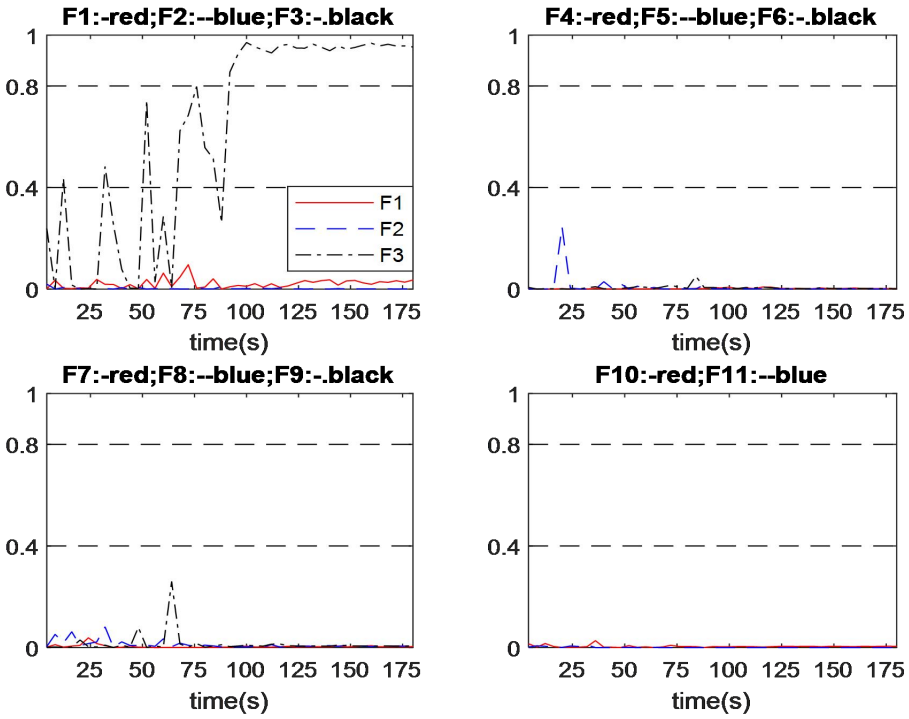

Figure 4.11 Output of fault no.3 diagnosis in incipient fault with scheme A with developing speed of
No.1

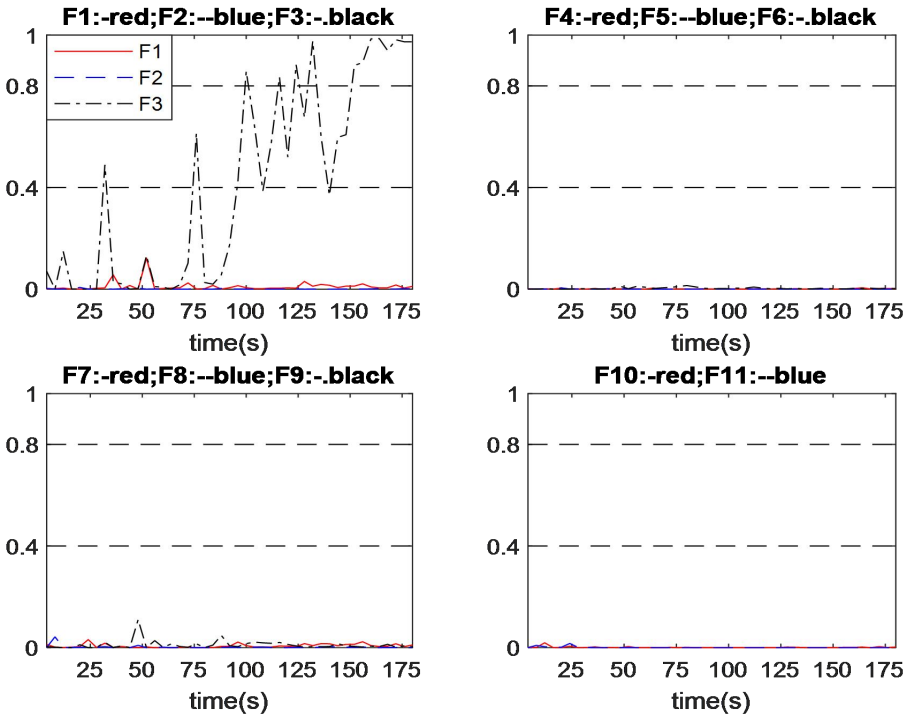

Figure 4.12 Output of fault no.3 diagnosis in incipient fault with the baseline scheme with developing speed of No.1

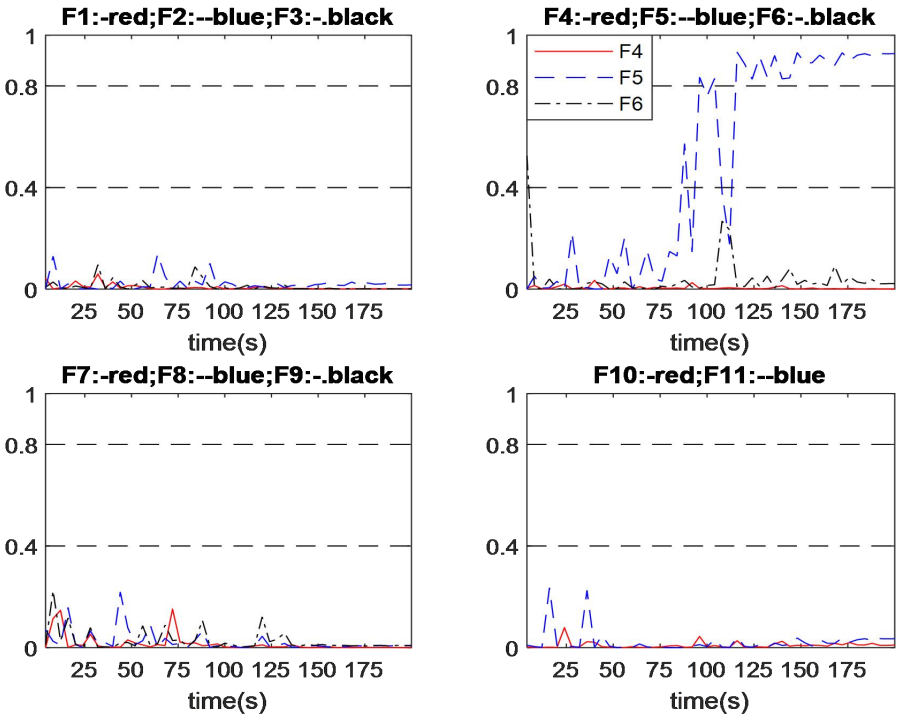

Figure 4.13 Output of fault no.5 diagnosis in incipient fault with scheme A with developing speed of
No.1

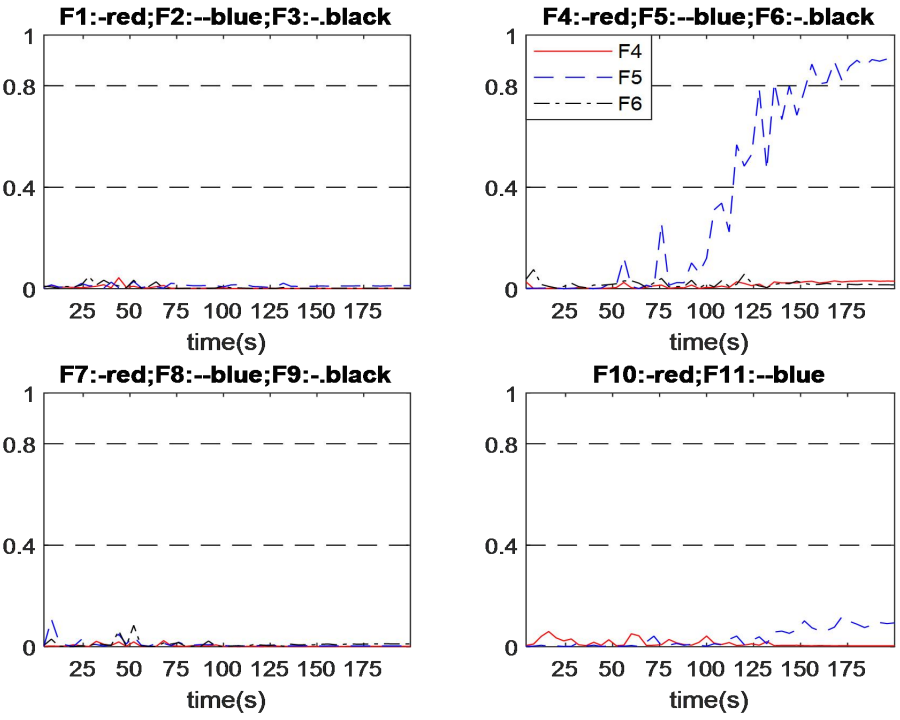

Figure 4.14 Output of fault no.5 diagnosis in incipient fault with the baseline scheme with developing speed of No.1

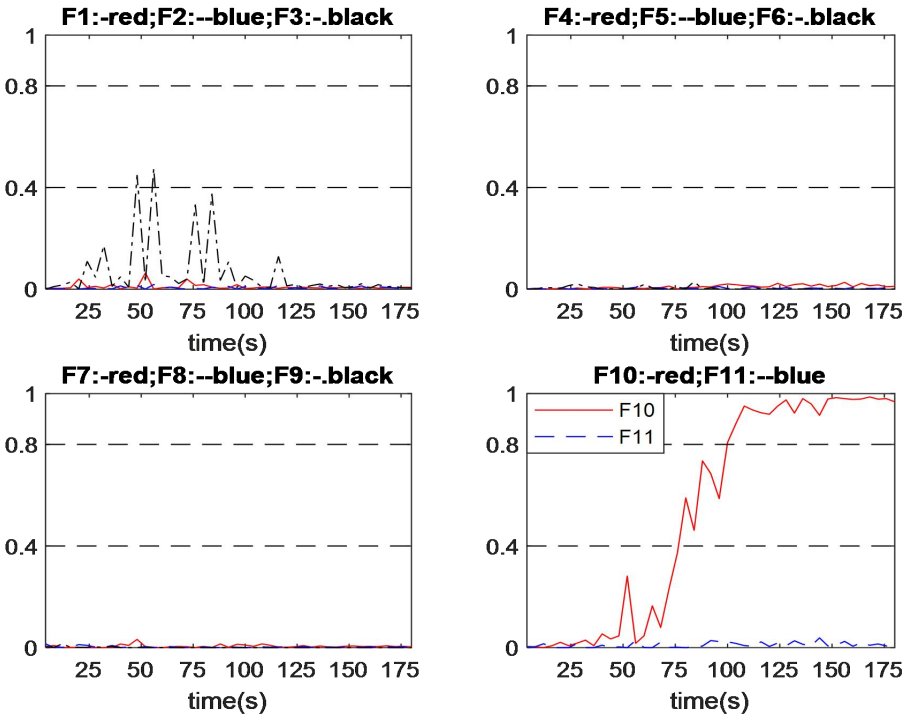

Figure 4.15 Output of fault no.10 diagnosis in incipient fault with scheme A with developing speed of No.1

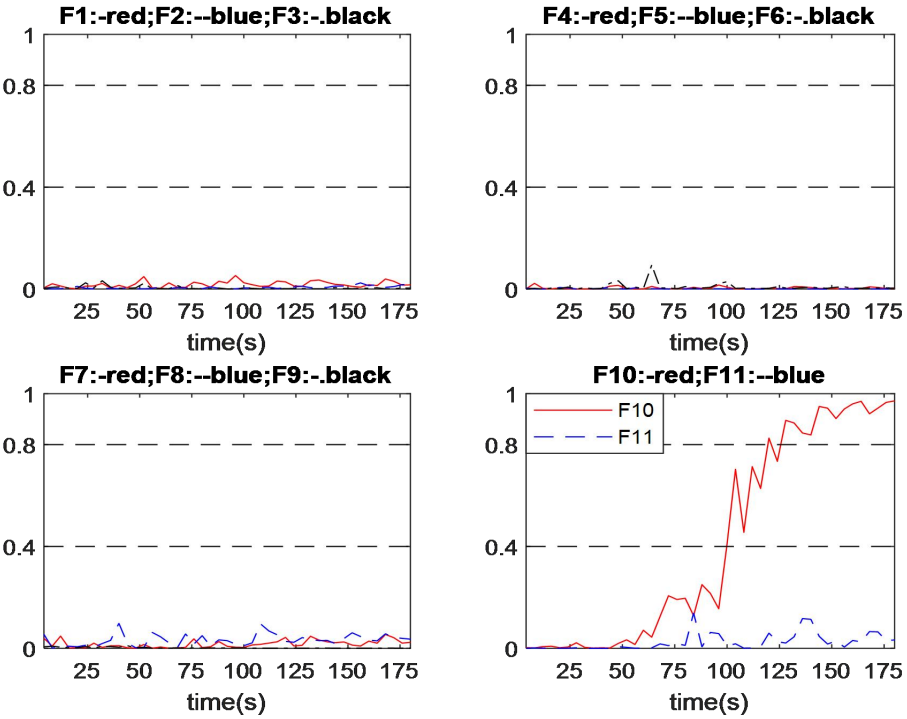

Figure 4.16 Output of fault no.10 diagnosis in incipient fault with the baseline scheme with developing speed of No.1

4.6. Summary

This chapter proposes an enhanced intelligent neural network based online process fault diagnosis system by integrating Andrews plot and neural network techniques. By using features extracted from Andrews plot as the inputs to a neural network, the diagnosis speed and reliability are improved. A method for determining the important features in Andrews function is also proposed.

Applications to a simulated CSTR process give very encouraging results. It is shown that the proposed method can give good performance in terms of diagnosis speed and accuracy. In addition, the proposed data pre-processing method is highly effective in adjusting the high dimensional data to an appropriate size.

As with other neural network based fault diagnosis system, one limitation of the proposed method is that it requires the availability of process data covering various faults. Another limitation of the proposed method is that some uncertainties will inevitably arise when making parameter selections. Time consuming of *t*-value selection can be also a limitation.

In general, the intelligent fault diagnosis system integrating with Andrews plot is successful. Although two limitations in this scheme are hard to eradicate, the encouraging results of application to the CSTR process prove that the proposed diagnosis scheme and the parameter determining method are still feasible.

Chapter 5. Andrews Function based Intelligent Fault Diagnosis System in Combination with Qualitative Trend Analysis

5.1. Introduction

This chapter proposes a neural network based fault diagnosis system developed through the Andrews function and fusion with qualitative trend analysis method. As previously mentioned, unknown uncertainties may occur with parameter selection in Andrews function processing. This chapter attempts integrating Andrews function with the qualitative trend analysis method to obtain better diagnosis performance. The proposed scheme in this chapter is applied to the CSTR system and the diagnosis performance is compared with the baseline scheme for abrupt faults and incipient faults. The classifier in this scheme is the same as the baseline scheme, i.e. a single hidden layer neural network.

The chapter is organised as follows. Section 5.2 gives the proposed fault diagnosis scheme. Section 5.3 gives the development of the proposed fault diagnosis system, and diagnosis results in abrupt faults and incipient faults. Section 5.4 concludes the chapter.

5.2. Proposed Fault Diagnosis System

A fault diagnosis system combining neural network with Andrews function can obtain encouraging performance. But the parameter selection of t values in system development may accompany uncertainties. In order to guard against the uncertainties and improve the system stability, this chapter proposes a fault diagnosis scheme (refer to as scheme B) that additionally combines scheme A with qualitative trend analysis. The establishment of the proposed fault diagnosis system includes the following steps:

Step 1. Data arrangement by using principal components.

Step 2. Information-maintained features are extracted from the principal components by using Andrews function with several selected *t*-values. The method of *t* value selection is same as the previously proposed scheme A.

Step 3. The features from Andrews function are converted into qualitative trend form by using the specified trend analysis equation.

Step 4. The trend features are fed into a neural network to obtain the diagnosis results.

Andrews plot has been suggested several different variations to the functions in past, and this study using the mentioned function (4.1). The *t*-values are selected manually, i.e., generated large numbers of *t*-value distributed uniformly in the range of -π to π and then take several attempts to find the appropriate numbers and values of *t* by observing the intersection of Andrews curves. The numbers of *t*-values involved in the operation determines the final feature dimensions.

As previous mentioned, features extracted by Andrews function can engage more dimensions in the operation, but may also generate uncertainties. The larger separations and more numbers of activated dimension can expect a favourable trend analysis. A limitation in trend analysis is that the unavoidable measurement noises in industrial chemical processes may result in uncertainty in trend analysis. Hence a neural network based classifier uses to deal with the final features.

Neural network is a popular machine learning technique in recent years, especially in the study of high-dimensional data. It can simultaneously handle the multiple independent and dependent variables without the requirement of first principle knowledge on the process. Since the purpose is to verify the superiority of the proposed diagnosis system, both the proposed scheme and the baseline scheme are using the same basic shallow neural network structure, i.e. a neural network with a single hidden layer, as the classifier.

Figure 5.1 shows the framework of the proposed fault diagnosis scheme and the right part in this figure shows each sample $X = (X_1, X_2, X_3, \cdots, X_k)$ being processed in

the Andrews function operation. The monitored process information, X, is replaced by its principal components, $X^{'} = (X'_1, X'_2, X'_3, \cdots, X'_k)$, which, after Andrews function calculation with *m* *t*-values, is converted into the information-contained feature dataset $F_x(t) = \left(F_x(t_1), F_x(t_2), F_x(t_3), \cdots, F_x(t_m)\right)$. Through Andrews function operation, the feature dimension is changed to *m* from *k*.

The feature dataset processed by Andrews function are transformed into qualitative trend as the following:

$$f_x(t)_{i,p} = \begin{cases} 1, & f_x(t)_i - \overline{f_x(t)}_{i,normal} > 3\,\sigma_i \\ -1, & f_x(t)_i - \overline{f_x(t)}_{i,normal} < -3\,\sigma_i \\ 0, & Otherwise \end{cases} \tag{5.1}$$

where $f_x(t)_i$ is the ith Andrews function output, $f_x(t)_{i,p}$ is the qualitative trend value of $f_x(t)_i$, $\overline{f_x(t)}_{i,normal}$ is the mean value of the ith Andrews function output under normal operation condition, σ_i is the standard deviation of the ith Andrews function output under normal operating condition, 1, 0, and −1 represent, respectively, the trend data “increase”, “steady”, and “decrease”. The qualitative trend analysis equation is taken from (Zhang, 2006.). The activation function used in this neural network is sigmoidal function.

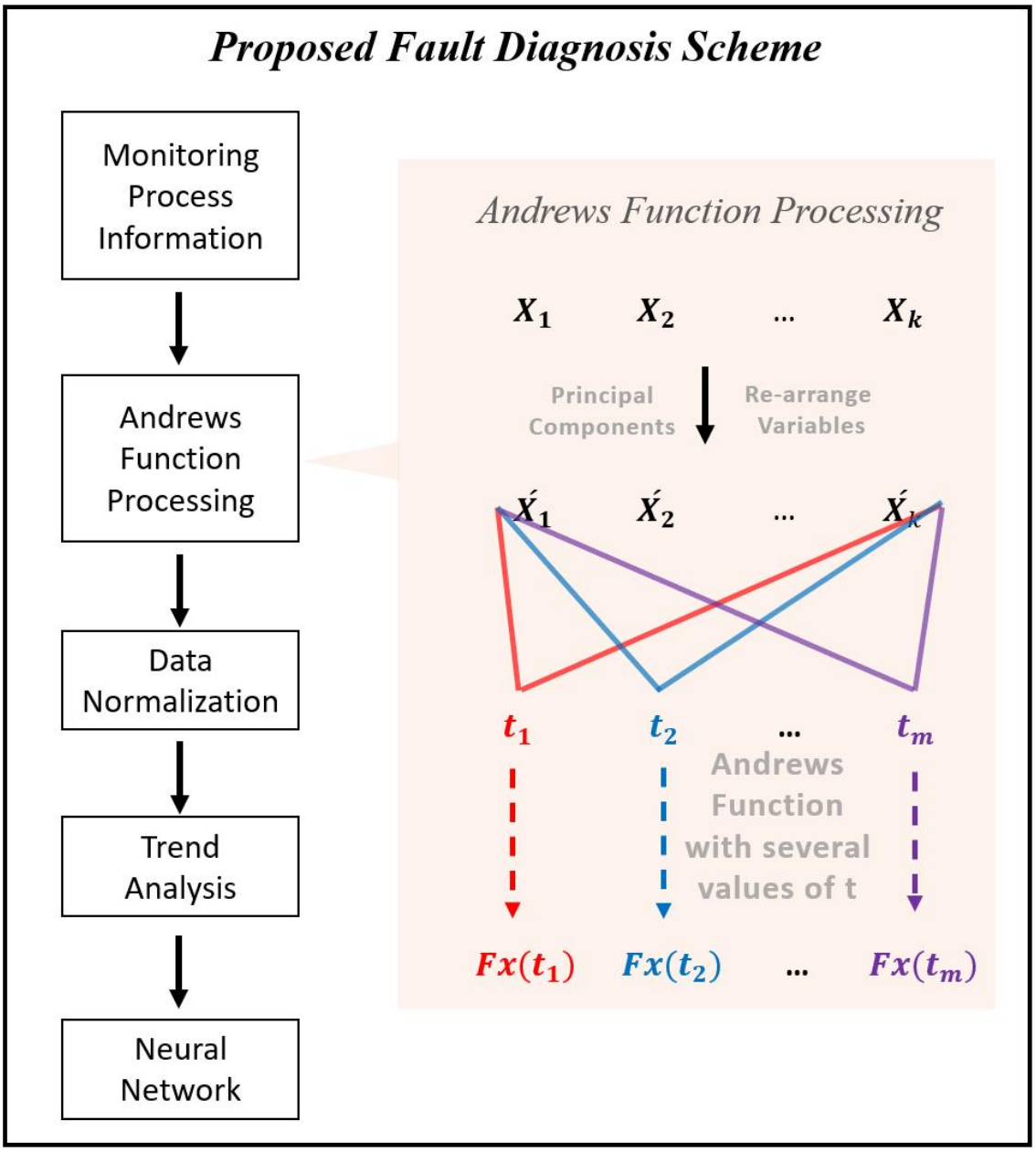

Figure 5.1 Framework of fault diagnosis system

5.3. Fault Diagnosis Results

5.3.1. Development of the Proposed Fault Diagnosis System

The performance of the proposed diagnosis scheme B is compared with the baseline scheme under abrupt faults and incipient faults. The evaluation is based on diagnosis speed and system reliability. The incipient fault is initiated at 0 s, and the abrupt fault is initiated at 40 s. In the proposed scheme B, the selection of *t* values (the

number of features) in Andrews function is important, and the same selection method as scheme A is used. Hence the selected *t* values are [-3.1414, -0.6846, -0.5586, -0.4956, -0.4326, -0.3696, 0.2604, 2.9064, 209694, 3.3024, 3.0954]. The neural network development method is same as the baseline scheme.

Table 5.1 gives the accuracy of different numbers of hidden neurons in the proposed scheme B on the testing data. The best performance is marked with bold font. Hence the number of neurons in the hidden layer is 13. Left of the table gives the neuron numbers in different layer of the neural network. The performance of the proposed scheme B is compared with the baseline scheme in abrupt faults and incipient faults.

Table 5.1 Determination of the Numbers of Hidden Neurons in the Proposed Scheme

Proposed Scheme B							
Determined Neuron Number		**Accuracy of Different Numbers of Hidden Neuron**					
Layer	**Neuron**	Numbers of HN	Accuracy	Numbers of HN	Accuracy	Numbers of HN	Accuracy
Input	**11**	12	91.67%	15	93.33%	18	91.11%
Hidden	**13**	**13**	**98.33%**	16	88.61%	19	93.33%
Output	**11**	14	89.44%	17	96.67%	20	88.61%

5.3.2. Performance under Abrupt Faults

Table 5.2 indicates that both diagnosis systems successfully diagnosed all the considered abrupt faults. In terms of diagnosis speed, the proposed diagnosis scheme B diagnosed the abrupt faults 8.49 s earlier on average than the baseline scheme.

Table 5.2 Fault Diagnosis Time in Abrupt Faults

Fault No.	Diagnosis time (s)					
	Proposed Scheme B			Baseline Scheme		
	Mag. 1	Mag. 2	Mag. 3	Mag. 1	Mag. 2	Mag. 3
1	16	12	4	48	24	12
2	20	8	4	32	12	12
3	32	28	8	44	32	28
4	20	8	8	32	28	8
5	16	8	4	36	24	8
6	24	8	8	12	8	8
7	52	16	16	52	36	16
8	24	8	8	52	20	4
9	20	8	8	40	8	8
10	16	8	8	36	20	4
11	8	8	8	12	8	8
Average	13.69			22.18		

Figure 5.2 shows the outputs from scheme B in diagnosing abrupt fault no.1 with the fault relative magnitude of 1.67%. The output corresponding to fault no.1 from scheme B responds sharply and reaches close 1 at 16 s after the abrupt fault occurred. The output corresponding to fault no.8 also responds quickly after the fault occurred and gives a brief advance warning lasting 12 s and then the warning lifted. All other outputs are close to 0. Hence, the proposed scheme B successfully diagnosed the fault at 16 s after the fault occurred. Figure 5.3 shows the outputs from the baseline scheme in diagnosing abrupt fault no.1 with the fault relative magnitude of 1.67%. The output corresponding to fault no.1 from the baseline scheme responds quickly after the abrupt fault occurred, but the speed of output increasing drops off before reaching the diagnosis threshold (0.8). Then the output rises slowly with irregularly fluctuating and exceeds the diagnosis threshold (0.8) at 48 s. All other outputs are close to 0. Hence, scheme B successfully diagnosed the fault at 48 s after the fault occurred. In this case,

the proposed scheme B diagnosed the fault 32 s earlier than the baseline scheme, and the classifier output curve is smoother and steadier than that of the baseline scheme.

Figure 5.4 shows the outputs from scheme B in diagnosing abrupt fault no.5 with the fault relative magnitude of 9.09%. The output corresponding to fault no.1 from scheme B responds quickly when the abrupt fault occurred. Following a sudden drop, the output curve then rapidly grows close to 1. All other outputs are close to 0. Hence, scheme B successfully diagnosed the fault at 16 s after the fault occurred. Figure 5.5 shows the outputs from the baseline scheme in diagnosing abrupt fault no.5 with the fault relative magnitude of 9.09%. The output corresponding to fault no.5 from the baseline scheme also responds quickly when the abrupt fault occurred. However, the output has oscillations with large amplitude at the advance warning section. After a period of oscillation, the output grows close to 1. All other outputs remain lower than 0.2. Hence, the baseline scheme successfully diagnosed the fault at 36 s after the fault occurred. In this case, the proposed scheme B diagnosed the fault 20 s earlier than the baseline scheme, and the classifier output curve is smoother than that of the baseline scheme.

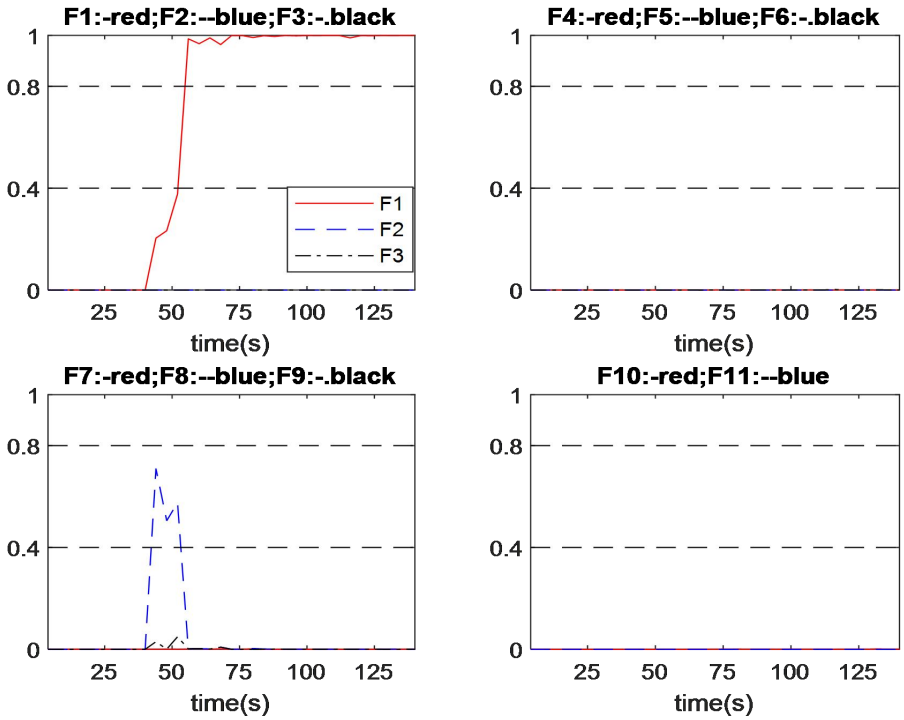

Figure 5.2 Output of fault no.1 diagnosis in abrupt fault with scheme B with relative magnitude of 1.67% (No.1)

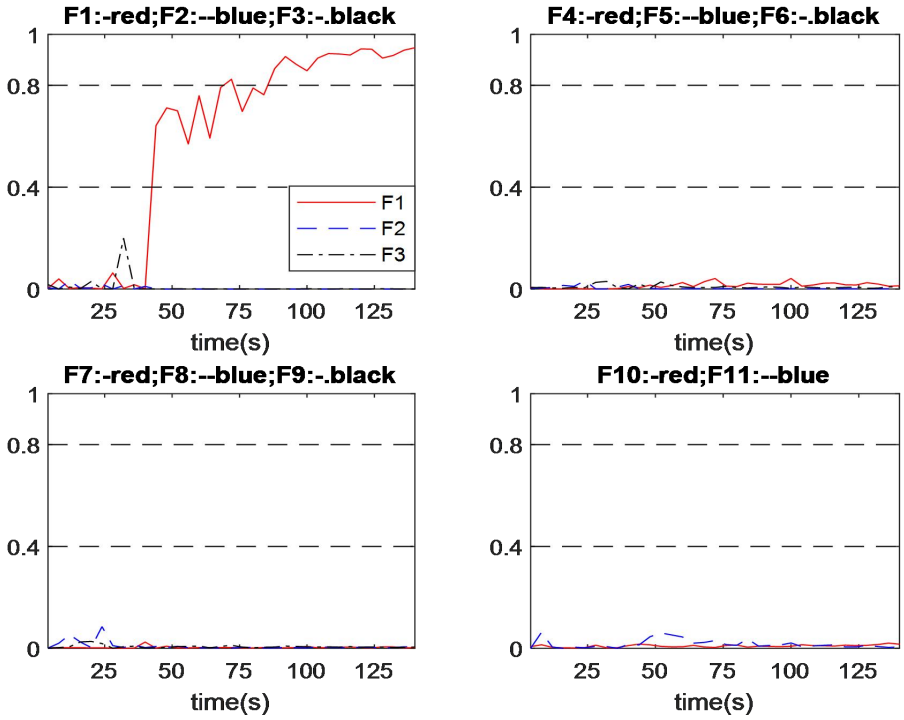

Figure 5.3 Output of fault no.1 diagnosis in abrupt fault with the baseline scheme with relative magnitude of 1.67% (No.1)

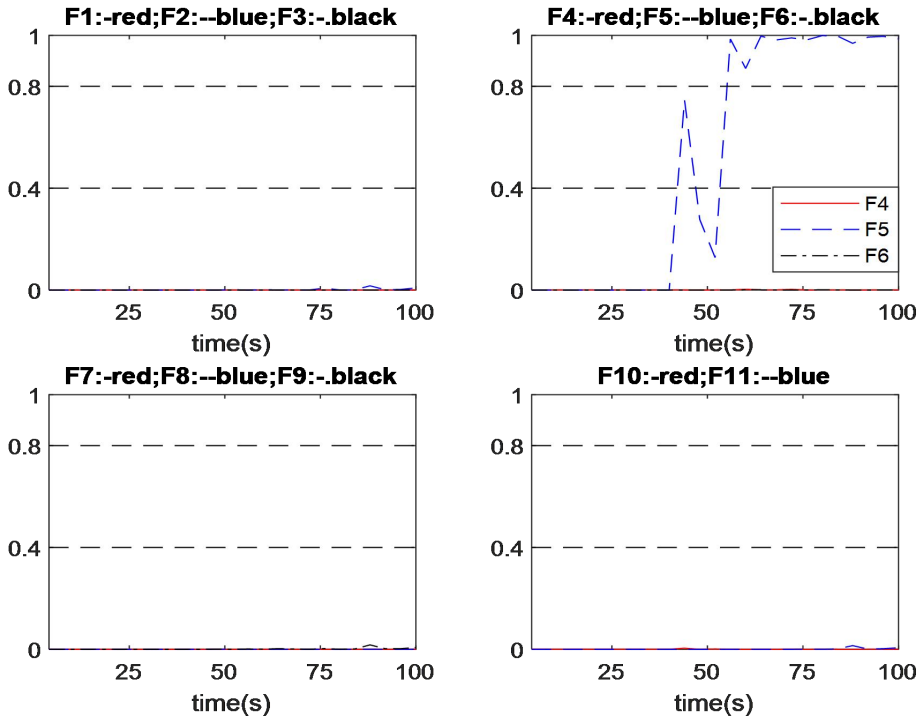

Figure 5.4 Output of fault no.5 diagnosis in abrupt fault with scheme B with relative magnitude of 9.09%
(No.1)

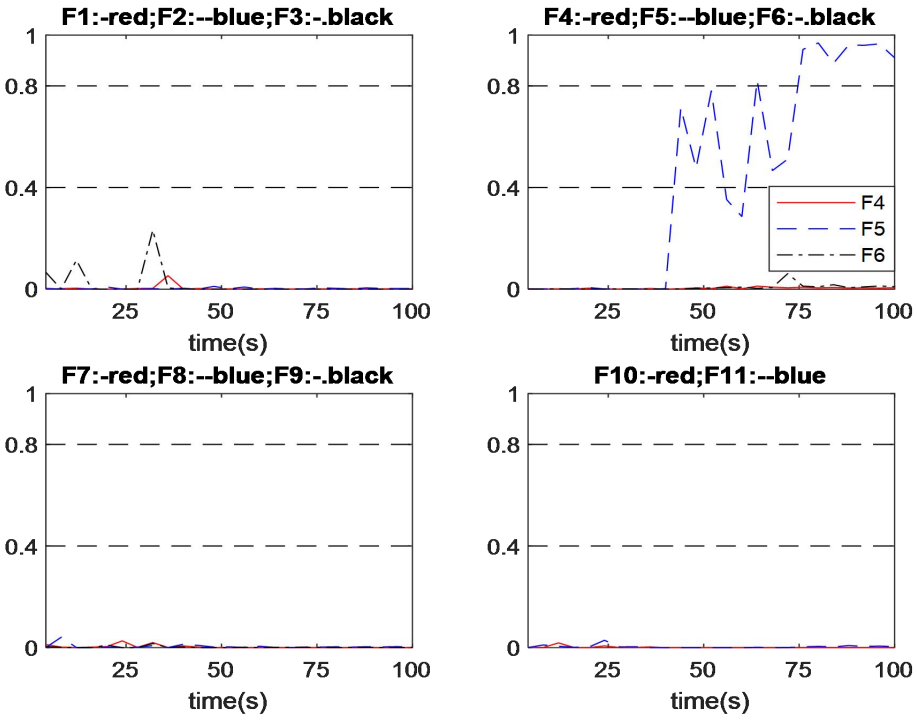

Figure 5.5 Output of fault no.5 diagnosis in abrupt fault with the baseline scheme with relative magnitude of 9.09% (No.1)

5.3.3. Performance under Incipient Faults

Table 5.3 indicates that both diagnosis systems successfully diagnosed all the considered incipient faults. In terms of diagnosis speed, the proposed diagnosis scheme B diagnosed the incipient faults 19.88 s earlier on average than the baseline scheme.

Table 5.3 Fault Diagnosis Time in Incipient Faults

Fault No.	Diagnosis time (s)					
	Proposed Scheme B			Baseline Scheme		
	γ_1	γ_2	γ_3	γ_1	γ_2	γ_3
1	92	52	36	92	64	48
2	64	56	32	88	56	44
3	96	36	32	152	68	52
4	64	40	36	84	76	44
5	120	56	52	152	72	60
6	116	80	68	156	136	88
7	168	88	68	160	100	64
8	76	64	32	96	80	40
9	64	36	28	136	72	48
10	96	56	28	124	80	40
11	132	88	48	144	96	44
Average	66.67			86.55		

Figure 5.6 and Figure 5.7 show, respectively, the performance of scheme B and the baseline scheme in diagnosing incipient fault No. 2, with the fault developing speed $\gamma = 6.67 \times 10^{-5}(s^{-1})$. As shown in Figure 5.6, after a period of damage accumulation, the output corresponding to fault No. 2 from scheme B responds rapidly and reaches close to 1. All other network outputs remain close to 0. Scheme B successfully diagnosed the fault at 64 s. As shown in Figure 5.7, after a period of damage accumulation, the output corresponding to fault No. 2 from the baseline scheme increases with a lower speed with some irregularly fluctuations. The output corresponding to fault No. 3 has low amplitude oscillations, which can be ignored. All

other network outputs remain close to 0. Scheme B successfully diagnosed the fault at 88 s. In this case, scheme B diagnosed the fault 24 s earlier than the baseline scheme, and the outputs is steadier than the baseline scheme.

Figure 5.8 and Figure 5.9 show, respectively, the performance of scheme B and the baseline scheme in diagnosing incipient fault no. 4, with the fault developing speed of $\gamma = -1.67 \times 10^{-4}(s^{-1})$. As shown in Figure 5.8, after a period of damage accumulation, the output corresponding to fault no. 4 from scheme B responds rapidly and reaches close to 1 but next drops close to 0. Then re-jumps close to 1. Some outputs corresponding to fault no. 5 and fault no.11 have different amplitudes can be ignored because they are just some isolated samples. All other network outputs remain close to 0. Scheme B successfully diagnosed the fault at 64 s. As shown in Figure 5.9, after a period of damage accumulation, the output corresponding to fault no. 4 from the baseline scheme raises with more oscillations and the speed is slower. All other network outputs remain lower than 0.2. The baseline scheme successfully diagnosed the fault at 84 s. In this case, scheme B diagnosed the fault 20 s earlier than the baseline scheme, and the classifier output curve is smoother than that of the baseline scheme.

Figure 5.10 and Figure 5.11 show, respectively, the performance of scheme B and the baseline scheme in diagnosing incipient fault no. 8, with the fault developing speed of $\gamma = 7.13 \times 10^{-5}(s^{-1})$. As shown in Figure 5.10, after a period of damage accumulation, the output corresponding to fault no. 8 from scheme B responds rapidly and reaches close to 1. All other network outputs remain close to 0. Scheme B successfully diagnosed the fault at 76 s. As shown in Figure 5.11, after a period of damage accumulation, the output corresponding to fault no. 8 from the baseline scheme slowly raises with more oscillations. All other network outputs remain lower than 0.2. The baseline scheme successfully diagnosed the fault at 96 s. In this case, scheme B diagnosed the fault 20 s earlier than the baseline scheme, and the classifier output curve is smoother than that of the baseline scheme.

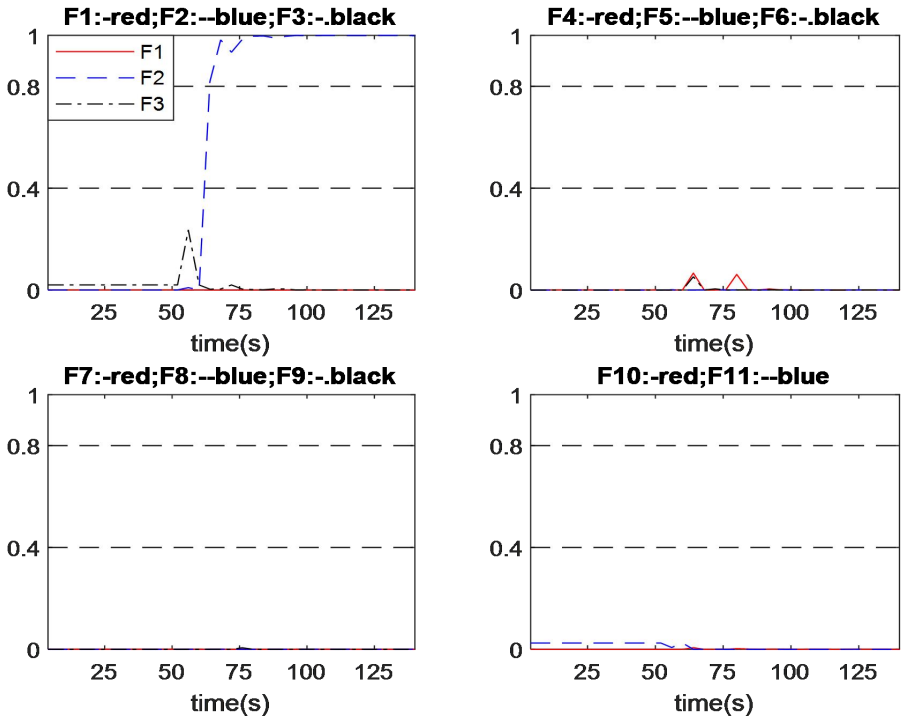

Figure 5.6 Output of fault no.2 diagnosis in incipient fault with scheme B with developing speed of No.1

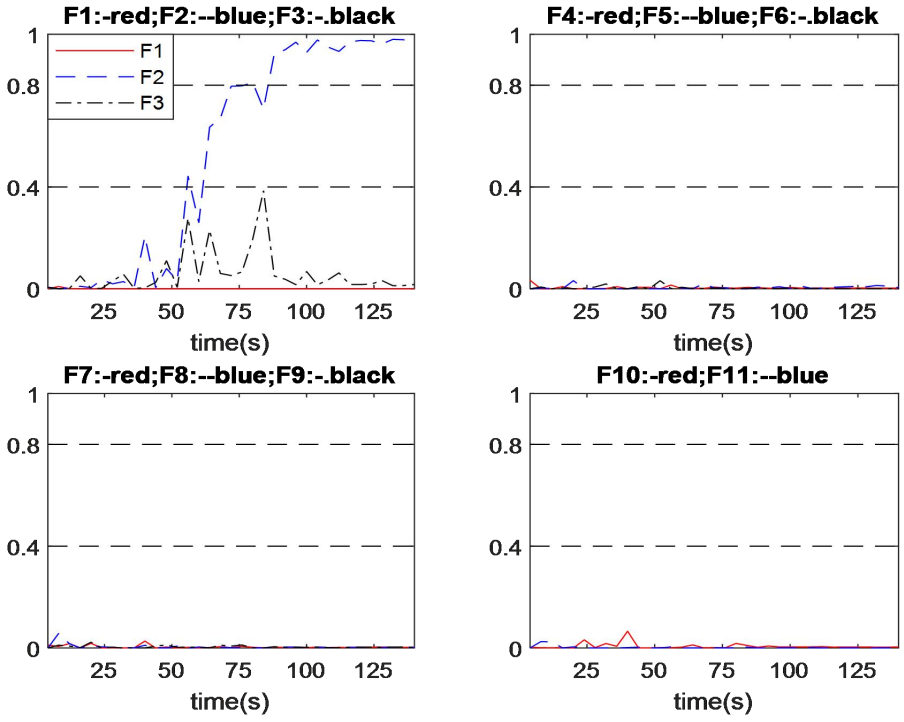

Figure 5.7 Output of fault no.2 diagnosis in incipient fault with the baseline scheme with developing speed of No.1

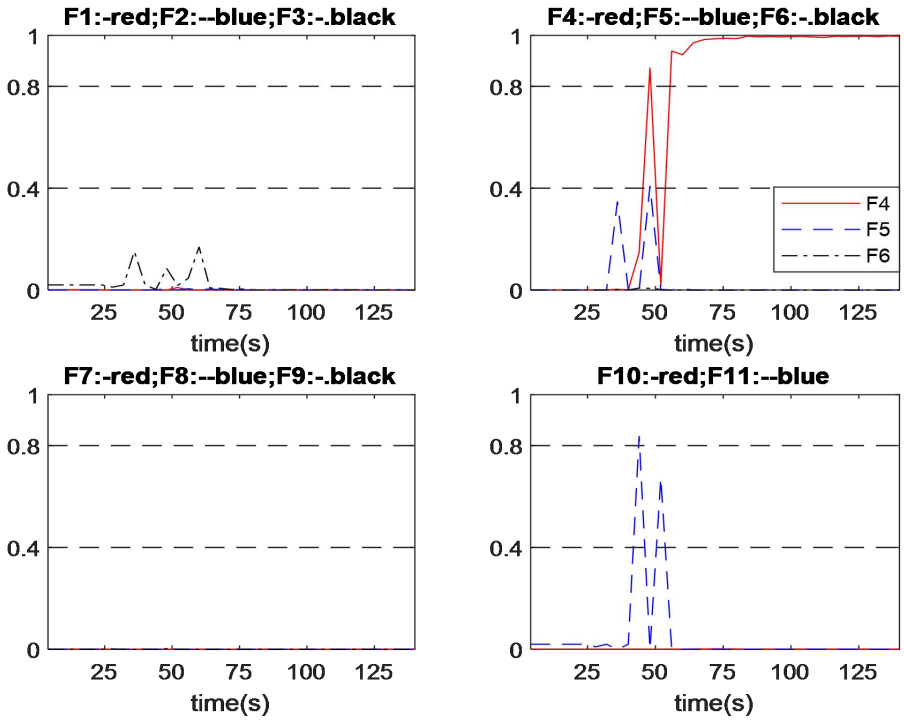

Figure 5.8 Output of fault no.4 diagnosis in incipient fault with scheme B with developing speed of No.1

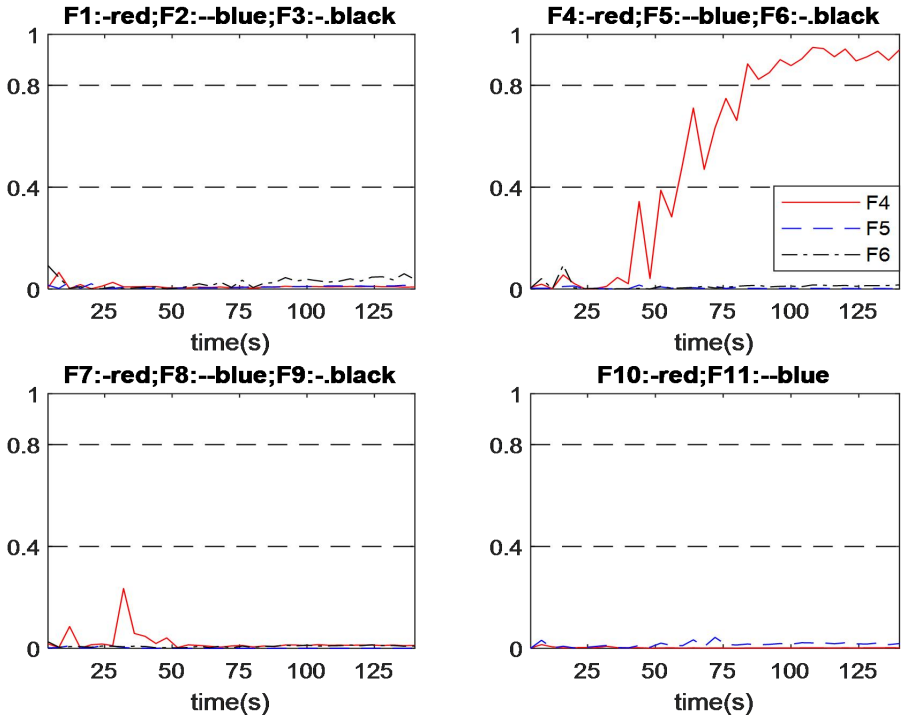

Figure 5.9 Output of fault no.4 diagnosis in incipient fault with the baseline scheme with developing speed of No.1

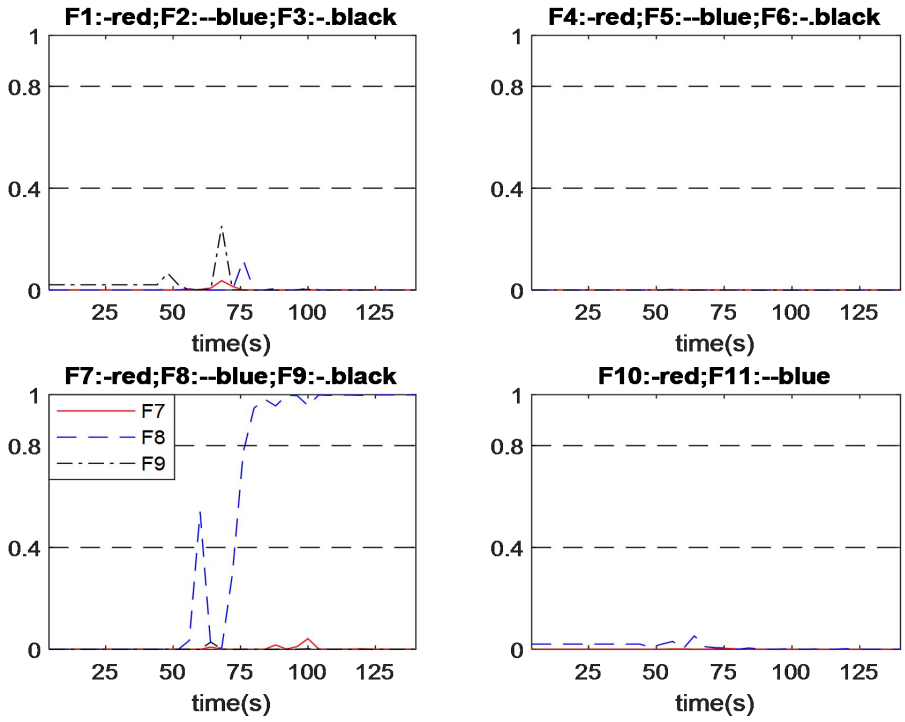

Figure 5.10 Output of fault no.8 diagnosis in incipient fault with scheme B with developing speed of No.1

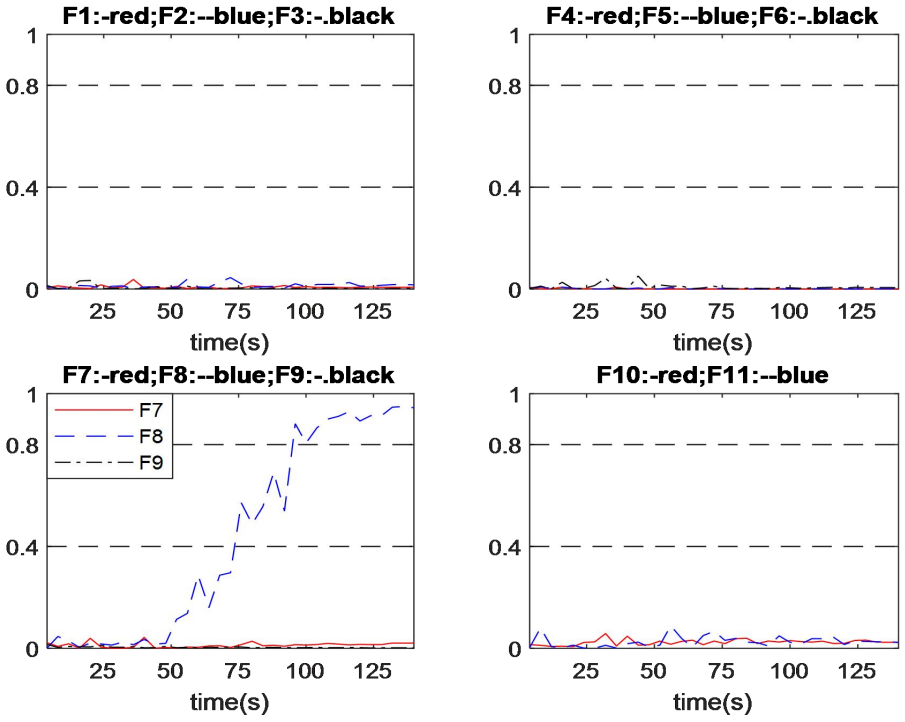

Figure 5.11 Output of fault no.8 diagnosis in incipient fault with the baseline scheme with developing speed of No.1

5.4. Summary

This chapter proposes a neural network based fault diagnosis system developed through the fusion of Andrews function and qualitative trend analysis. The features from the original measurements are extracted using Andrews function with certain numbers of *t*-values. The extracted features are next converted into qualitative trend form, before fed into a neural network.

Applications to a simulated CSTR process proved that the proposed diagnosis scheme can advance the fault diagnosis performance. The feature extracted from Andrews function changed the data distribution. This can make the system more sensitive to varying of parameter corresponding to a particular fault, after converted into qualitative trend form.

Chapter 6. Andrews Function based Intelligent Fault Diagnosis System in Combination with Principal Component Analysis

6.1. Introduction

This chapter proposes a neural network based fault diagnosis system developed through the fusion of Andrews function and principal component analysis (PCA) method. Principal components are calculated using the algorithm of singular value decomposition (SVD). As previously mentioned, unknown uncertainties may occur with parameter selection in Andrews function processing. This work attempts to achieve a faster diagnosis system development, while the diagnosis accuracy is maintained or improved. The proposed scheme C is applied to the simulated CSTR system and compared with the baseline scheme in diagnosing abrupt faults and incipient faults. The classifier in this work is the same as the baseline scheme, i.e., a single hidden layer neural network.

The chapter is organised as follows. Section 6.2 gives the proposed fault diagnosis scheme. Section 6.3 gives the development of the proposed fault diagnosis system, and diagnosis results in abrupt faults and incipient faults. Section 6.4 concludes the chapter.

6.2. Proposed Fault Diagnosis System

The proposed neural network based fault diagnosis system uses Andrews function integrated with PCA to pre-process the monitoring information (refer to as scheme C). In this fault diagnosis scheme, the measured process data is pre-processed by Andrews plot integrated with PCA before fed into a neural network to obtain the diagnosis output. The difference with sheme A is the determination of *t*-values. Manual selection of *t*-values is a time-consuming process and it is hard to find ideal values, due to the relationships in multivariable process information are complex and secretive. Hence, this work uses SVD to find the principal components on the features from Andrews

function processing. In other words, the idea is to utilize PCA based dimension reduction on relatively large number of Andrews function features to achieve a faster data pre-processing. Finally, the principal components of the Andrews function features are used as the inputs to neural network. The automatic selection can save time and improve the data pre-processing performance. The establishment of the proposed fault diagnosis system includes the following steps:

Step 1. Data are re-arranged by using principal components.

Step 2. Information-maintained features are extracted from the principal components by using Andrews function with large numbers, such as 100, of t-values, which are uniformly distributed in the range of $-\pi$ to π.

Step 3. The high-dimensional features are processed using SVD to reduce their dimension.

Step 4. The reduced dimension features are fed into a neural network to obtain the diagnosis results.

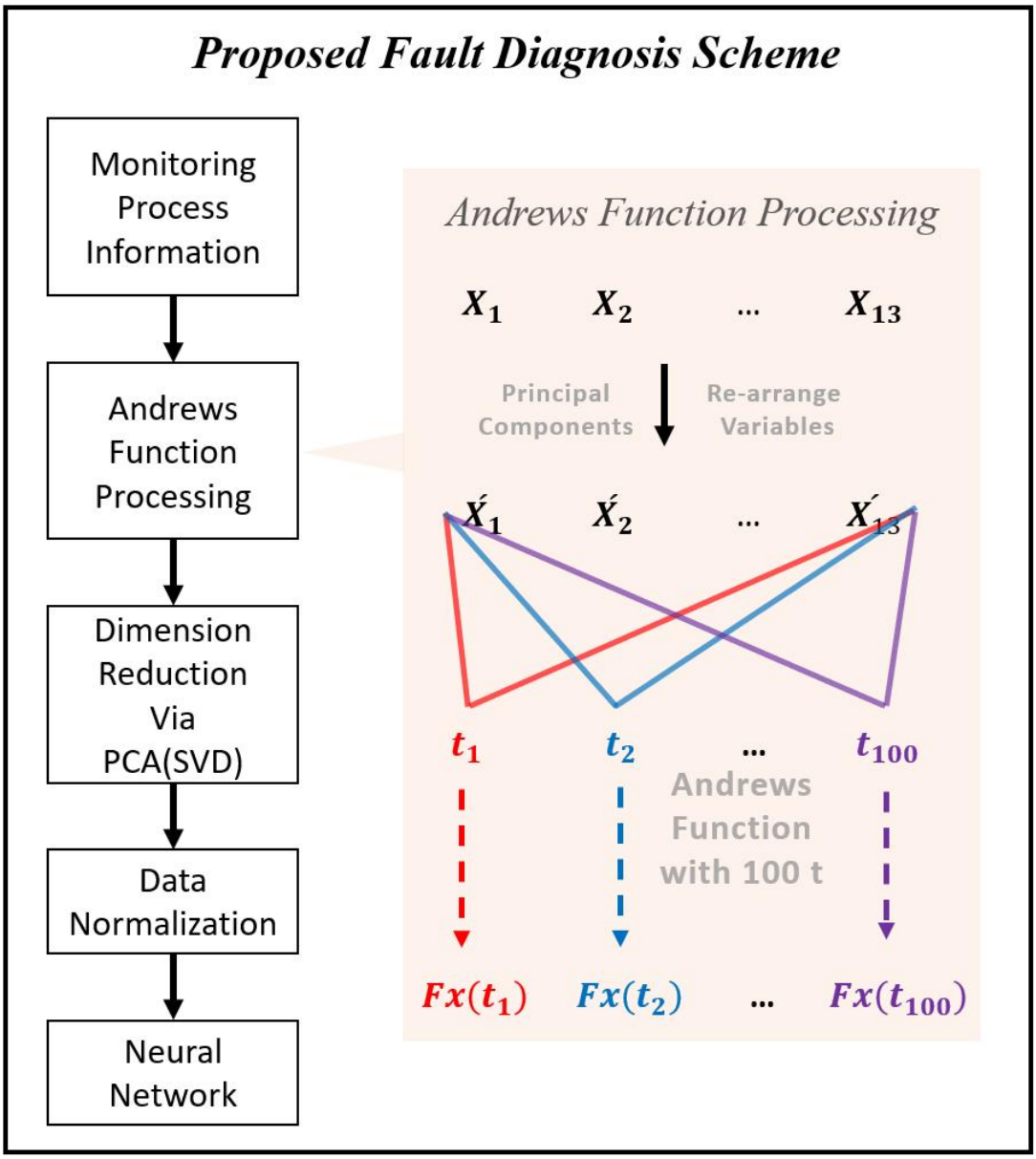

Figure 6.1 Framework of the proposed fault diagnosis scheme

6.3. Fault Diagnosis Results

6.3.1. Development of the Proposed Fault Diagnosis System

The proposed fault diagnosis scheme C firstly uses a large number of *t*-values to obtain a high-dimensional feature dataset. Next an appropriate numbers of principal components of these high-dimensional Andrews function features are used as inputs to a neural network. The neural network development method is the same as the baseline scheme. The important parameter in this scheme is the number of principal components, which determines the dimensions of pre-processed features. The right part of Table 6.1 gives the accuracy on the testing data of different numbers of hidden neurons in the proposed scheme C with different input dimensions. It shows several sets with high accuracy. The best performance is marked with bold font. The left part of the table summarises the neuron numbers in different layer of the neural network. The performance of the proposed scheme C is compared with the baseline scheme in diagnosing abrupt faults and incipient faults.

Table 6.1 Determination of the Numbers of Hidden Neurons and classifier input dimension in the Proposed Scheme

Proposed Scheme C							
Determined Neuron Number		**Accuracy of Different Numbers of Hidden Neuron**					
Layer	**Neuron**	Dimension of Inputs	Numbers of HN	Accuracy	Dimension of Inputs	Numbers of HN	Accuracy
Input	**12**	10	13	93.61%	**12**	**13**	**96.67%**
Hidden	**13**	11	13	95.28%	12	16	93.33%
Output	**11**	11	15	91.67%	12	17	94.72%

6.3.2. Performance under Abrupt Faults

Table 6.2 indicates that both diagnosis systems successfully diagnosed all the considered abrupt faults. In terms of diagnosis speed, the proposed diagnosis scheme C diagnosed the abrupt faults 8.12 s earlier on average than the baseline scheme.

Figure 6.2 shows the outputs from scheme C in diagnosing abrupt fault no.1 with the fault relative magnitude of 1.67%. The output corresponding to fault no.1 from scheme C responds rapidly after the abrupt fault occurred, but it falls downward for a short period and then grows quickly close to 1. All other outputs are close to 0. Scheme C successfully diagnosed the fault at 20 s after the fault occurred. Figure 6.3 shows the outputs from the baseline scheme in diagnosing abrupt fault no.1 with the fault relative magnitude of 1.67%. The output corresponding to fault no.1 from the baseline scheme responds quickly after the abrupt fault occurred, but the speed drops off before reaching the diagnosis threshold (0.8). Then the output rises slowly with irregularly fluctuating and exceeds the diagnosis threshold (0.8). All other outputs are close to 0. The baseline scheme successfully diagnosed the fault at 48 s after the fault occurred. In this case, the proposed scheme C diagnosed the fault 28 s earlier than the baseline scheme.

Figure 6.4 shows the outputs from Scheme A in diagnosing abrupt fault no.8 with the fault relative magnitude of 2.10%. The output corresponding to fault no.8 from scheme C responds quickly when the abrupt fault occurred. Then it grows close to 1 with a slower speed and the classifier output curve has fluctuation with slight amplitude. All other outputs are close to 0. Scheme C successfully diagnosed the fault at 24 s after the fault occurred. Figure 6.5 shows the outputs from the baseline scheme in diagnosing abrupt fault no.8 with the fault relative magnitude of 2.10%. The output corresponding to fault no.8 from the baseline scheme responds slowly with some irregular fluctuations. All other outputs are lower than 0.2. The baseline scheme successfully diagnosed the fault at 52 s after the fault occurred. In this case, scheme C diagnosed the fault 28 s earlier than the baseline scheme, and the classifier output curve is smoother than that of the baseline scheme.

Table 6.2 Fault Diagnosis Time in Abrupt Faults

Fault No.	Diagnosis time (s)					
	Proposed Scheme C			Baseline Scheme		
	Mag. 1	Mag. 2	Mag. 3	Mag. 1	Mag. 2	Mag. 3
1	20	12	4	48	24	12
2	16	12	8	32	12	12
3	24	20	8	44	32	28
4	28	8	4	32	28	8
5	28	16	4	36	24	8
6	32	20	8	12	8	8
7	36	20	12	52	36	16
8	24	4	4	52	20	4
9	20	8	8	40	8	8
10	16	8	4	36	20	4
11	12	8	8	12	8	8
Average	14.06			22.18		

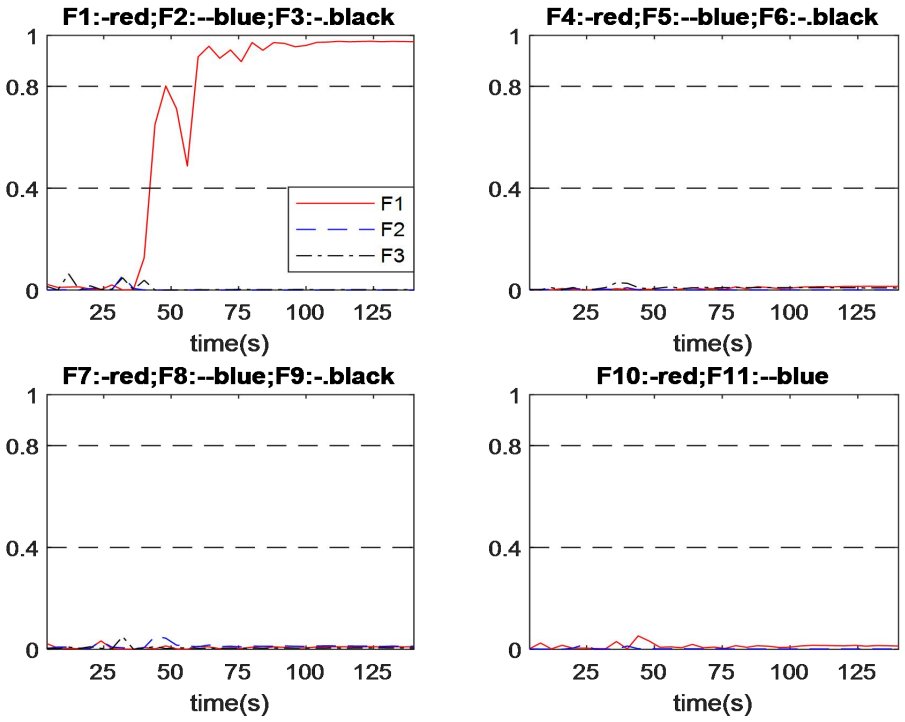

Figure 6.2 Output of fault no.1 diagnosis in abrupt fault with scheme C with relative magnitude of 1.67% (No.1)

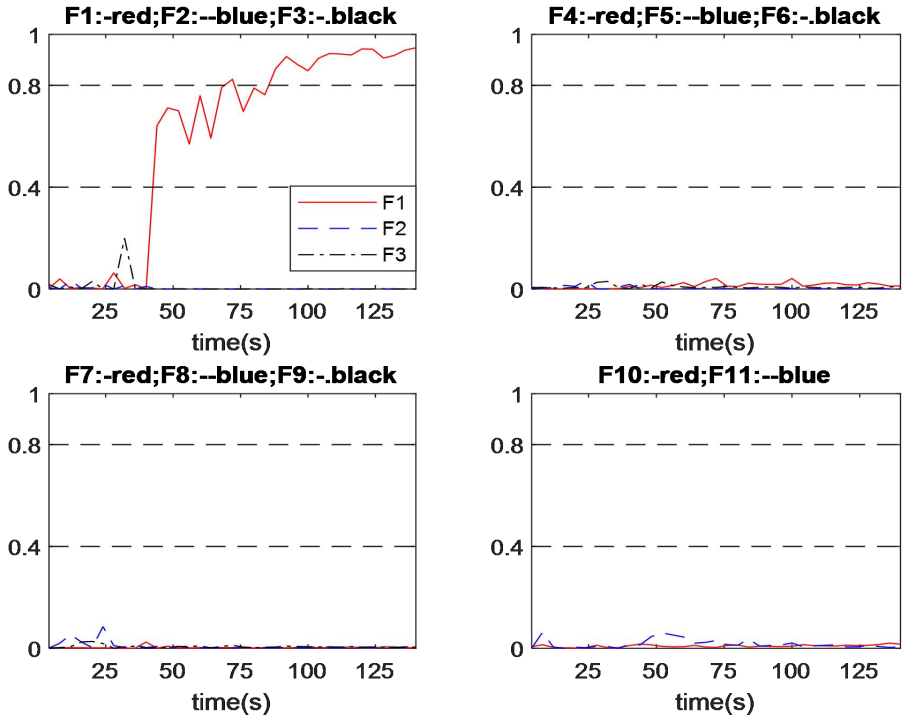

Figure 6.3 Output of fault no.1 diagnosis in abrupt fault with the baseline scheme with relative magnitude of 1.67% (No.1)

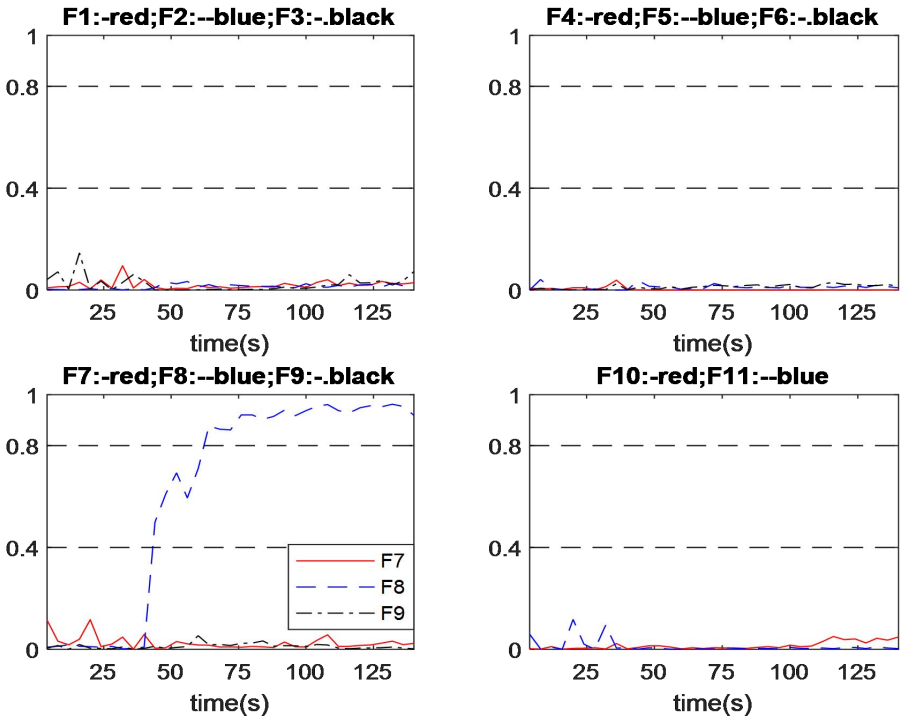

Figure 6.4 Output of fault no.8 diagnosis in abrupt fault with scheme C with relative magnitude of 2.10% (No.1)

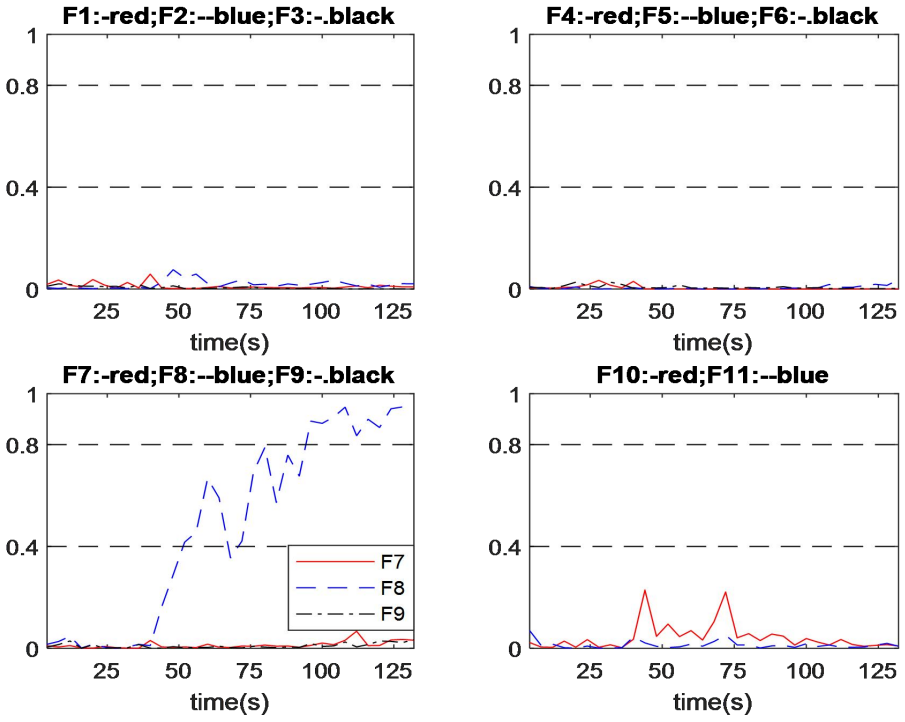

Figure 6.5 Output of fault no.8 diagnosis in abrupt fault with the baseline scheme with relative magnitude of 2.10% (No.1)

6.3.3. Performance under Incipient Faults

Table 6.3 indicates that both diagnosis systems successfully diagnosed all the considered incipient faults. In terms of diagnosis speed, the proposed diagnosis scheme C diagnosed the incipient faults 17.22 s earlier on average than the baseline scheme.

Table 6.3 Fault Diagnosis Time in Incipient Faults

Fault No.	Diagnosis time (s)					
	Proposed Scheme C			Baseline Scheme		
	γ_1	γ_2	γ_3	γ_1	γ_2	γ_3
1	88	56	36	92	64	48
2	76	52	32	88	56	44
3	92	48	36	152	68	52
4	68	44	36	84	76	44
5	120	68	52	152	72	60
6	116	84	88	156	136	88
7	136	88	60	160	100	64
8	68	60	24	96	80	40
9	96	60	48	136	72	48
10	92	72	44	124	80	40
11	128	76	44	144	96	44
Average		69.33			86.55	

Figure 6.6 and Figure 6.7 show, respectively, the performance of scheme C and the baseline scheme in diagnosing incipient fault no. 3, with the fault developing speed of $\gamma = -1.29 \times 10^{-4}(s^{-1})$. As shown in Figure 6.6, after a period of damage accumulation, the output corresponding to fault No. 3 from scheme C responds with large amplitude irregular fluctuations until reaching close to 1. All other network outputs remain close to 0. Scheme C successfully diagnosed the fault at 92 s. As shown in Figure 6.7 after a period of damage accumulation, the output corresponding to fault No. 3 from the baseline scheme responds with large amplitude irregular fluctuations

until reaching close to 1. The responding time is later and the fluctuation time is longer than scheme C. All other network outputs remain close to 0. The baseline scheme successfully diagnosed the fault at 152 s. In this case, scheme C diagnosed the fault 60 s earlier than the baseline scheme, and the classifier output curve from scheme C is comparatively steadier than that from the baseline scheme.

Figure 6.8 and Figure 6.9 show, respectively, the performance of scheme C and the baseline scheme in diagnosing incipient fault no. 6, with the fault developing speed of $\gamma = 6.67 \times 10^{-4}(s^{-1})$. As shown in Figure 6.8, after a period of damage accumulation, the output corresponding to fault no. 6 from scheme C increases with fluctuations. The output stays at the section of advance warning for a period of time with fluctuations and then rapidly increases to close to 1. Ignoring the isolated samples, all other network outputs are close to 0. Scheme C successfully diagnosed the fault at 116 s. As shown in Figure 6.9, after a period of damage accumulation, the output corresponding to fault no. 6 from the baseline scheme has a responding trend similar to that from scheme C but with a slower speed. All other network outputs are close to 0. The baseline scheme successfully diagnosed the fault at 156 s. In this case, the scheme C diagnosed the fault 40 s earlier than the baseline scheme.

Figure 6.10 and Figure 6.11 show, respectively, the performance of scheme C and the baseline scheme in diagnosing incipient fault no. 8, with the fault developing speed of $\gamma = 7.13 \times 10^{-5}(s^{-1})$. As shown in Figure 6.10, after a period of damage accumulation, the output corresponding to fault no. 8 from scheme C responds with a slow speed. Then the output grows rapidly close to 1. Ignoring the isolated samples, all other network outputs are close to 0. Scheme C successfully diagnosed the fault at 68 s. As shown in Figure 6.11, after a period of damage accumulation, the output corresponding to fault no. 8 in the baseline scheme increases with a comparatively slower speed. All other network outputs are close to 0. The baseline scheme successfully diagnosed the fault at 96 s. In this case, scheme C diagnosed the fault 28 s earlier than the baseline scheme.

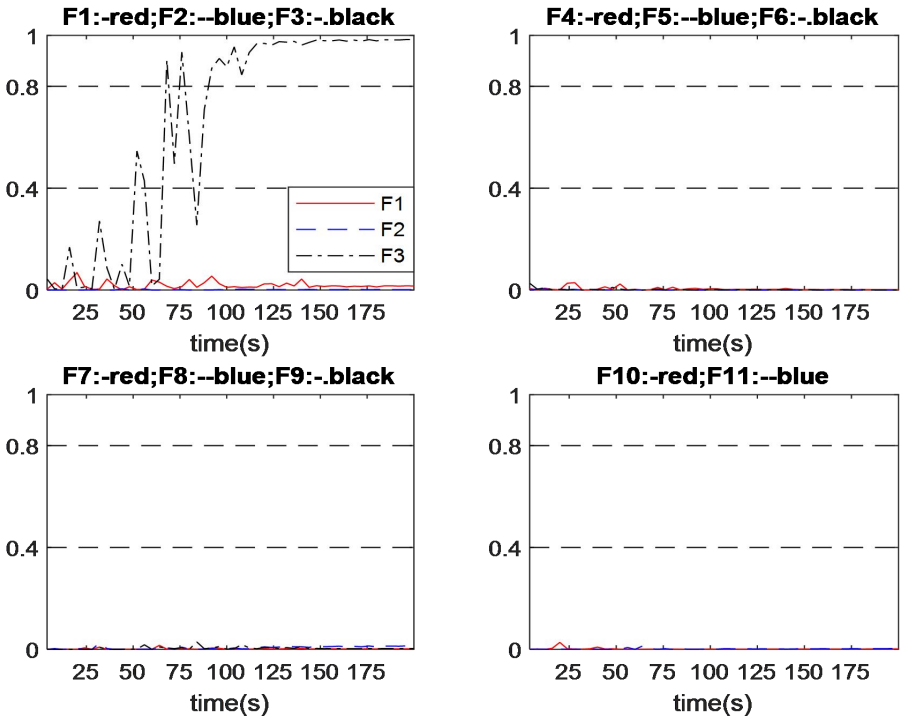

Figure 6.6 Output of fault no.3 diagnosis in incipient fault with scheme C with developing speed of No.1

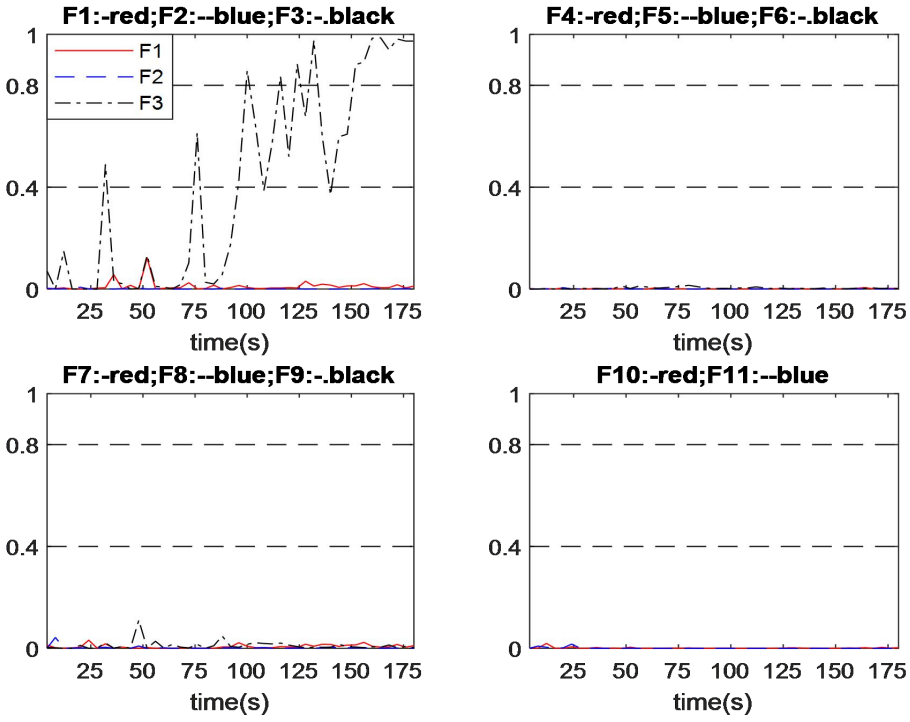

Figure 6.7 Output of fault no.3 diagnosis in incipient fault with the baseline scheme with developing speed of No.1

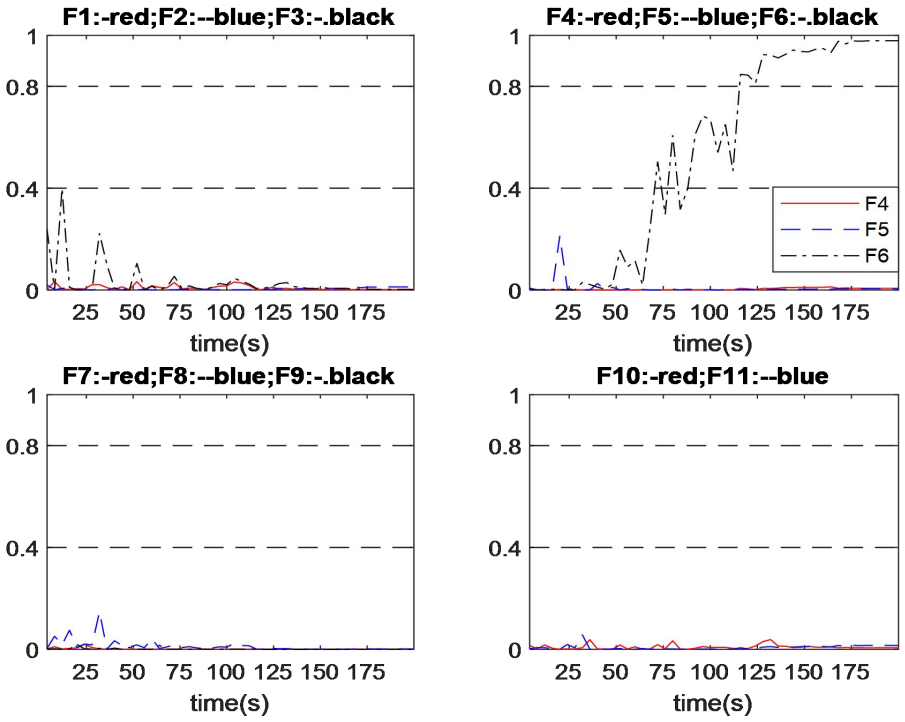

Figure 6.8 Output of fault no.6 diagnosis in incipient fault with scheme C with developing speed of No.1

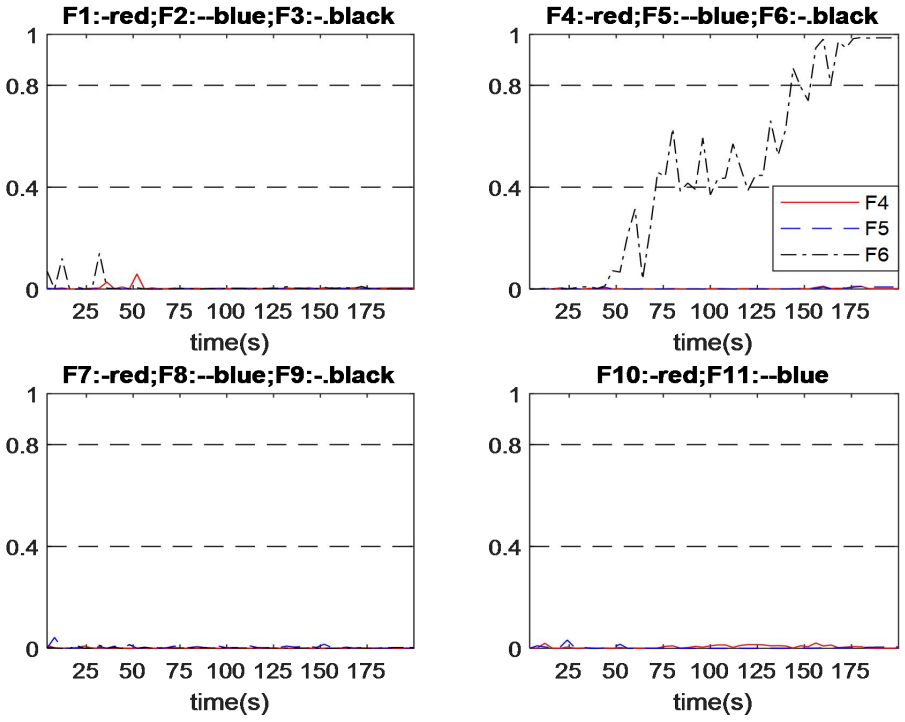

Figure 6.9 Output of fault no.6 diagnosis in incipient fault with the baseline scheme with developing speed of No.1

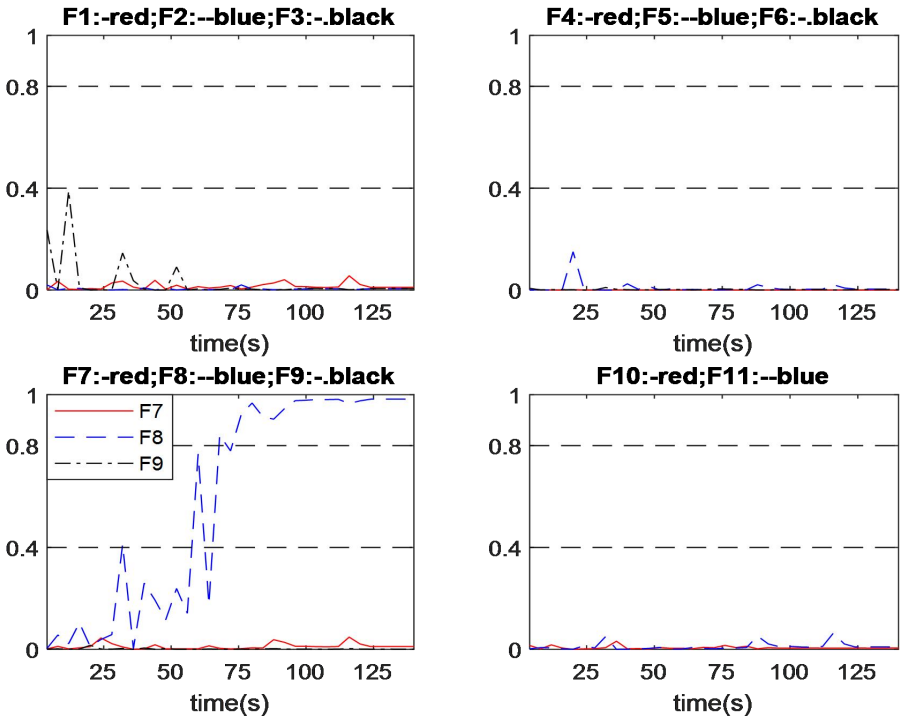

Figure 6.10 Output of fault no.8 diagnosis in incipient fault with scheme C with developing speed of No.1

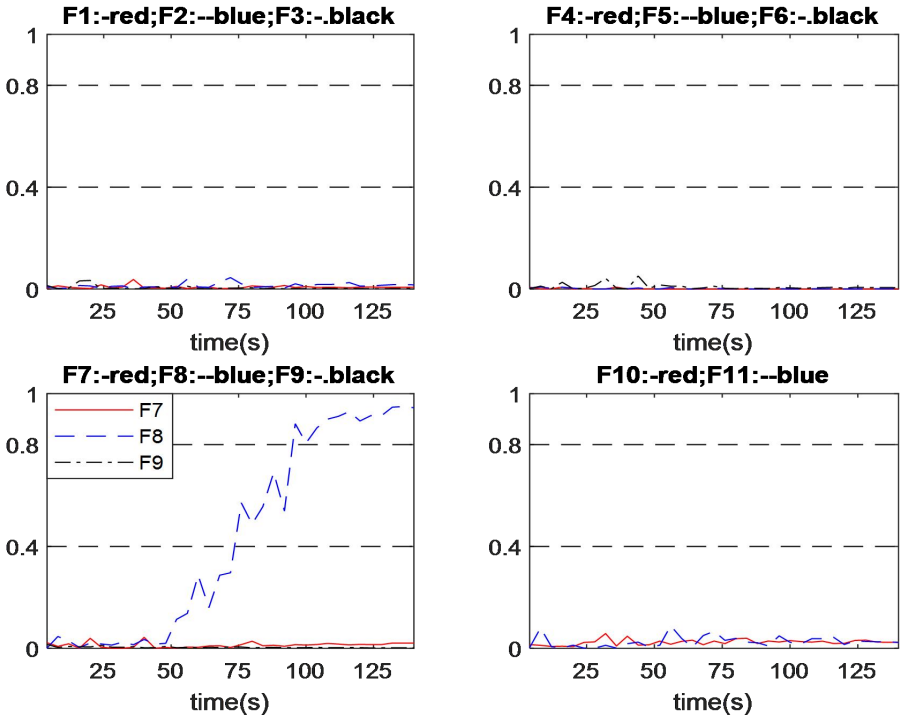

Figure 6.11 Output of fault no.8 diagnosis in incipient fault with the baseline scheme with developing speed of No.1

6.4. Summary

This chapter proposes a neural network based fault diagnosis system developed through the fusion of Andrews function and dimension reduction by principal component analysis. The information-maintained features in the Andrews function with large numbers of *t*-values are extracted by using principal component analysis. Then the principal components are fed into a neural network.

Compared with the baseline conventional neural network based fault diagnosis system, as well as scheme A presented in Chapter 4, the proposed fault diagnosis system in this chapter gives better diagnosis performance on the CSTR system. System development integrating PCA to reduce the dimension of Andrews function features can save a lot of time in determining the appropriate number of features in Andrews function, and also utilizing the final pre-processed features can achieve better fault diagnosis performance.

Uncertainties in industrial processes ,under the existing technology, are unobservable, ubiquitous, vary over time, and may incubate new uncertainties from the selection of parameter. The scheme used in this chapter tried to weaken the impact from uncertainties. According to the application results, fault diagnosis system using this scheme successfully weakens the impacts of uncertainties and significantly reduced the development time of diagnosis system.

In general, the intelligent based fault diagnosis system integrating Andrews plot and PCA is successful. Aiming at overcoming the uncertainties caused by parameter selection in scheme A, PCA based dimension reduction technique is used for parameter selection and this has achieved good results and greatly reduced the system development time. This indicates that the dimension reduction technique has a great potential to overcome the impacts of uncertainties from parameter selection in Andrews function processing.

Chapter 7. Andrews Function based Intelligent Fault Diagnosis System Combined with Autoencoder

7.1. Introduction

This chapter proposes a neural network based fault diagnosis system developed through the fusion of Andrews function and autoencoder. The core idea is use feature dimensionality reduction instead of manual selection of features, similar as scheme C proposed in Chapter 6. However, the PCA in scheme C is a linear dimension reduction technique which may face problems when the features are nonlinearly related. An autoencoder can be utilized for nonlinear dimension reduction based on neural network. As previously mentioned, unknown uncertainties may occur with parameter selection in Andrews function processing. This work attempts to achieve a faster diagnosis system development, while the accurate diagnosis is maintained or improved. The proposed diagnosis scheme is applied to the CSTR system and the diagnosis performance is compared with the baseline scheme in abrupt faults and incipient faults. The classifier in this work is the same as the baseline scheme, i.e. a single hidden layer neural network.

The chapter is organised as follows. Section 7.2 gives the proposed fault diagnosis scheme. Section 7.3 gives the development of the proposed fault diagnosis system, and diagnosis results in abrupt faults and incipient faults. Section 7.4 concludes the chapter.

7.2. Proposed Fault Diagnosis System

The idea in this proposed diagnosis scheme (refer to as scheme D) is similar to scheme C proposed in the previous chapter. In this proposed scheme, Andrews function with a large number of *t*-values is used and the encoder from a trained autoencoder is used to implement dimension reduction on the large number of Andrews function features. This method can save time and reduce uncertainties on

t-value selection. The establishment of the proposed fault diagnosis system includes the following steps:

Step 1. Data are re-arranged by using all principal components.

Step 2. Information containing features are extracted from the principal components by using Andrews function with large numbers of selected *t*-values.

Step 3. The dimension of the extracted features is reduced through the encoder from a trained autoencoder.

Step 4. The resulting features are normalized and then fed into a single neural network to obtain the diagnosis outputs.

This work uses the Andrews function (4.1) with a large number of *t*-values uniformly distributed in the range from -π to π. The dimension of features extracted from principal components is equal to the number of *t*-values. In contrast with manual selection of *t*-values, features pre-processed after dimensionality reduction via encoder can be expected to subside uncertainties. Also using the encoder can reduce the computation time of *t*-value selection. Figure 7.1 shown the framework of the proposed scheme. Figure 7.2 shows the diagnosis process from a sampling vector $x = [x_1, x_2, ..., x_{var}]$ being fed into the proposed scheme to the output of diagnosis.

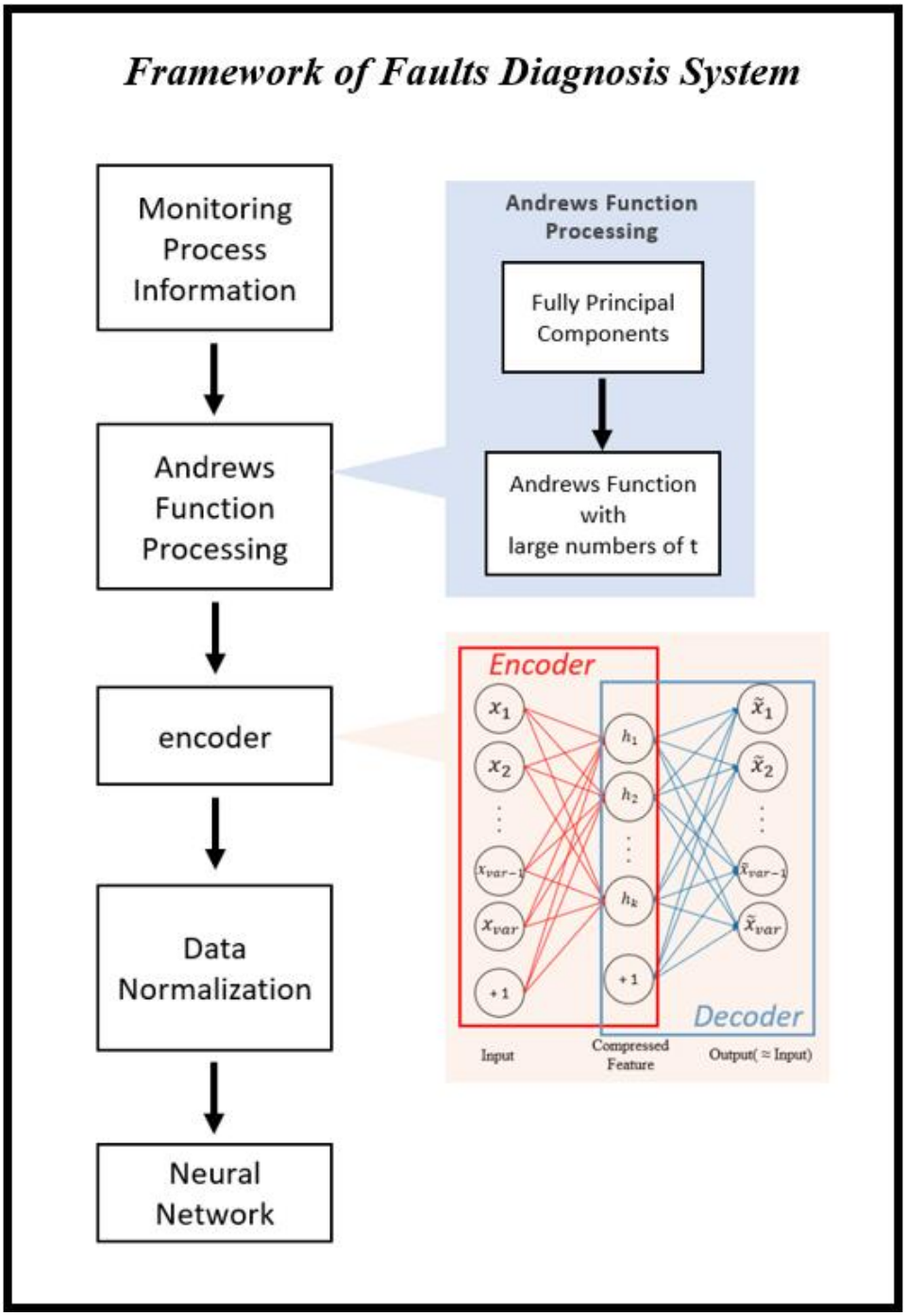

Figure 7.1 Framework of the proposed diagnosis scheme

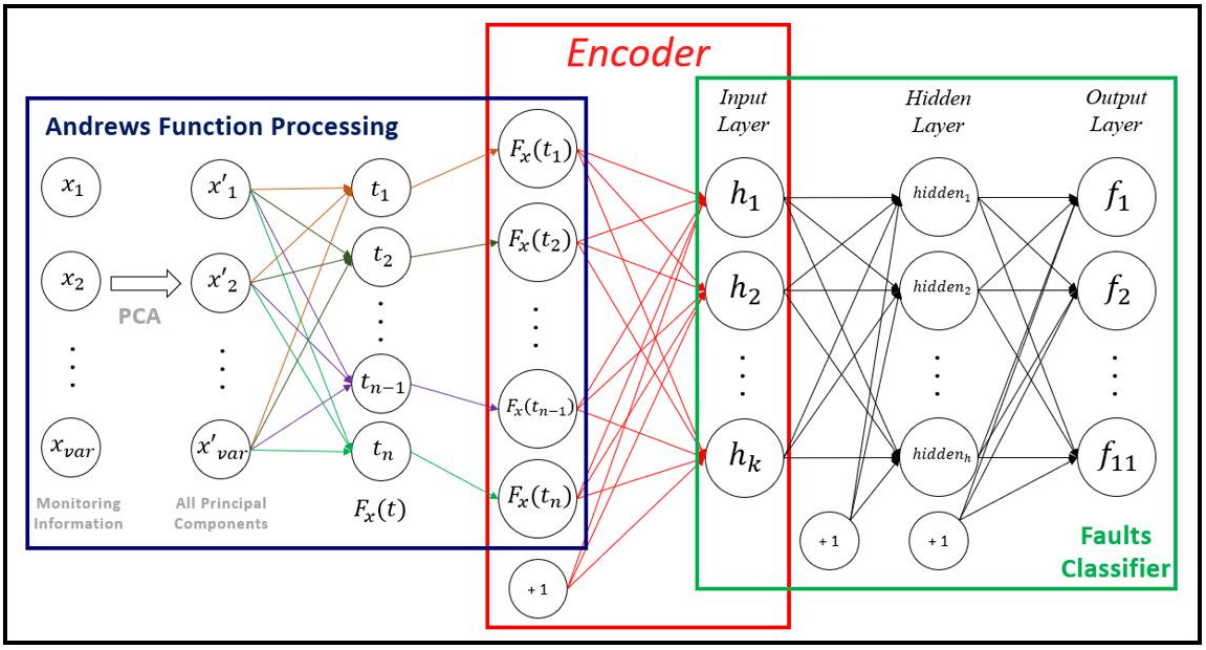

Figure 7.2 Diagnosis process of the proposed scheme

7.3. Fault Diagnosis Results

7.3.1. Development of the Proposed Fault Diagnosis System

The proposed fault diagnosis scheme D firstly uses a large number (63) of *t*-values to obtain a high-dimensional feature dataset. These Andrews function features are compressed using the encoder of a trained autoencoder and these compressed features are used as inputs to the classifier. The neural network development method is the same as the baseline scheme. The important parameter in this scheme is the neuron number of the encoder, which determines the dimensions of pre-processed feature. Table 7.1 gives the accuracy on the testing data of different numbers of hidden neurons in the proposed scheme D with different input dimensions. The best performance is marked with bold font. Hence, the neuron numbers in input layer, hidden layer and output layer of the neural network are 11, 18, and 11, respectively. The performance of the proposed scheme D is compared with the baseline scheme in abrupt faults and incipient faults.

Table 7.1 Accuracy of Different Numbers of Hidden Neurons and Inputs in Scheme D

Accuracy		**Numbers of Hidden Neurons**					
		13	15	17	**18**	19	21
Dimension of Inputs (Encoder Size)	10	93.89%	87.78%	96.67%	96.67%	96.67%	90.83%
	11	93.33%	93.33%	96.67%	**98.33%**	97.78%	91.39%
	12	96.67%	96.11%	96.67%	96.67%	97.22%	91.67%
	13	93.33%	90.56%	96.67%	96.67%	96.94%	90.83%
	15	91.67%	92.78%	95.28%	94.72%	96.67%	83.61%
	18	86.39%	93.33%	92.22%	91.67%	93.06%	96.67%

7.3.2. Performance under Abrupt Faults

Table 7.2 indicates that both diagnosis systems successfully diagnosed all the considered abrupt faults. In terms of diagnosis speed, the proposed diagnosis scheme D diagnosed the abrupt faults 8.85 s earlier on average than the baseline scheme.

Figure 7.3 shows the outputs from scheme D in diagnosing abrupt fault no.7 with the fault relative magnitude of 16.76%. The output corresponding to fault no.7 from scheme D responds rapidly after the abrupt fault occurred. The output stays at section of advance warning for a period time with oscillations, and then rapidly increases to close to 1. All other outputs are close to 0. The proposed scheme D successfully diagnosed the fault at 36 s after the fault occurred. Figure 7.4 shows the outputs from the baseline scheme in diagnosing abrupt fault no.7 with the fault relative magnitude of 16.76%. The output corresponding to fault no.1 from the baseline scheme initially responds faster than that in scheme D after the abrupt fault occurred, but stays for a longer time at the section of advance warning, and the oscillations have larger amplitude. All other outputs are close to 0. The baseline scheme successfully

diagnosed the fault at 52 s after fault occurred. In this case, the proposed scheme D diagnosed the fault 16 s earlier than the baseline scheme.

Figure 7.5 shows the outputs from scheme D in diagnosing abrupt fault no.10 with the fault relative magnitude of 2.54%. The output corresponding to fault no.10 from scheme D responds quickly when the abrupt fault occurred. It grows fast and soon reaches the diagnosis threshold (0.8). All other outputs are close to 0. The proposed scheme D successfully diagnosed the fault at 16 s after the fault occurred. Figure 7.6 shows the outputs from the baseline scheme in diagnosing abrupt fault no.10 with the fault relative magnitude of 2.54%. The output corresponding to fault no.10 from the baseline scheme responds with a slower speed and some irregular fluctuations. All other outputs remain close to 0. The baseline scheme successfully diagnosed the fault at 36 s after the fault occurred. In this case, the proposed scheme D diagnosed the fault 20 s earlier than the baseline scheme.

Table 7.2 Fault Diagnosis Time in Abrupt Faults

Fault No.	**Diagnosis time (s)**					
	Proposed Scheme D			**Baseline Scheme**		
	Mag. 1	**Mag. 2**	**Mag. 3**	**Mag. 1**	**Mag. 2**	**Mag. 3**
1	16	4	4	48	24	12
2	16	8	8	32	12	12
3	20	16	8	44	32	28
4	28	16	4	32	28	8
5	28	12	4	36	24	8
6	32	16	12	12	8	8
7	36	20	12	52	36	16
8	28	8	8	52	20	4
9	20	4	4	40	8	8
10	16	8	4	36	20	4
11	8	8	4	12	8	8
Average		13.33			22.18	

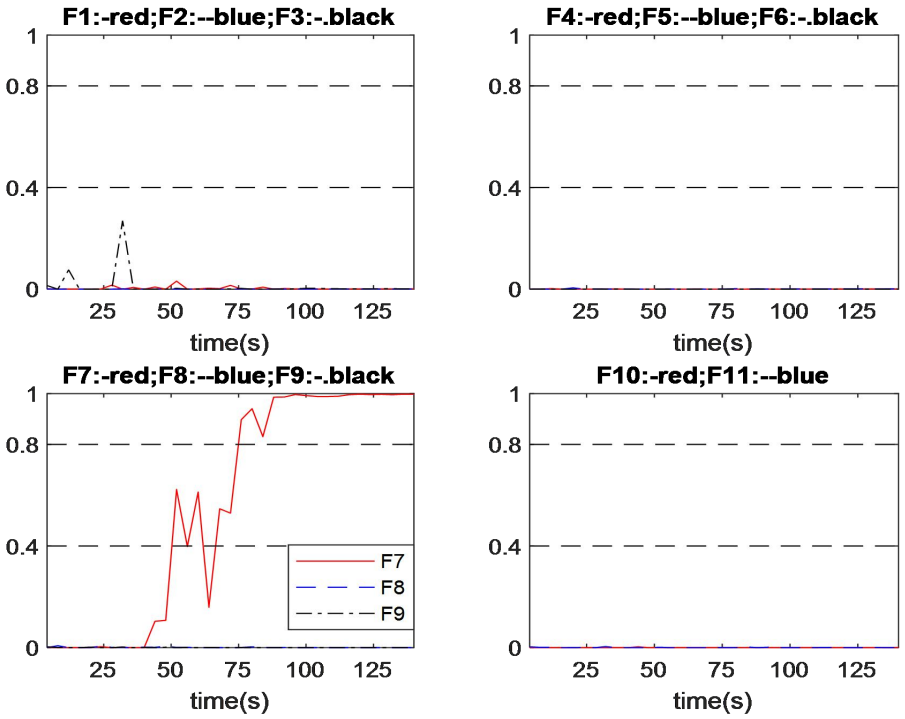

Figure 7.3 Output of fault no.7 diagnosis in abrupt fault with scheme D with relative magnitude of 16.76% (No.1)

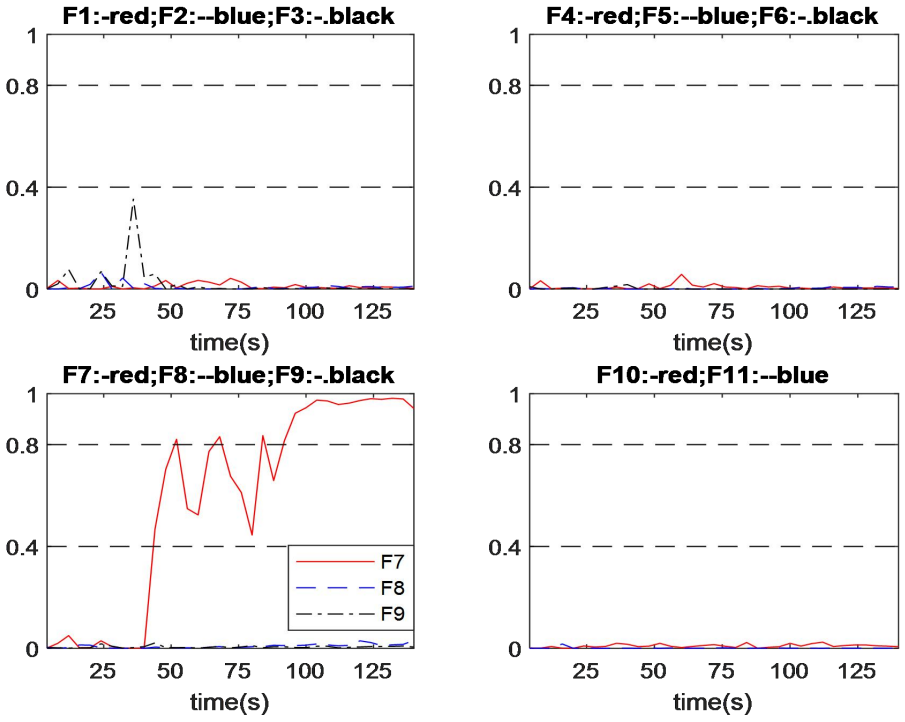

Figure 7.4 Output of fault no.7 diagnosis in abrupt fault with the baseline scheme with relative magnitude of 16.76% (No.1)

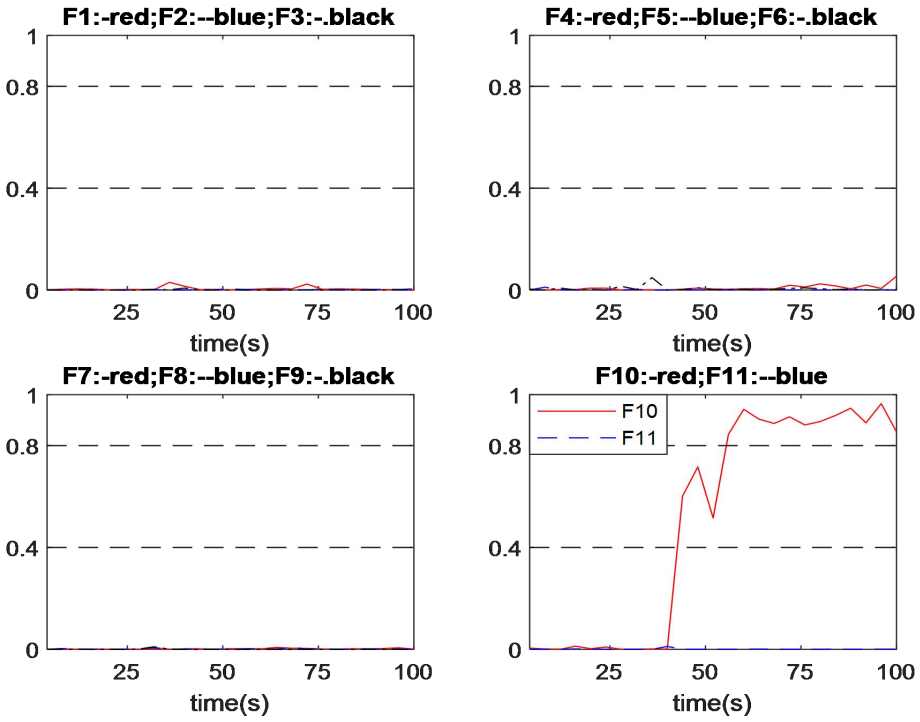

Figure 7.5 Output of fault no.10 diagnosis in abrupt fault with scheme D with relative magnitude of 2.54% (No.1)

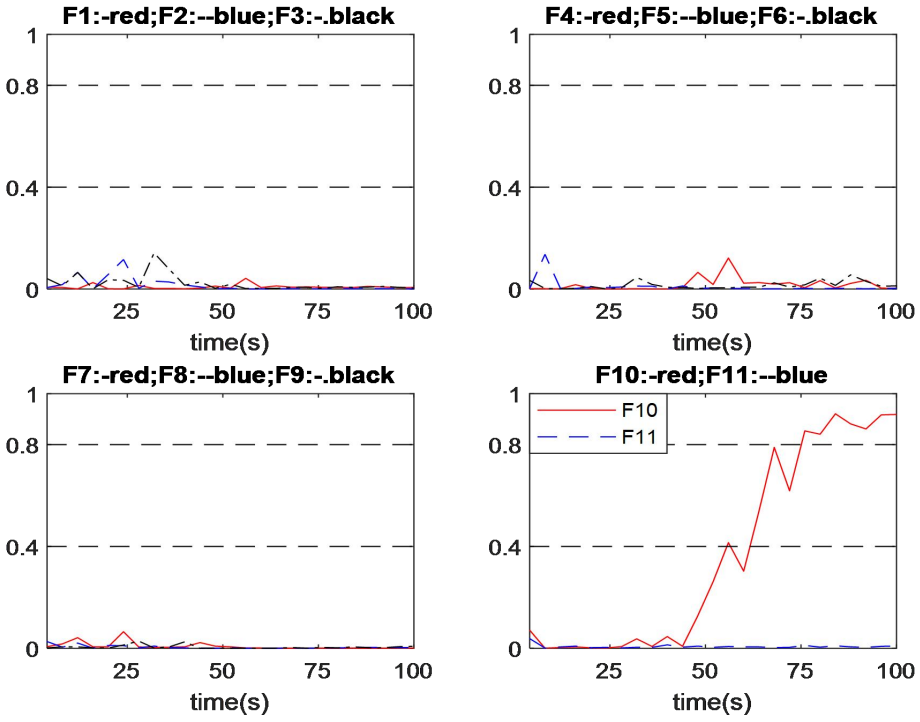

Figure 7.6 Output of fault no.10 diagnosis in abrupt fault with the baseline scheme with relative magnitude of 2.54% (No.1)

7.3.3. Performance under Incipient Faults

Table 7.3 indicates that both diagnosis systems successfully diagnosed all the considered incipient faults. In terms of diagnosis speed, the proposed diagnosis scheme D diagnosed the incipient faults 19.88 s earlier on average than the baseline scheme.

Table 7.3 Fault Diagnosis Time in Incipient Faults

Fault No.	Diagnosis time (s)					
	Proposed Scheme D			Baseline Scheme		
	γ_1	γ_2	γ_3	γ_1	γ_2	γ_3
1	84	60	40	92	64	48
2	72	48	40	88	56	44
3	96	48	32	152	68	52
4	60	40	36	84	76	44
5	116	56	48	152	72	60
6	112	80	72	156	136	88
7	132	88	68	160	100	64
8	76	64	32	96	80	40
9	68	48	40	136	72	48
10	76	72	36	124	80	40
11	132	76	52	144	96	44
Average	66.67			86.55		

Figure 7.7 and Figure 7.8 show, respectively, the performance of scheme D and the baseline scheme in diagnosing incipient fault no. 3, with the fault developing speed of $\gamma = -1.29 \times 10^{-4}(s^{-1})$. As shown in Figure 7.7, after a period of damage accumulation, the output corresponding to fault No. 3 from scheme D responds with fluctuations until reaching close to 1. All other network outputs remain close to 0. The proposed scheme D successfully diagnosed the fault at 96 s. As shown in Figure 7.8, after a period of damage accumulation, the output corresponding to fault No. 3 from the baseline scheme responds with large amplitude fluctuations until reaching close to 1. The responding time is later than that of scheme D, and the fluctuation time is

longer than scheme D. All other network outputs remain close to 0. The baseline scheme successfully diagnosed the fault at 152 s. In this case, the proposed scheme D diagnosed the fault 56 s earlier than the baseline scheme, and the classifier output curve from the proposed scheme D is comparative steady than that of the baseline scheme.

Figure 7.9 and Figure 7.10 show, respectively, the performance of scheme D and the baseline scheme in diagnosing incipient fault no. 7, with the fault developing speed of $\gamma = -3.12 \times 10^{-4}(s^{-1})$. As shown in Figure 7.9, after a period of damage accumulation, the output corresponding to fault no. 7 from scheme D responds quickly with large amplitude fluctuation, even lands downward to 0 one time. All other network outputs are close to 0. The proposed scheme D successfully diagnosed the fault at 132 s. As shown in Figure 7.10, after a period of damage accumulation, the output corresponding to fault no. 7 from the baseline scheme increases to close to 1 with comparatively slower speed and more fluctuations. All other network outputs are close to 0. The baseline scheme successfully diagnosed the fault at 160 s. In this case, the proposed scheme D diagnosed the fault 28 s earlier than the baseline scheme.

Figure 7.11 and Figure 7.12 show, respectively, the performance of scheme D and the baseline scheme in diagnosing incipient fault no. 10, with the fault developing speed of $\gamma = 6.71 \times 10^{-5}(s^{-1})$. As shown in Figure 7.11, after a period of damage accumulation, the output corresponding to fault no. 10 from the proposed scheme D responds rapidly. Ignoring the isolated samples in the outputs, all other network outputs are close to 0. The proposed scheme D successfully diagnosed the fault at 76 s. As shown in Figure 7.12, after a period of damage accumulation, the output corresponding to fault no. 10 from the baseline scheme increases with a slower speed and has fluctuations. All other network outputs are close to 0. The baseline scheme successfully diagnosed the fault at 124 s. In this case, the proposed scheme D diagnosed the fault 48 s earlier than the baseline scheme.

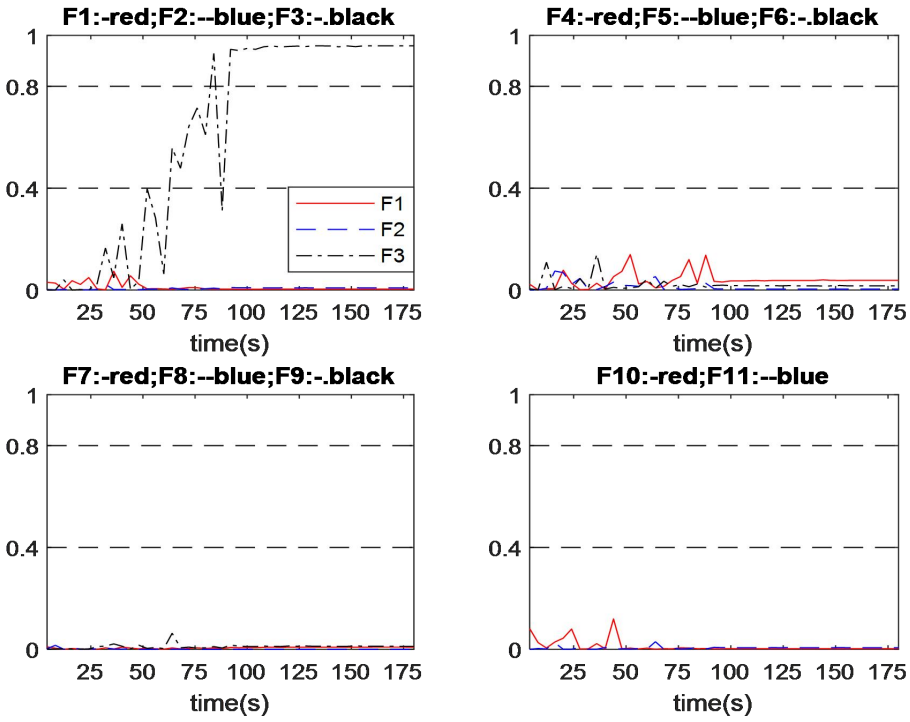

Figure 7.7 Output of fault no.3 diagnosis in incipient fault with scheme D with developing speed of No.1

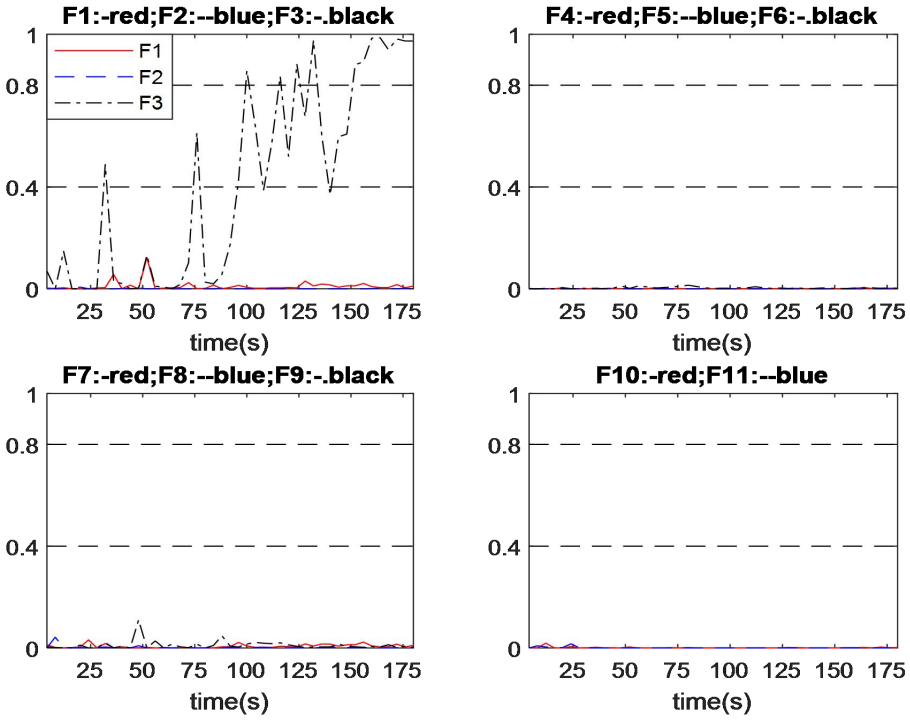

Figure 7.8 Output of fault no.3 diagnosis in incipient fault with the baseline scheme with developing speed of No.1

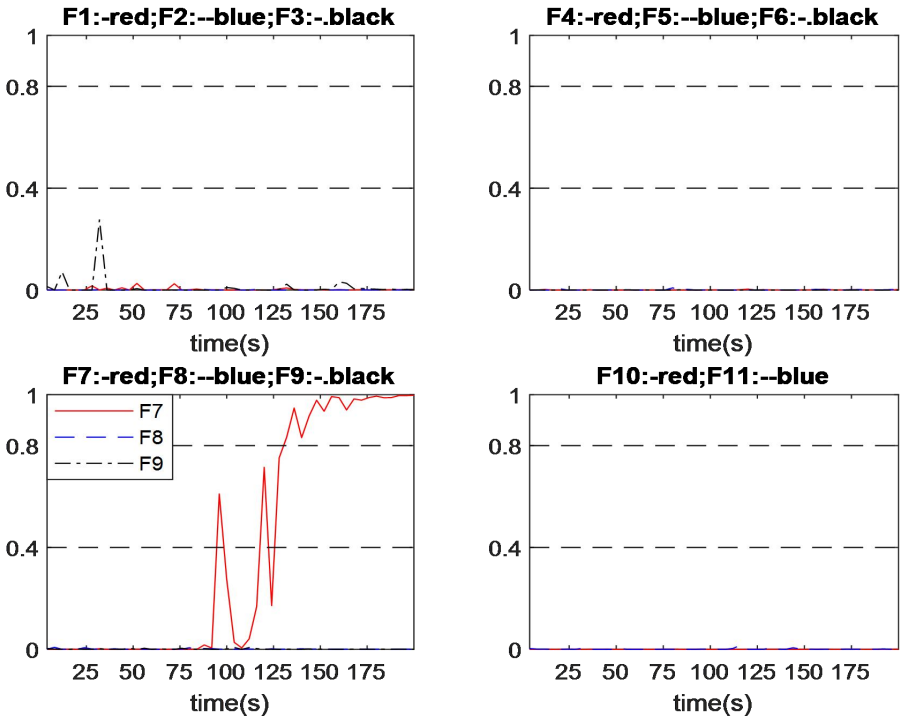

Figure 7.9 Output of fault no.7 diagnosis in incipient fault with scheme D with developing speed of No.1

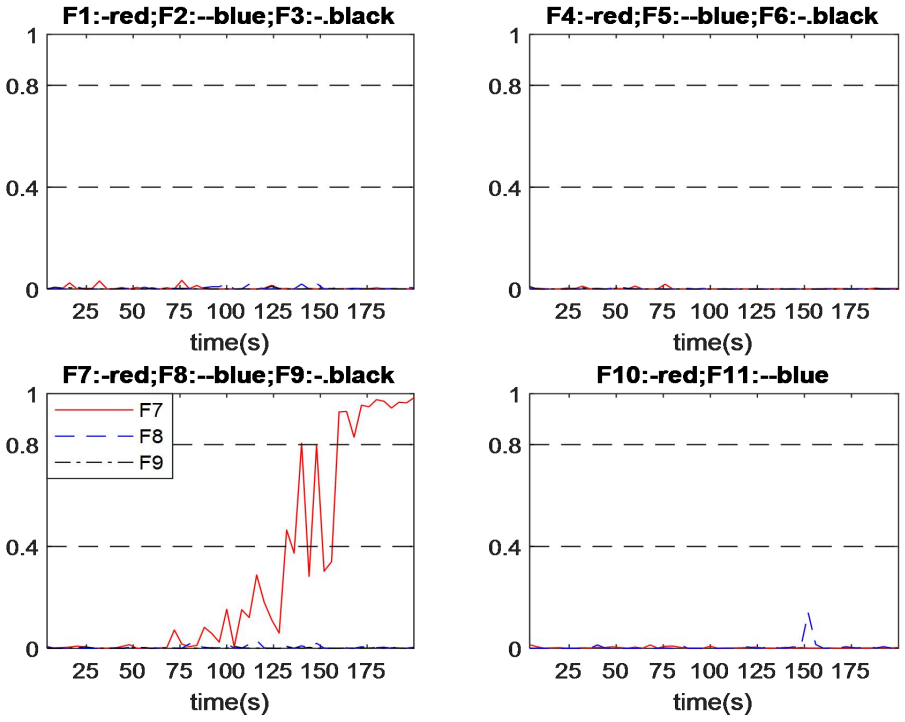

Figure 7.10 Output of fault no.7 diagnosis in incipient fault with the baseline scheme with developing speed of No.1

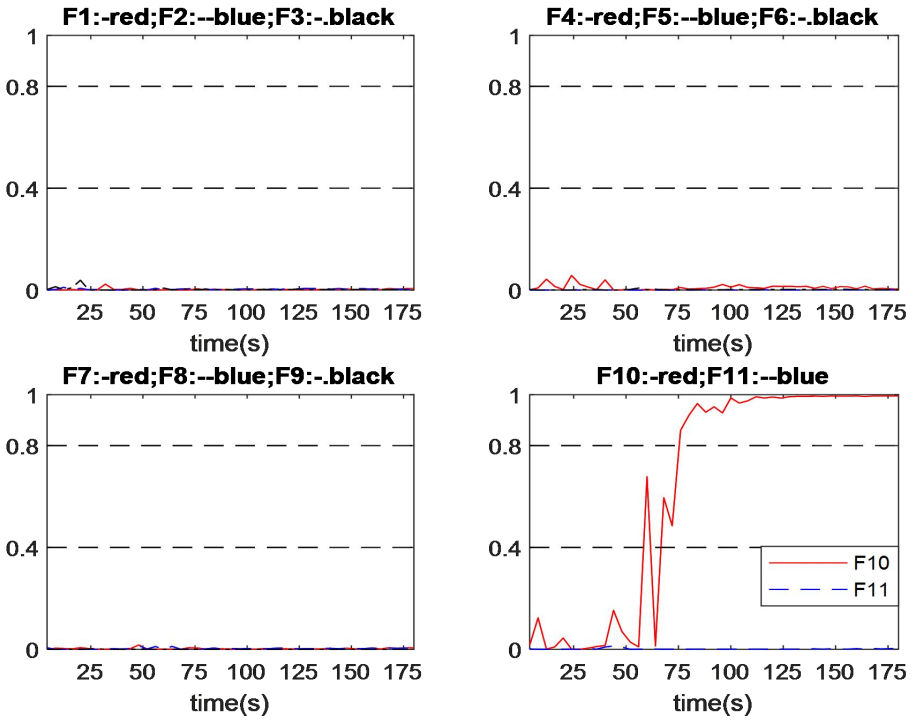

Figure 7.11 Output of fault no.10 diagnosis in incipient fault with scheme D with developing speed of No.1

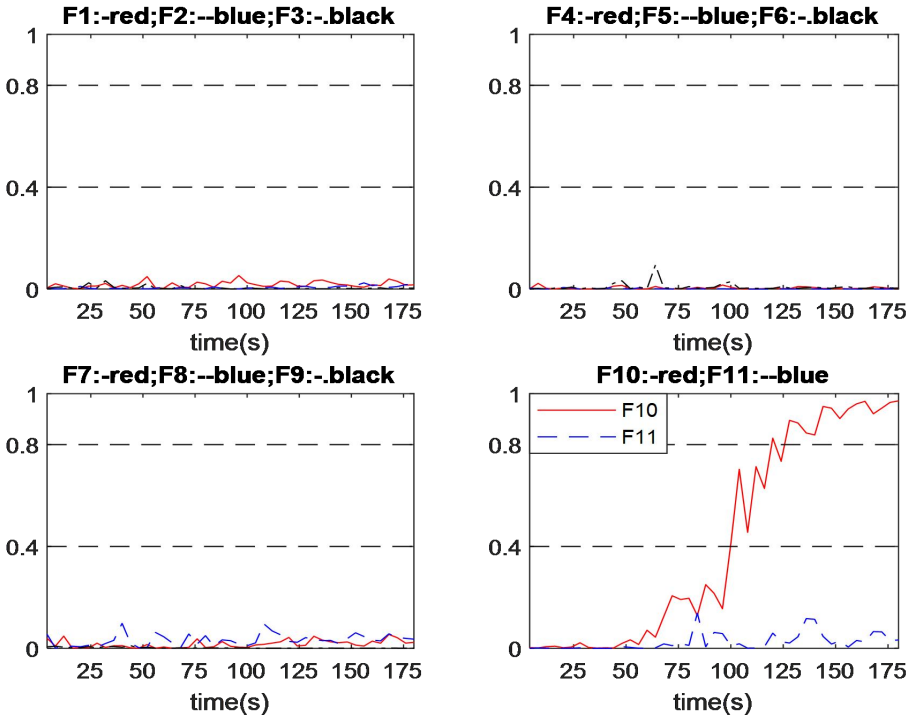

Figure 7.12 Output of fault no.10 diagnosis in incipient fault with the baseline scheme with developing speed of No.1

7.4. Summary

This chapter proposes an intelligent fault diagnosis system based on the techniques of Andrews plot, autoencoder, and neural network. The core idea is the same as scheme C, i.e. using dimensionality reduction to implement fault diagnosis system. This scheme using encoder of autoencoder instead the principal components to achieve the dimensionality reduction. In contrast to PCA which is a linear technique, autoencoder can reduce data dimension in a nonlinear way. Thus, this scheme is expected to give enhanced performance when nonlinear correlation exists among the features in Andrews function.

Compared with the baseline conventional neural network based fault diagnosis system, scheme A, and scheme C, the proposed fault diagnosis system in this chapter gives better diagnosis performance on the CSTR system. Fault diagnosis system using this scheme achieved successful to weaken the limitation of uncertainties impacts and significantly reduced the development time cost of diagnosis system.

Chapter 8. Andrews Function based Intelligent Fault Diagnosis System Integrating Multiple Neural Networks with Information fusion

8.1. Introduction

This chapter proposes a neural network based fault diagnosis system integrating multiple neural networks with information fusion. A single shallow neural network model is hard to be trained perfectly. To overcome the non-robustness of an imperfect single shallow neural network, multiple networks are combined using information fusion for enhanced fault diagnosis performance. It utilizes the technique of bootstrap re-sampling with replacement to produce multiple different replications of the original training data. The proposed fault diagnosis scheme is applied to the CSTR system and compared with the baseline scheme in abrupt faults and incipient faults.

The chapter is organised as follows. Section 8.2 introduces the information fusion in multiple neural networks. Section 8.3 gives the proposed fault diagnosis scheme. Section 8.4 gives the development of the proposed fault diagnosis system, and diagnosis results in abrupt faults and incipient faults. Section 8.5 concludes the chapter.

8.2. Information Fusion in Stacked Neural Network

In chemical industrial process monitoring, it is vital to have reliable fault diagnosis performance. Previous works reveal that a neural network is hard to be trained perfectly on measured process data, but several imperfect neural network models combined via information fusion can improve the reliability of process fault diagnosis (Zhang, 2006). The bootstrap re-sampling with replacement method can be used to produce multiple different replications of the original training data (Efron and Tibshirani, 1993). Figure 8.1 shows the structure of the stacked neural network. The

final output of multiple neural networks is a function of the individual neural network outputs as shown in the following:

$$f[X(t)] = g\{f_i[X(t)]\} \tag{8.1}$$

where $f[X(t)]$ is the final diagnosis result at time t, $f_i[X(t)]$ is the ith neural network output at time t, g is a function for combining the individual neural networks, $X(t)$ is a vector of neural network inputs at time t, N is the number of individual neural networks to be combined.

Previous works suggested several methods to combining the individual networks, such as the methods of averaging and weighted averaging, the method of majority voting, and the method of modified majority voting (Zhang, 2006). The averaging and weighted averaging methods are popularly used in regression problems. But in the classification works, they may give undesired results. The simple averaging method just averages the individual neural networks simply with the combination function as follows:

$$g = \frac{1}{N} \sum_{i=1}^{N} f_i[X(t)] \tag{8.2}$$

The weighted averaging combination function is as the following:

`` $$g = \frac{1}{N} \sum_{i=1}^{N} w_i f_i[X(t)] \tag{8.3}$$

where w_i is the aggregating weight for combining the ith neural network.

The aggregated neural network output in majority voting method is to use the median of individual neural network outputs. The combination function is as the following:

$$g = median\{f_i[X(t)]\}, \quad i = 1,2, \ldots, N \tag{8.4}$$

In this work, the classifier is developed via the modified majority voting combination scheme, which has been evidenced its superiority on chemical process fault diagnosis in previous studies (Zhang, 2006). The scheme is given as:

$$g_j = \begin{cases} Median\{f_{ij}[X(t)]\}, if\ Median\{f_{ij}[X(t)]\} \leq 0.6 \\ Max\{f_{ij}[X(t)]\}, \quad if\ Median\{f_{ij}[X(t)]\} > 0.6 \end{cases} \tag{8.5}$$

$$g = (g_1, g_1, \ldots, g_N);\ \ i = 1,2, \ldots, N;\ j = 1,2, \ldots, m;$$

where g_j is the jth element of g corresponding to the jth fault and f_{ij} is the jth element of f_i corresponding to the jth fault.

According to the outputs of individual networks, the final output can be broadly divided into two cases. If the majority of the individual networks give outputs that are close to 1, the final output is the maximum output of them. This indicates that the corresponding fault exists in the process. If the majority of the individual networks give outputs that are not much larger than 0, the final output is the median of all outputs. This can achieve a more reliable advance warning. The threshold of 0.6 is set based on heuristics and can be possible to fine tune based on the monitoring process dataset.

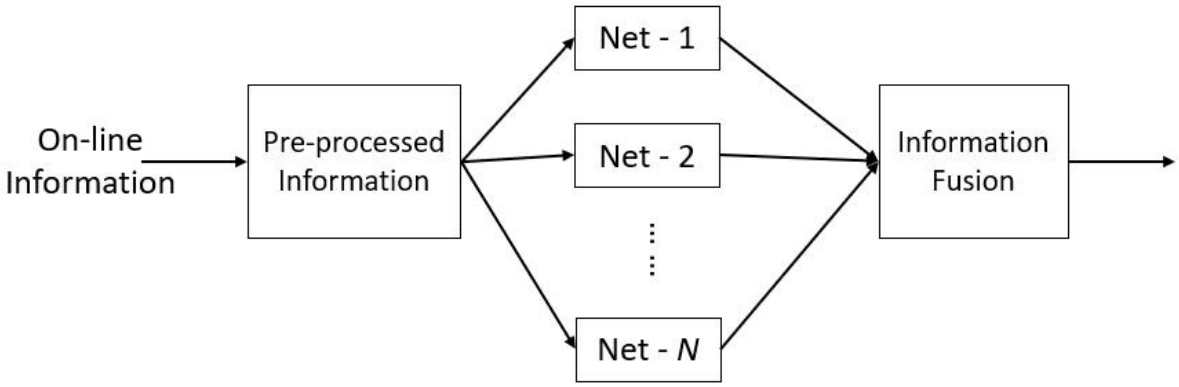

Figure 8.1 Structure of Information Fusion in Multiple Neural Network

8.3. Proposed Fault Diagnosis System

This work proposed a fault diagnosis system based on Andrews function and information fusion in multiple neural networks. In the proposed fault diagnosis scheme (referred to as scheme E), the monitoring process information is pre-processed by Andrews plot method before fed into several trained neural networks and the outputs obtained through the information fusion is the final diagnosis output. In order to demonstrate the improved diagnosis performance of the proposed scheme, it is compared with the baseline scheme. Figure 8.2 gives the framework of scheme E. In scheme E, the monitoring process information is firstly converted into a principal component dataset to re-ordering the data arrangement. The principal components are used in Andrews function with several *t*-values to extract a feature dataset, while maintain the primary information. Finally, the extracted feature dataset fed into the classifier, i.e., stacked neural networks, to produce the outputs and using the information fusion to obtain the final diagnosis outputs. The activation function used in this neural network is sigmoidal function.

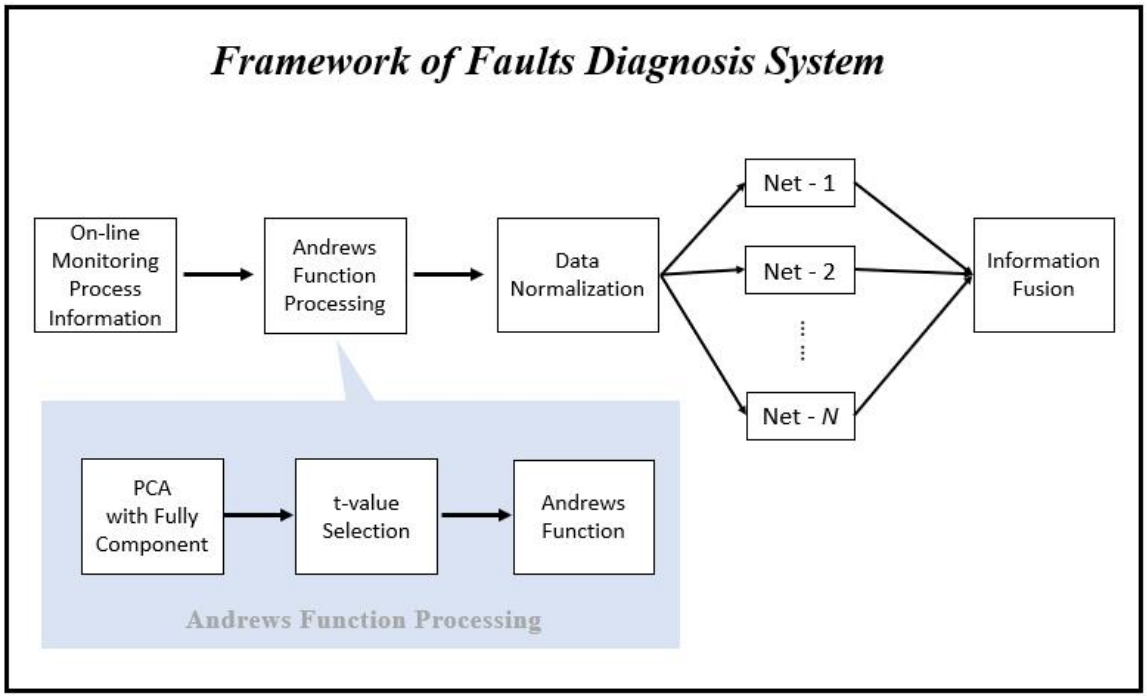

Figure 8.2 Framework of the proposed fault diagnosis system

8.4. Fault Diagnosis Results

8.4.1. Development of the Proposed Fault Diagnosis System

In the proposed scheme E, the selection of *t* values (the number of features) in Andrews function is the same as scheme A. Hence the selected *t* values are [-3.1414, -0.6846, -0.5586, -0.4956, -0.4326, -0.3696, 0.2604, 2.9064, 209694, 3.3024, 3.0954]. In addition, this scheme uses the same numbers of hidden neurons in the neural network as scheme A, i.e., the hidden layer has 15 neurons. In total 10 neural networks are developed and combined. The performance of the proposed scheme E is compared with the baseline scheme in abrupt faults and incipient faults.

8.4.2. Performance under Abrupt Faults

Table 8.1 indicates that modified fault diagnosis scheme E successfully diagnosed all the considered abrupt faults and obtained encouraging performance. In terms of diagnosis speed, the proposed diagnosis scheme E diagnosed the abrupt fault 9.33 s earlier on average than the baseline scheme.

Table 8.1 Fault Diagnosis Time in Abrupt Faults

Fault No.	**Diagnosis time (s)**					
	Proposed Scheme E			**Baseline Scheme**		
	Mag. 1	**Mag. 2**	**Mag. 3**	**Mag. 1**	**Mag. 2**	**Mag. 3**
1	16	4	4	48	24	12
2	16	8	8	32	12	12
3	20	20	8	44	32	28
4	28	8	8	32	28	8
5	12	8	8	36	24	8
6	32	20	8	12	8	8
7	32	20	12	52	36	16
8	20	12	4	52	20	4
9	20	8	8	40	8	8
10	12	8	8	36	20	4
11	8	8	8	12	8	8
Average		12.85			22.18	

Figure 8.3 and Figure 8.4 show, respectively, the performance of scheme E and the baseline scheme in diagnosing abrupt fault no. 4, with the fault relative magnitude of 6.78%. Figure 8.3 shows that the output corresponding to fault no.4 from scheme E responds sharply, when the fault has been initiated, and the classifier output curve is smooth, while all other network outputs remain close to 0. Hence, scheme E successfully diagnosed the fault at 8 s after fault occurred. Figure 8.4 shows that the output corresponding to fault no.4 from the baseline scheme also responds quickly, when the fault has been initiated, but then the capability of fault diagnosis is disturbed by the fluctuations arise. Finally, the outputs are over the diagnosis threshold (0.8) at 28 s after fault occurred. Some slight response in the output corresponding to fault no.11 can be ignored due to the low amplitude. In this case, the proposed scheme E diagnosed the fault 20 s earlier than the baseline scheme, and the classifier output curve is smoother than that of the baseline scheme.

Figure 8.5 and Figure 8.6 show, respectively, the performance of scheme E and the baseline scheme in diagnosing abrupt fault no. 10, with the fault relative magnitude of 3.79%. Figure 8.5 and Figure 8.6 show the output curves corresponding to fault no.10 from scheme E and the baseline scheme respectively. Both schemes successfully diagnosed fault no.10 in this process. The output from the proposed scheme E is smooth and responds quickly. The proposed scheme E diagnosed the fault at 8 s after the fault occurred. The baseline scheme diagnosed the fault at 20 s after fault occurred. Therefore, the proposed scheme E in this case diagnosed the fault 12 s earlier than the baseline scheme, and the classifier output curve is much smoother than that of the baseline scheme.

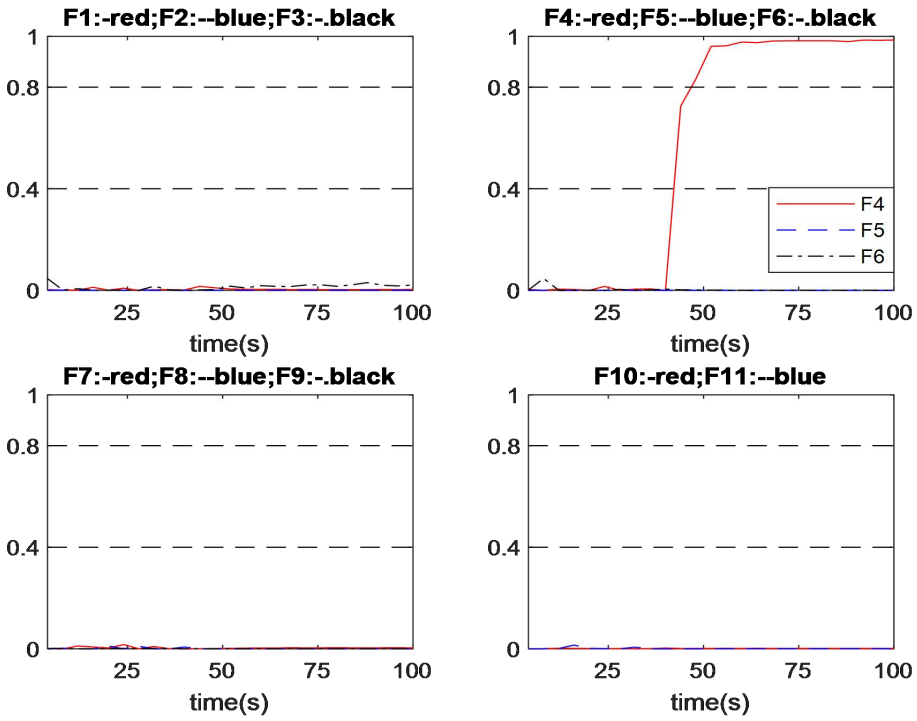

Figure 8.3 Output of fault no.4 diagnosis in abrupt fault with scheme E with relative magnitude of 6.78% (No.2)

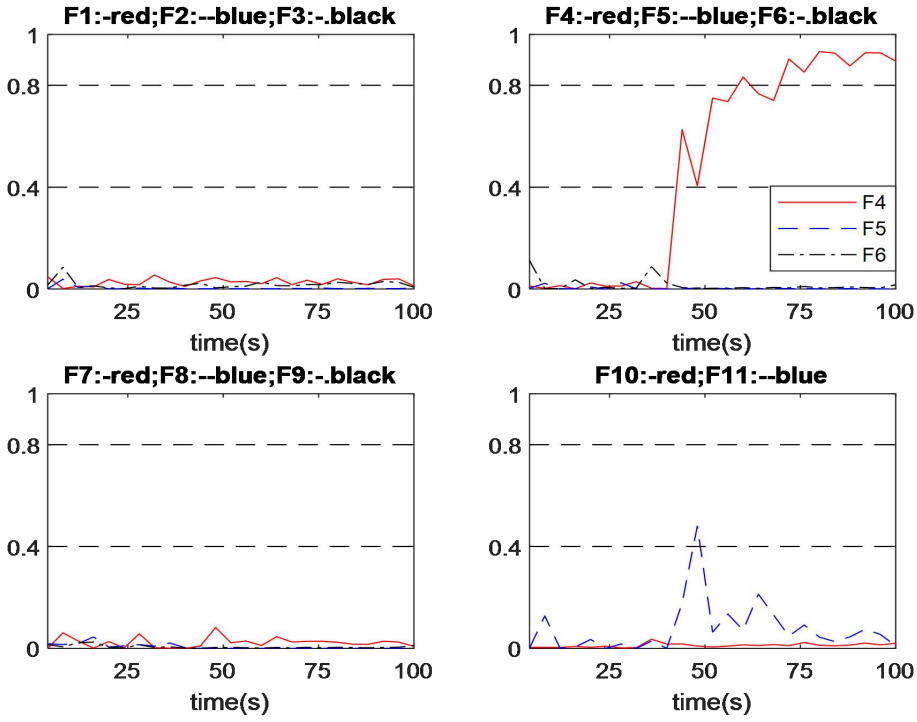

Figure 8.4 Output of fault no.4 diagnosis in abrupt fault with the baseline scheme with relative magnitude of 6.78% (No.2)

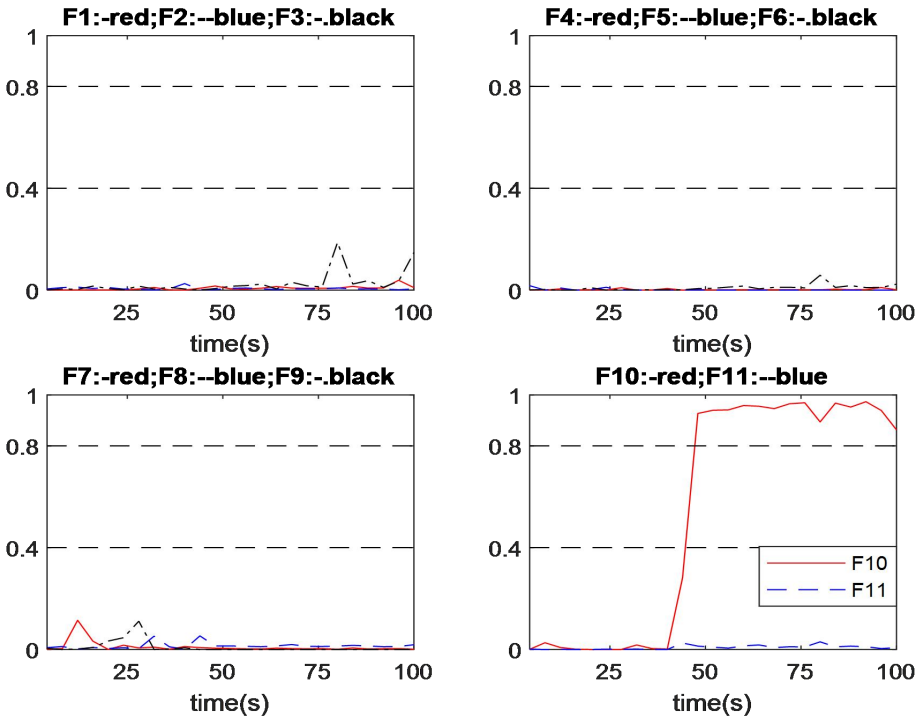

Figure 8.5 Output of fault no.10 diagnosis in abrupt fault with scheme E with relative magnitude of 3.79% (No.2)

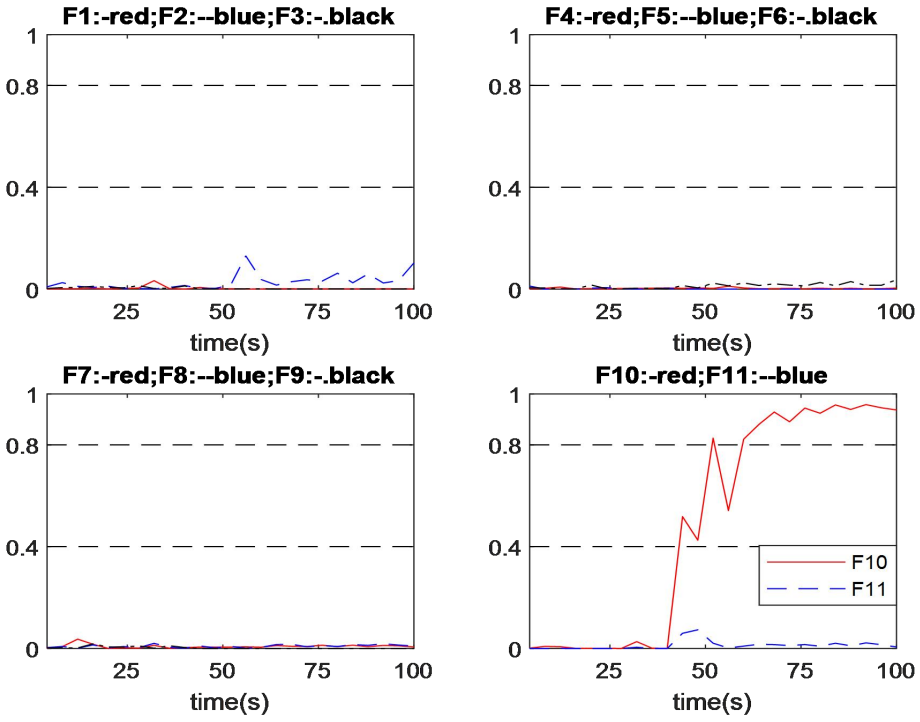

Figure 8.6 Output of fault no.10 diagnosis in abrupt fault with the baseline scheme with relative magnitude of 3.79% (No.2)

8.4.3. Performance under Incipient Faults

Table 8.2 indicates that the proposed fault diagnosis scheme E successfully diagnosed all the considered incipient faults. In terms of diagnosis speed, the proposed diagnosis scheme diagnosed the incipient faults 20.86 s earlier on average than the baseline scheme.

Table 8.4 Fault Diagnosis Time in Incipient Faults

Fault No.	**Diagnosis time (s)**					
	Proposed Scheme E			**Baseline Scheme**		
	γ_1	γ_2	γ_3	γ_1	γ_2	γ_3
1	80	48	36	92	64	48
2	76	48	36	88	56	44
3	88	36	36	152	68	52
4	60	52	32	84	76	44
5	108	56	48	152	72	60
6	128	92	72	156	136	88
7	132	88	60	160	100	64
8	80	72	28	96	80	40
9	80	40	36	136	72	48
10	80	68	32	124	80	40
11	124	72	44	144	96	44
Average		65.69			86.55	

Figure 8.7 and Figure 8.8 show, respectively, the performance of scheme E and the baseline scheme in diagnosing incipient fault no. 4, with the fault developing speed of $\gamma = -1.67 \times 10^{-4} (s^{-1})$. As shown in Figure 8.7, after a period of damages accumulated, the network output corresponding to fault no. 4 from the proposed scheme E raises expeditiously with less oscillations, while all other network outputs remain close to 0. The proposed scheme E successfully diagnosed the fault at 60 s. By contrast, Figure 8.8 shows that after a period of damages accumulated, the output corresponding to fault no. 4 from the baseline scheme raises with more oscillations and at a lower speed. All other network outputs remain lower than 0.2. The baseline scheme successfully diagnosed the fault at 84 s. Overall, in this case, scheme E

diagnosed the fault 24 s earlier than the baseline scheme, and the output curve is smoother than the baseline scheme.

Figure 8.9 and Figure 8.10 show, respectively, the performance of scheme E and the baseline scheme in diagnosing incipient fault no. 9, with the fault developing speed of $\gamma = -1.23 \times 10^{-4}(s^{-1})$. As shown in Figure 8.9 and Figure 8.10, after a period of damage accumulation, output corresponding to fault no. 9 from both schemes responds and quickly reaches the advance warning section. The outputs from the two schemes have a period of slow decline afterwards, but the difference is that the output curve from scheme E is smooth and the output from the baseline scheme declines with fluctuations. Then the output from scheme E has a rapidly increasing and exceed the diagnosis threshold (0.8) at 80 s. By contrast, the output from the baseline scheme also rapidly increases but the rapid upward trend is stagnated under the diagnosis threshold (0.8), and slowly climbs over the diagnosis threshold (0.8) at 136 s. All other outputs are lower than 0.2. Hence, in this case, scheme E successful diagnosed the fault 56 s earlier than the baseline scheme, and the output is much steadier than the baseline scheme.

Figure 8.11 and Figure 8.12 show, respectively, the performance of scheme E and the baseline scheme in diagnosing incipient fault no. 10, with the fault developing speed of $\gamma = 6.71 \times 10^{-5}(s^{-1})$. As shown in Figure 8.11, after a period of damage accumulation, the output corresponding to fault No. 10 from scheme E is responding quickly and growing up close to 1, and the classifier output curve is smooth except a brief dip. As shown in Figure 8.12, after a period of damage accumulation, the output corresponding to fault No. 10 from the baseline scheme is increasing slower than that in scheme E with irregular variations. All other outputs are close to 0. The proposed scheme E successfully diagnosed fault no.10 at 80 s, and the baseline scheme is successful in diagnosing the same fault at 124 s. In this case, scheme E successful diagnosed the fault 44 s earlier than the baseline scheme and the output is steadier than that of the baseline scheme.

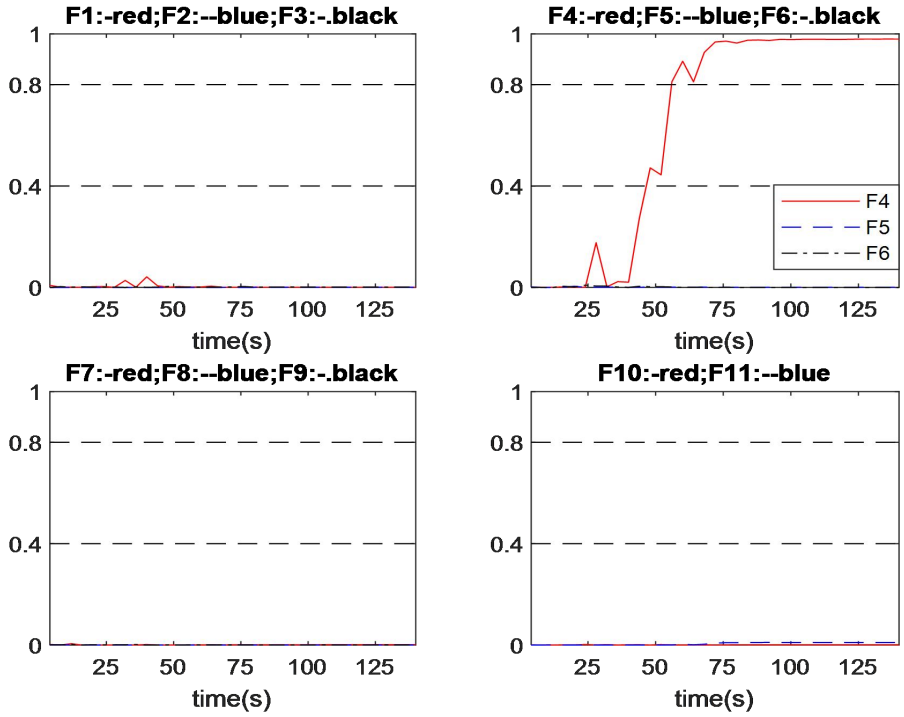

Figure 8.7 Output of fault no.4 diagnosis in incipient fault with scheme E with developing speed of No.1

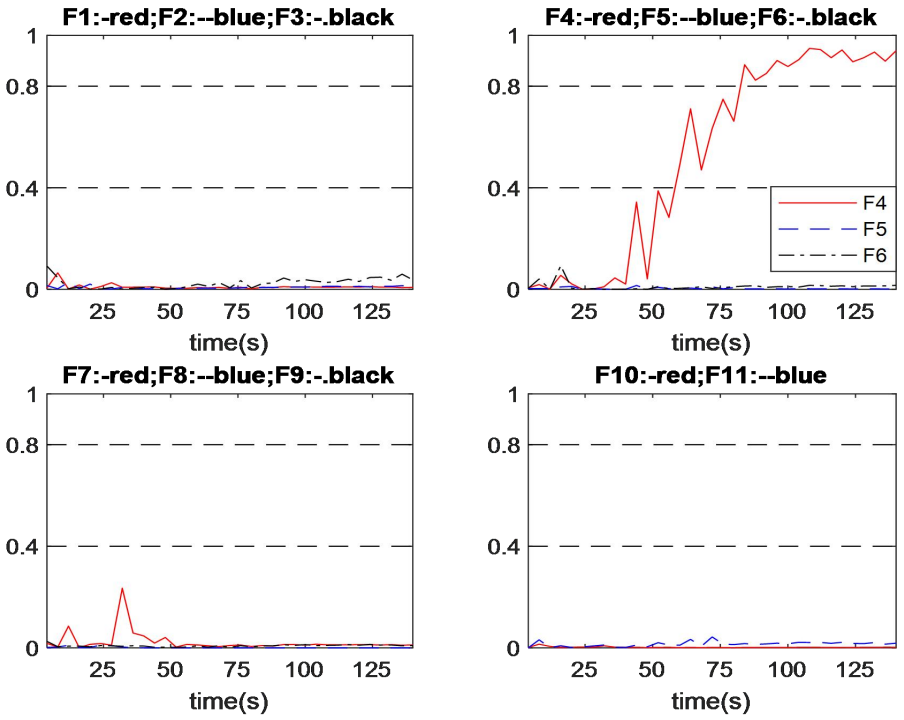

Figure 8.8 Output of fault no.4 diagnosis in incipient fault with the baseline scheme with developing speed of No.1

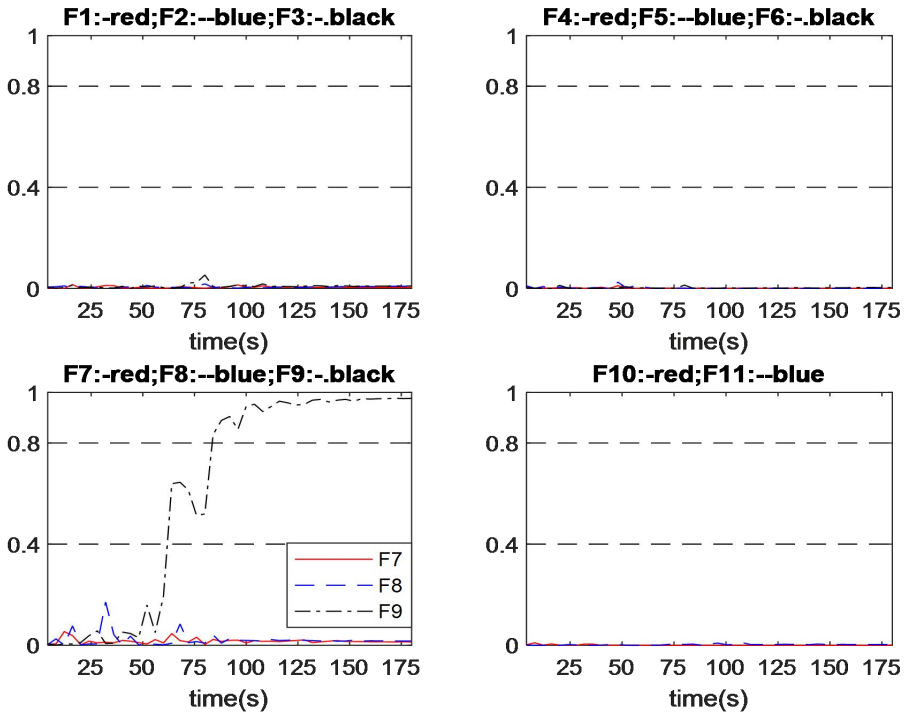

Figure 8.9 Output of fault no.9 diagnosis in incipient fault with scheme E with developing speed of No. 1

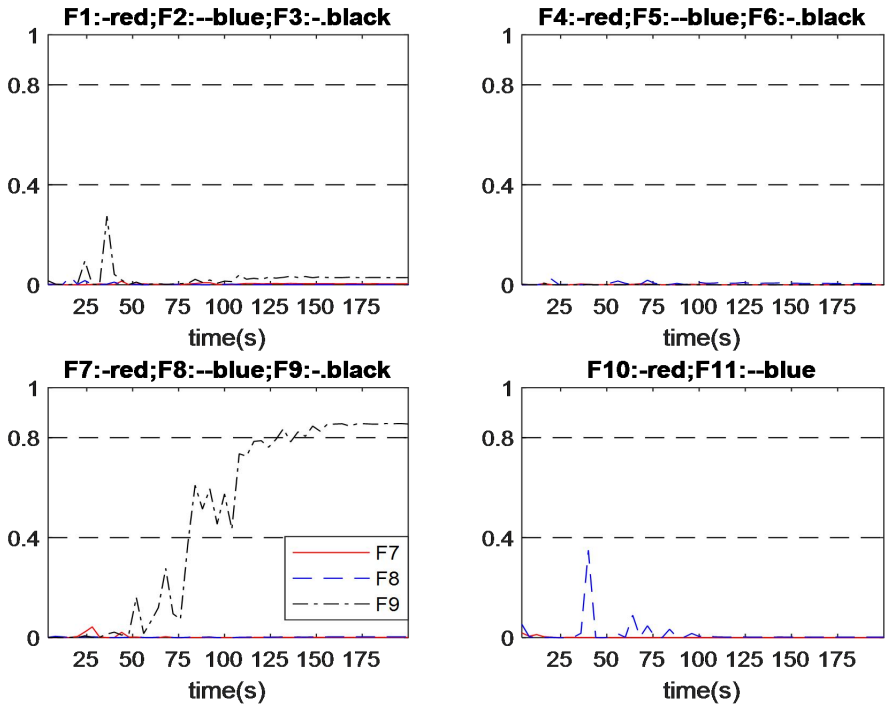

Figure 8.10 Output of fault no.9 diagnosis in incipient fault with the baseline scheme with developing speed of No.1

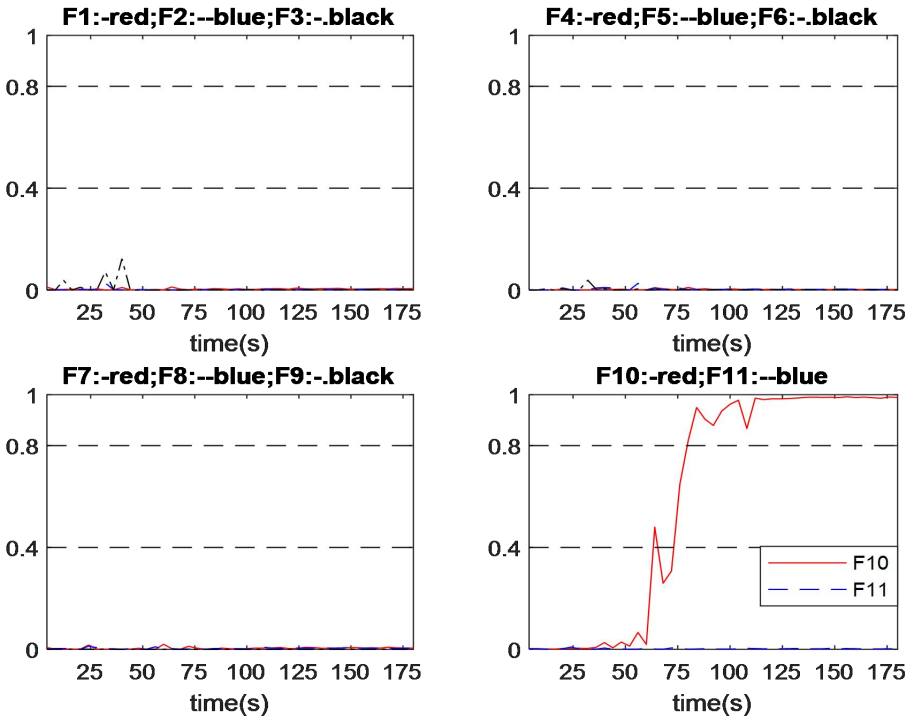

Figure 8.11 Output of fault no.10 diagnosis in incipient fault with scheme E with developing speed of No.1

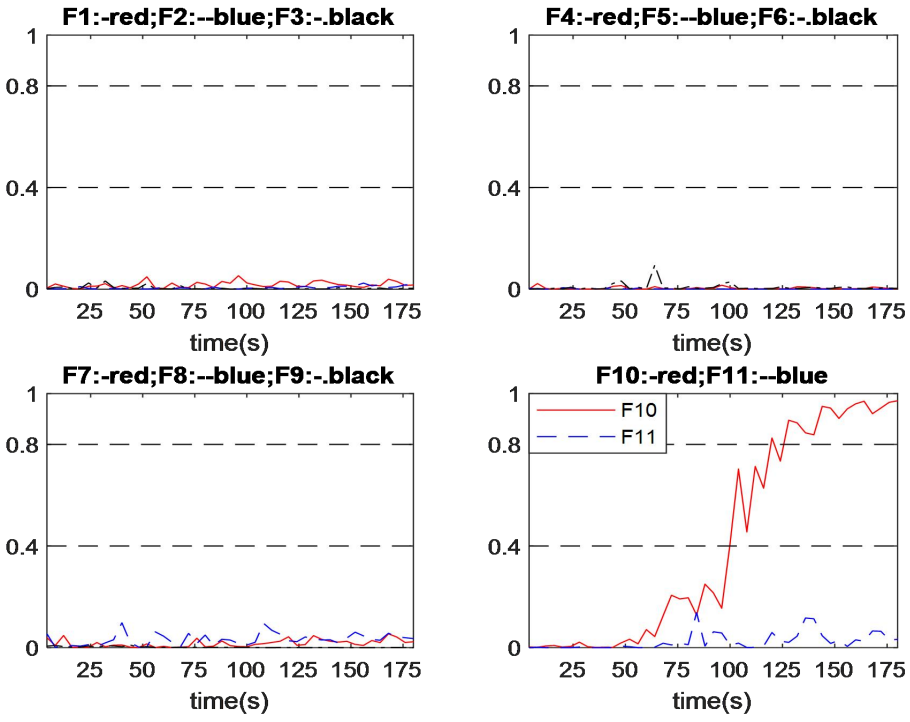

Figure 8.12 Output of fault no.10 diagnosis in incipient fault with the baseline scheme with developing speed of No.1

8.5. Summary

This chapter proposes a neural network based fault diagnosis system integrating Andrews function with multiple neural networks combined through information fusion. In order to guard against the non-robustness of an imperfect shallow neural network model, this work utilizes the bootstrap aggregation of 10 neural networks via information fusion. This scheme tries to use multiple networks with information fusion to attenuate the impact of uncertainty on results. Applications to the CSTR process show very encouraging results. Both diagnosis speed and reliability are improved.

In general, neural network based fault diagnosis system integrating Andrews function with multiple neural networks combined through information fusion is successful. Both diagnosis speed and reliability are obviously improved. The time consumption on development of fault diagnosis system relatively increased, due to the more classifiers need to be trained.

Chapter 9. Intelligent Fault Diagnosis System based on Andrews Plot and Convolutional Neural Network

9.1. Introduction

This chapter proposes an intelligent process fault diagnosis system based on the techniques of Andrews plot and convolutional neural network (CNN). The proposed fault diagnosis method extracts features from the on-line process measurements using Andrews function. The extracted features are then used as the inputs to a convolutional neural network. The core idea is based on that the Andrews function can easily adjust the size of extracted features, which can easily convert the data size to fit the convolutional operation in CNN. This extension and exploration attempt to explore more possibilities of Andrews function in the field of process fault diagnosis. The proposed diagnosis scheme is applied to the CSTR system and compared with the baseline scheme in abrupt faults and incipient faults.

The chapter is organised as follows. Section 9.2 gives the proposed fault diagnosis scheme. Section 9.3 gives the development of the proposed fault diagnosis system, and diagnosis results in abrupt faults and incipient faults. Section 9.4 concludes the chapter.

9.2. Proposed Fault Diagnosis System

In this work, the proposed fault diagnosis system (refer to as scheme F) first converts the measured information to an appropriate dimension by using Andrews function with certain number of *t*-values. Then the outputs of Andrews function are fed into a specific CNN. The length of first convolution kernel is better to be the same as the dimension of inputs. Its depth can be set to 1 or higher according to the need, i.e. 2-D CNN or 3-D CNN. Subsequent convolutional operations are the same as the regularly CNN. For the better comparison and reducing the uncertainties of result observation, the proposed scheme extracts the structure and parameters of a trained

CNN to pre-process the data before fed into the classifier. In other words, this scheme uses the trained CNN structure and parameters as part of the pre-processing. In order to reveal the feasibility of the proposed fault diagnosis scheme, the classifiers in the proposed fault diagnosis scheme and in the baseline scheme use the same method, i.e. a single neural network. Figure 9.1 shows the framework of the proposed fault diagnosis scheme. The proposed scheme can be divided into 3 main parts: Andrews function processing, CNN and classifier. Andrews function processing and CNN are integrated as a complete process information pre-processing system. Through the process information pre-processing system, the online monitoring information can be processed with its features extracted, while maintaining the main information. The dimensions of the final processed feature dataset are determined by the numbers of *t*-values and the adopted convolutional operations.

As previously mentioned, the system uses the principal component dataset X' instead the original information dataset X to eliminate the impact of variable ordering. Then Andrews function with certain number of *t*-values is used to obtain the information-containing feature dataset $F_x(t)$ from X'. Then $F_x(t)$ is fed into a trained CNN to acquire the final processed features which are then processed by a classifier. The process data used in this study is generated from the simulated CSTR system, hence each sample of monitoring information consists of 13 measured variables. Figure 9.2 gives the processing of each sample in the proposed scheme. In this case, each sample, $X = [x_1, x_2, ..., x_{13}]$, is first converted to principal components, $X' = [x'_1, x'_2, ..., x'_{13}]$, which are then fed into Andrews function (4.1) with 24 *t*-values. The *t*-values are uniformly distributed in the range from –π to π. Then they are fed to a CNN to extract the features. Convolution layer 1 convert the data into a feature matrix with size of 24×24. Then calculate the data with convolution layers 2 and 3. After the convolutional operations, the data are converted into a feature matrix with size of 4×4, which is then transformed into a feature vector with16 elements before fed into the input layer of classifier, i.e. a single network.

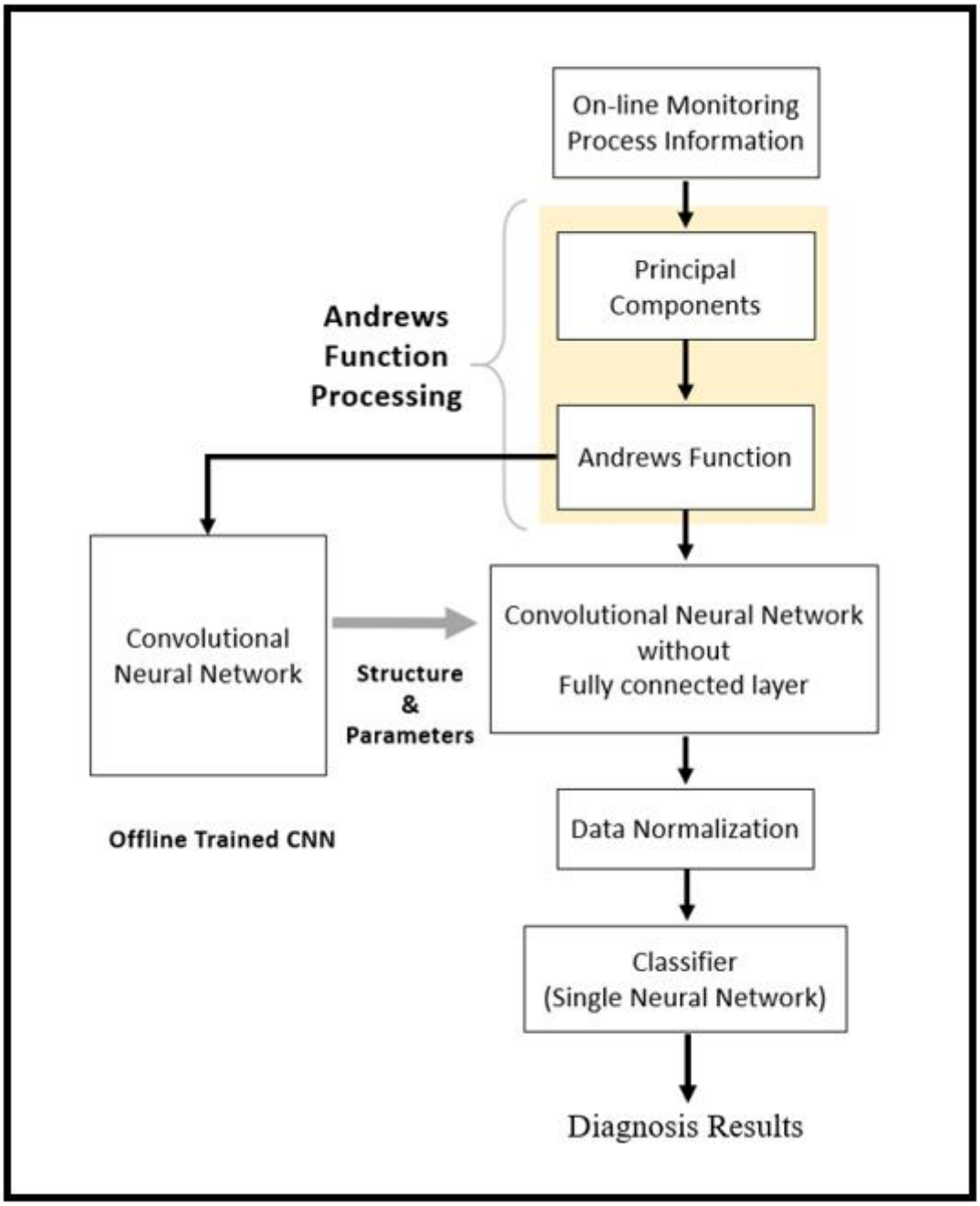

Figure 9.1 Framework of the proposed fault diagnosis system

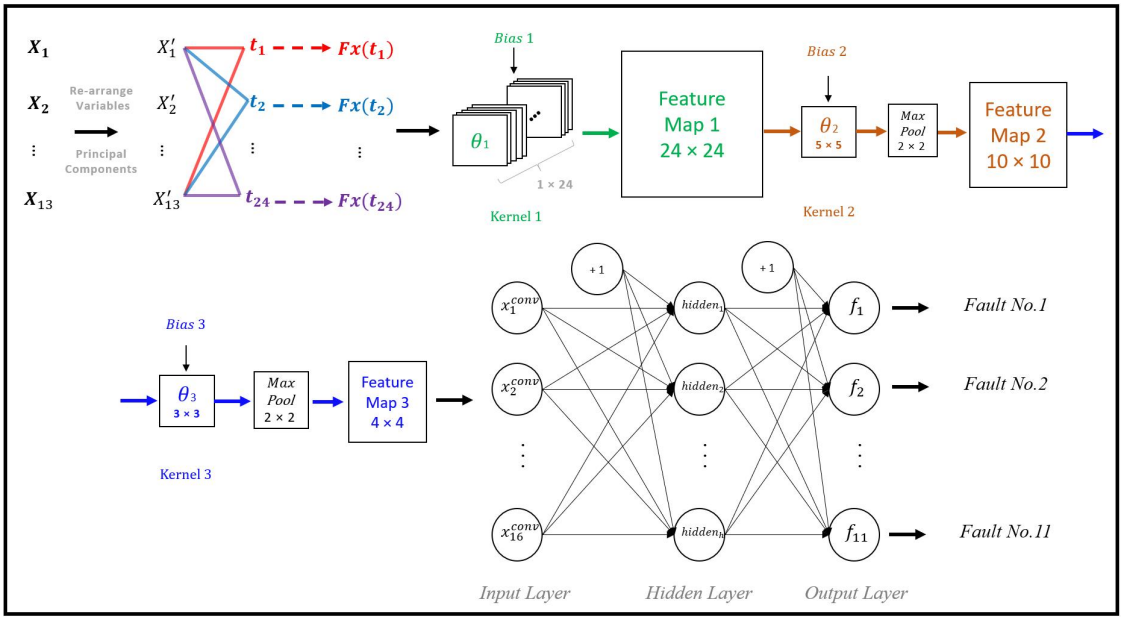

Figure 9.2 Process of information processing in the proposed scheme

9.3. Fault Diagnosis Results

9.3.1. Development of the Proposed Fault Diagnosis System

The proposed fault diagnosis system (refer to as scheme F) uses the 24 t values in Andrews function processing with the value distributed uniformly in the range [-π, π] to convert the process information dimension from 13 to 24. 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. These parameters are manually adjusted. The classifier used is the same as that in the baseline scheme. Table 9.1 gives the accuracy on the testing data of different numbers of hidden neuron in the classifier. The best performance is marked with bold font. Table 9.2 gives the numbers of neurons in different layers in scheme F.

Table 9.1 Determination of Numbers of Hidden Neurons in Classifier

Scheme F							
Determined Neuron Number		**Accuracy of Different Numbers of Hidden Neurons**					
Layer	**Neuron**	Numbers of HN	Accuracy	Numbers of HN	Accuracy	Numbers of HN	Accuracy
Input	**16**	12	87.78%	15	90.56%	18	96.67%
Hidden	**17**	13	93.89%	16	92.78%	19	92.78%
Output	**11**	14	96.11%	**17**	**97.22%**	20	85.83%

Table 9.2 Layer structure and neurons in system

Scheme F		
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

9.3.2. Performance under Abrupt Faults

Table 9.3 indicates that scheme F successfully diagnosed all the considered abrupt faults and obtained encouraging performance. In terms of diagnosis speed, the proposed diagnosis scheme F diagnosed the abrupt faults 8.6 s earlier on average than the baseline scheme.

Table 9.3 Fault Diagnosis Time in Abrupt Faults

Fault No.	Diagnosis time (s)					
	Proposed Scheme F			Baseline Scheme		
	Mag. 1	Mag. 2	Mag. 3	Mag. 1	Mag. 2	Mag. 3
1	12	4	4	48	24	12
2	16	8	4	32	12	12
3	16	8	8	44	32	28
4	32	8	8	32	28	8
5	20	12	4	36	24	8
6	36	32	16	12	8	8
7	32	16	8	52	36	16
8	16	8	8	52	20	4
9	24	12	12	40	8	8
10	20	12	8	36	20	4
11	8	8	8	12	8	8
Average	13.58			22.18		

Figure 9.3 and Figure 9.4 show, respectively, the diagnosis outputs from scheme F and the baseline scheme in diagnosing abrupt fault no. 1, with the fault relative magnitude of 1.67%. The proposed scheme F successfully diagnosed the fault at 12 s after the fault occurred. The overall response of the classifier output curves are quite stable. Figure 9.4 shows that the output corresponding to fault no. 1 from the baseline scheme responds quickly after the fault occurred, but then a period of oscillation arises in the area of advance warning. Then the output corresponding to fault no.1 increases and continuous remains over the diagnosis threshold (0.8), while all other network outputs are close to 0. Hence the baseline scheme success diagnosed fault no.1 at 48 s after the fault occurred. In this case, the proposed scheme F diagnosed the fault 36 s earlier than the baseline scheme.

Figure 9.5 and Figure 9.6 show, respectively, the performance of scheme F and the baseline scheme in diagnosing abrupt fault no. 8, with the fault relative magnitude of

2.10%. The output corresponding to fault no. 8 from scheme F responds quickly, then a low amplitude oscillation occurs, but most part of them are over the diagnosis threshold (0.8). Hence, scheme F successfully diagnosed the fault no.8 at 16 s after the fault occurred. The output corresponding to fault no.8 from the baseline scheme oscillates for a relatively long period before settling down to over the diagnosis threshold. The baseline scheme successfully diagnosed the fault at 52 s after the fault occurred. In this case, scheme F diagnosed the fault 36 s earlier than the baseline scheme.

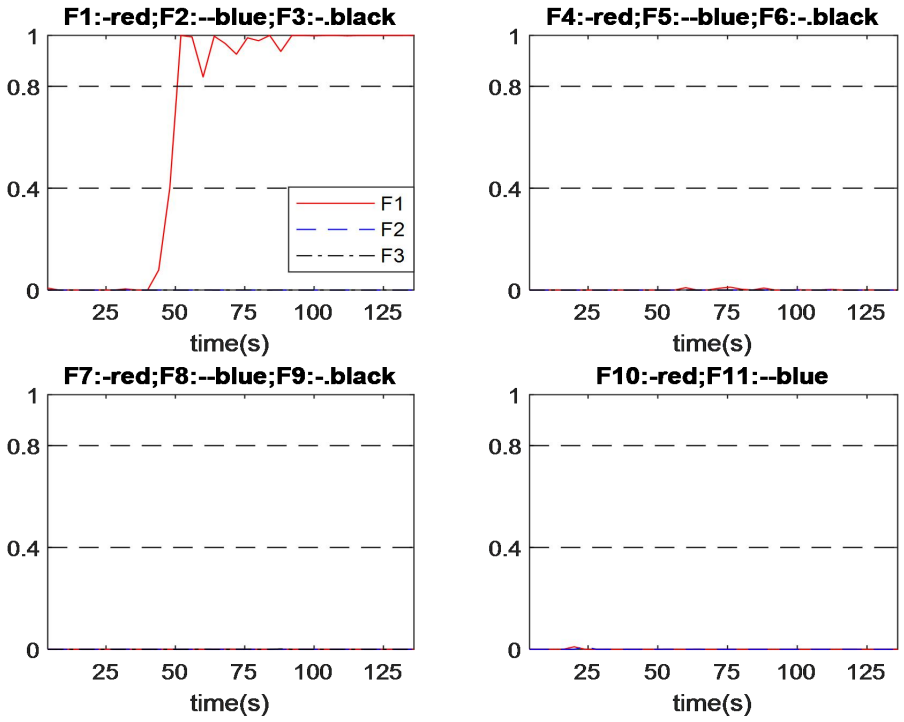

Figure 9.3 Output of fault no.1 diagnosis in abrupt fault with scheme F with relative magnitude of 1.67% (No.1)

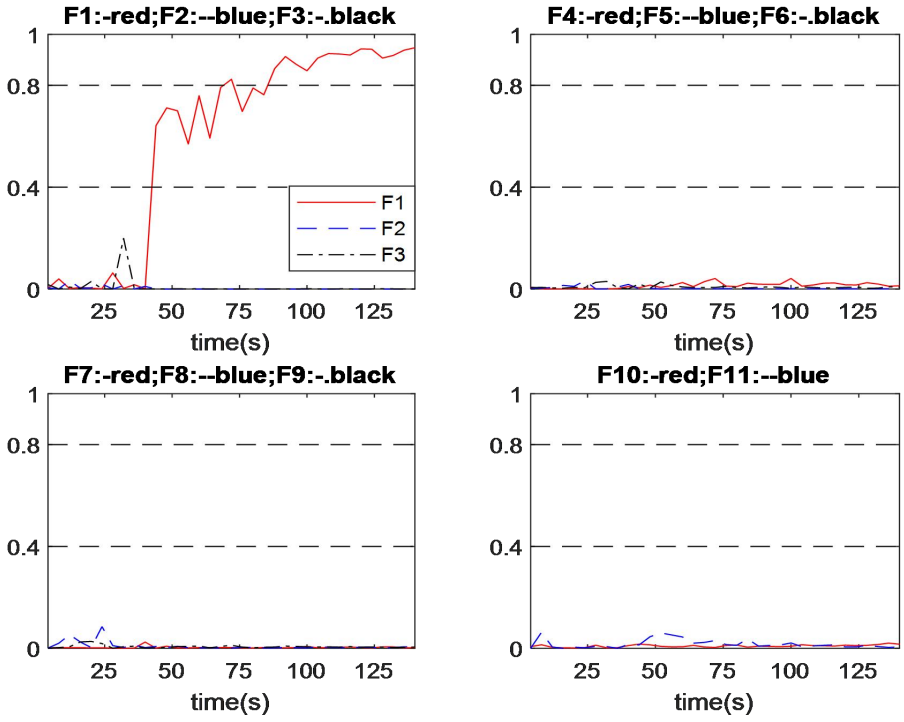

Figure 9.4 Output of fault no.1 diagnosis in abrupt fault with the baseline scheme with relative magnitude of 1.67% (No.1)

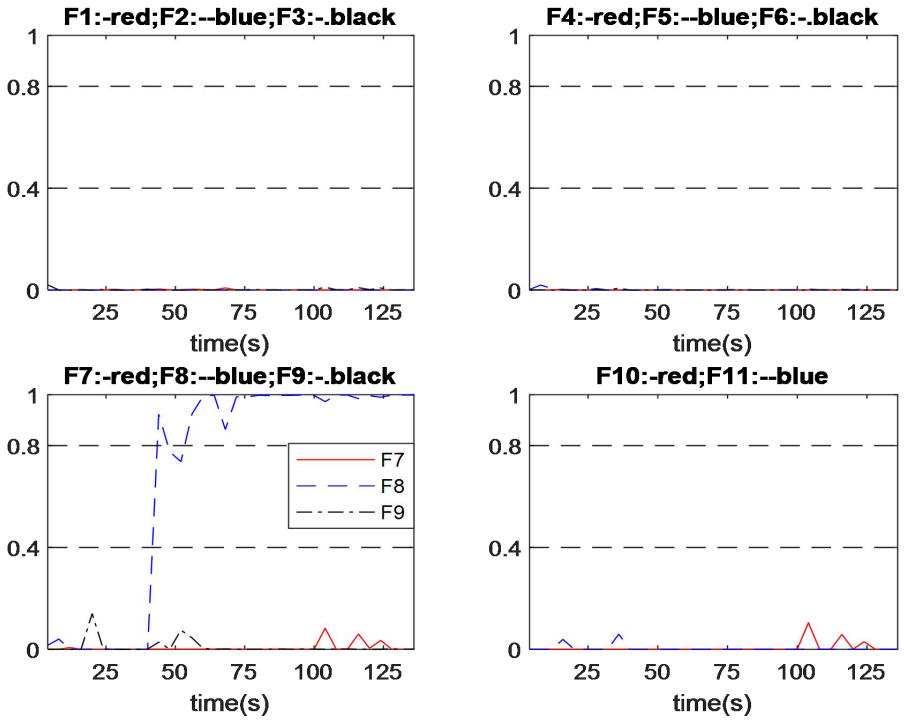

Figure 9.5 Output of fault no.8 diagnosis in abrupt fault with scheme F with relative magnitude of 2.10% (No.1)

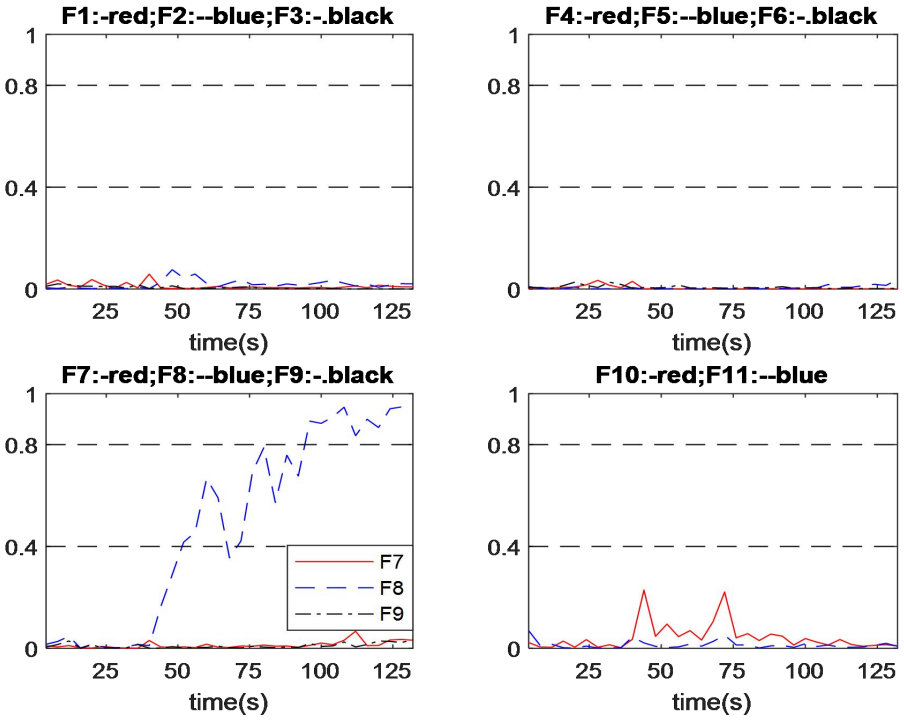

Figure 9.6 Output of fault no.8 diagnosis in abrupt fault with the baseline scheme with relative magnitude of 2.10% (No.1)

9.3.3. Performance under Incipient Faults

Table 9.4 indicates that the proposed scheme F successfully diagnosed all the considered incipient faults. In terms of diagnosis speed, the proposed diagnosis scheme diagnosed the incipient faults 18.43 s earlier on average than the baseline scheme.

Table 9.4 Fault Diagnosis Time in Incipient Faults

Fault No.	Diagnosis time (s)					
	Proposed Scheme F			Baseline Scheme		
	γ_1	γ_2	γ_3	γ_1	γ_2	γ_3
1	68	36	32	92	64	48
2	68	48	40	88	56	44
3	80	40	36	152	68	52
4	68	48	32	84	76	44
5	108	60	52	152	72	60
6	112	92	60	156	136	88
7	128	92	56	160	100	64
8	56	48	36	96	80	40
9	112	108	72	136	72	48
10	92	68	36	124	80	40
11	124	84	56	144	96	44
Average		68.12			86.55	

Figure 9.7 and Figure 9.8 show, respectively, the diagnosis outputs of scheme F and the baseline scheme in diagnosing incipient fault no. 3, with a fault developing speed of $\gamma = -1.29 \times 10^{-4}(s^{-1})$. As shown in Figure 9.7, the output corresponding to fault no.3 rises rapidly after a period of damage accumulation from the fault occurred. All other network outputs are close to 0. The proposed scheme F successfully diagnosed the fault at 68 s after fault no.3 occurred. As shown in Figure 9.8, the output corresponding to fault no.3 responds quickly, but has a long duration of fluctuations.

Finally, the baseline scheme successfully diagnosed the fault at 152 s after this fault occurred. In this case, scheme F diagnosed the fault 84 s earlier than the baseline scheme. The classifier output curves in scheme F are more stable than those in the baseline scheme.

Figure 9.9 and Figure 9.10 show, respectively, the performance of scheme F and the baseline scheme in diagnosing incipient fault no. 8, with a fault developing speed of $\gamma = 7.13 \times 10^{-5}(s^{-1})$. As shown in Figure 9.9 and Figure 9.10, after the period of fault developing, the network output corresponding to fault no. 8 in scheme F raises quickly close to 1, but that of the baseline scheme raises slowly. All other outputs are close to 0. The output curve in scheme F is very smooth. Scheme F and the baseline scheme successfully diagnosed the fault at 56 s and 96 s, respectively, after the fault occurred. In this case, the proposed scheme F diagnosed the fault 40 s earlier than the baseline scheme.

Figure 9.11 and Figure 9.12 show, respectively, the performance of scheme F and the baseline scheme in diagnosing incipient fault no. 10, with a fault developing speed of $\gamma = 6.71 \times 10^{-5}(s^{-1})$. As shown in Figure 9.11 and Figure 9.12, after the period of fault developing, the network output corresponding to fault no. 10 in scheme F raises quickly to close to 1, but that in the baseline scheme raises relatively slowly. The output corresponding to fault No. 10 in scheme F has some fluctuations before the output gets close to 1. These isolated samples can be ignored. Scheme F and the baseline scheme successfully diagnosed the fault at 92 s and 124 s, respectively, after the fault occurred. In this case, the proposed scheme F diagnosed the fault 32 s earlier than the baseline scheme.

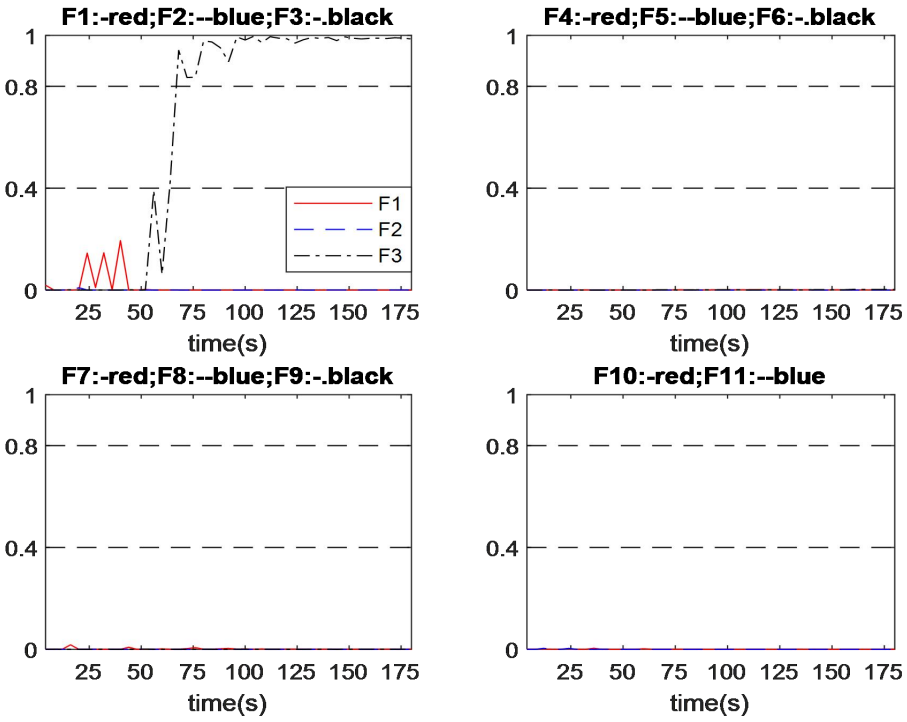

Figure 9.7 Output of fault no.3 diagnosis in incipient fault with scheme F with developing speed of No.1

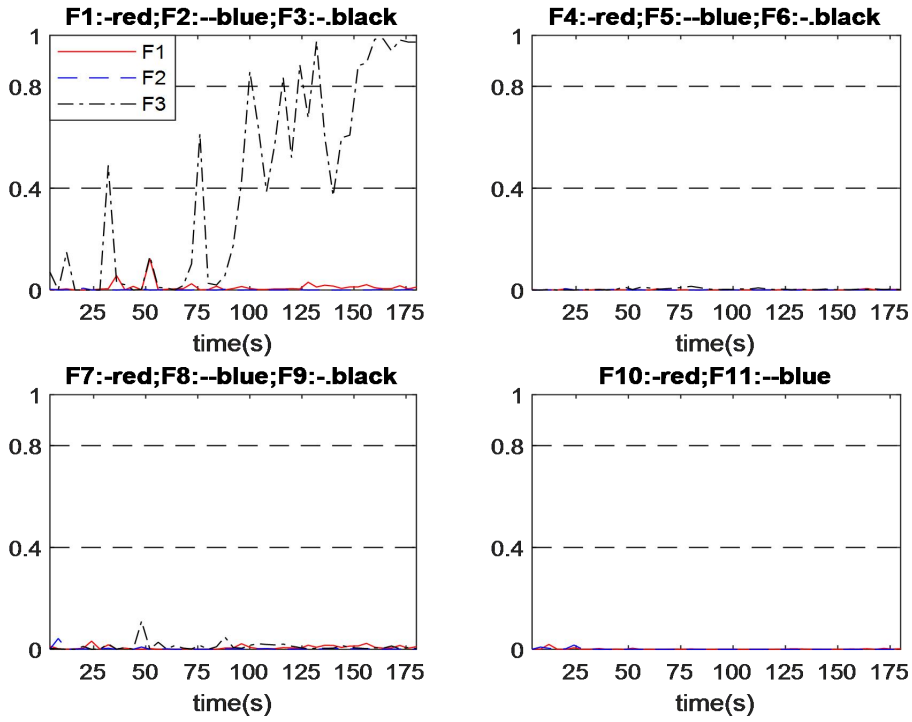

Figure 9.8 Output of fault no.3 diagnosis in incipient fault with baseline scheme with developing speed of No.1

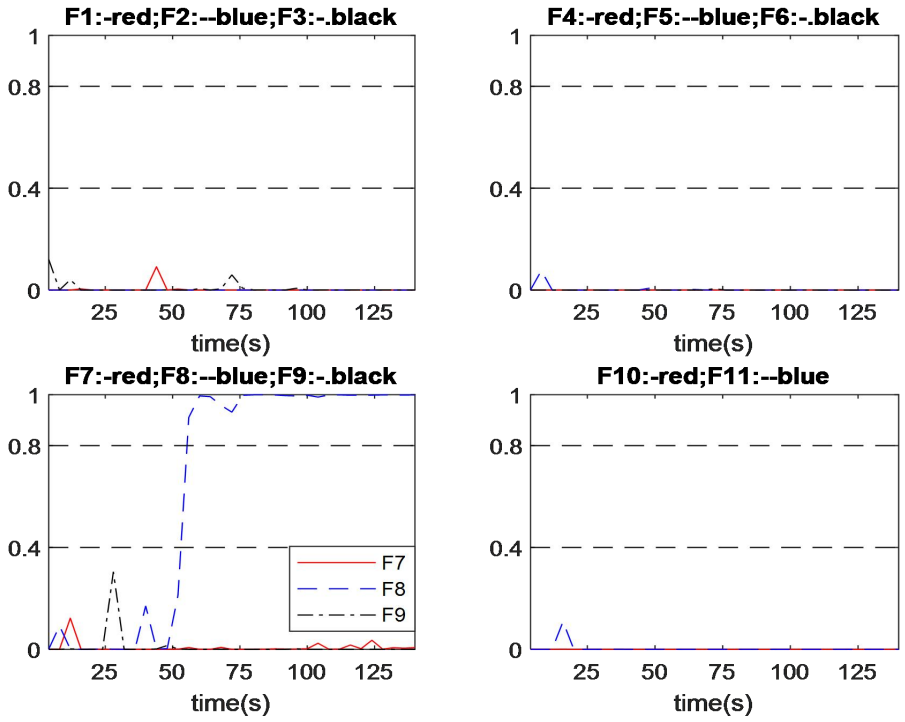

Figure 9.9 Output of fault no.8 diagnosis in incipient fault with scheme F with developing speed of No.1

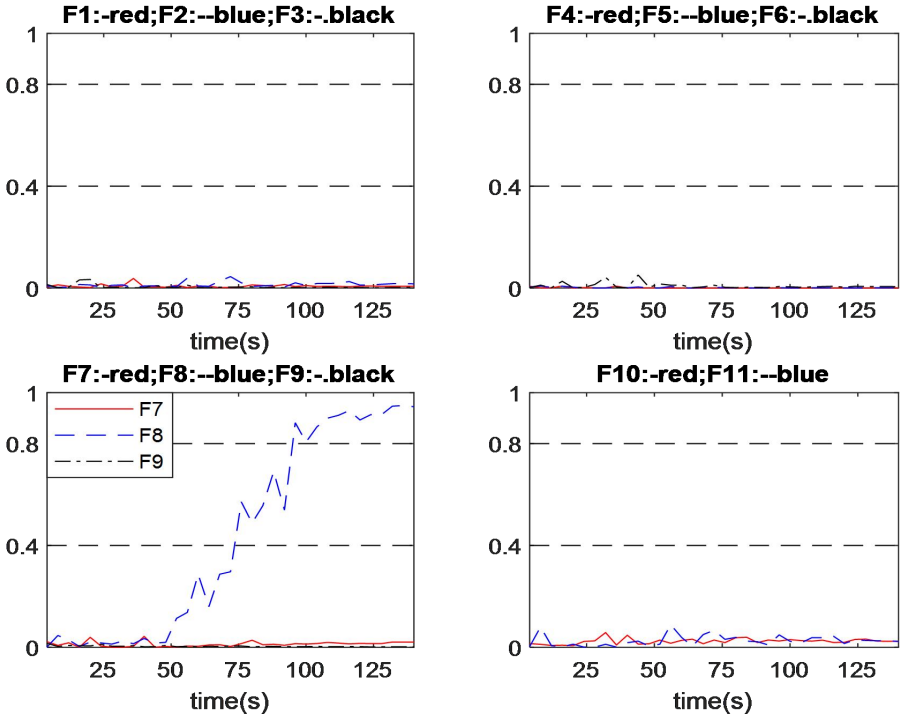

Figure 9.10 Output of fault no.8 diagnosis in incipient fault with baseline scheme with developing speed of No.1

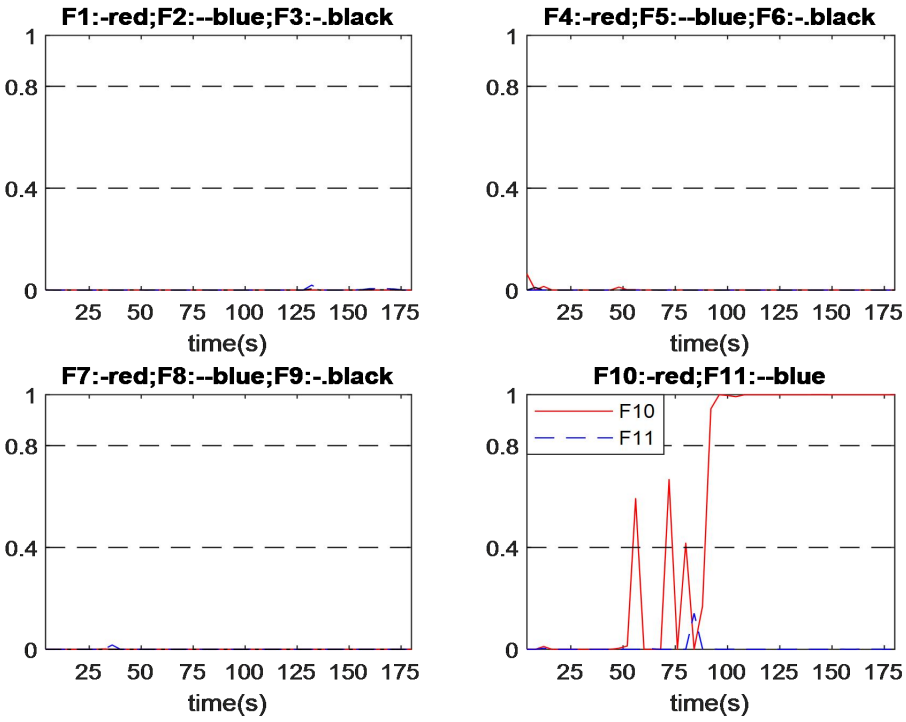

Figure 9.11 Output of fault no.10 diagnosis in incipient fault with scheme F with developing speed of No.1

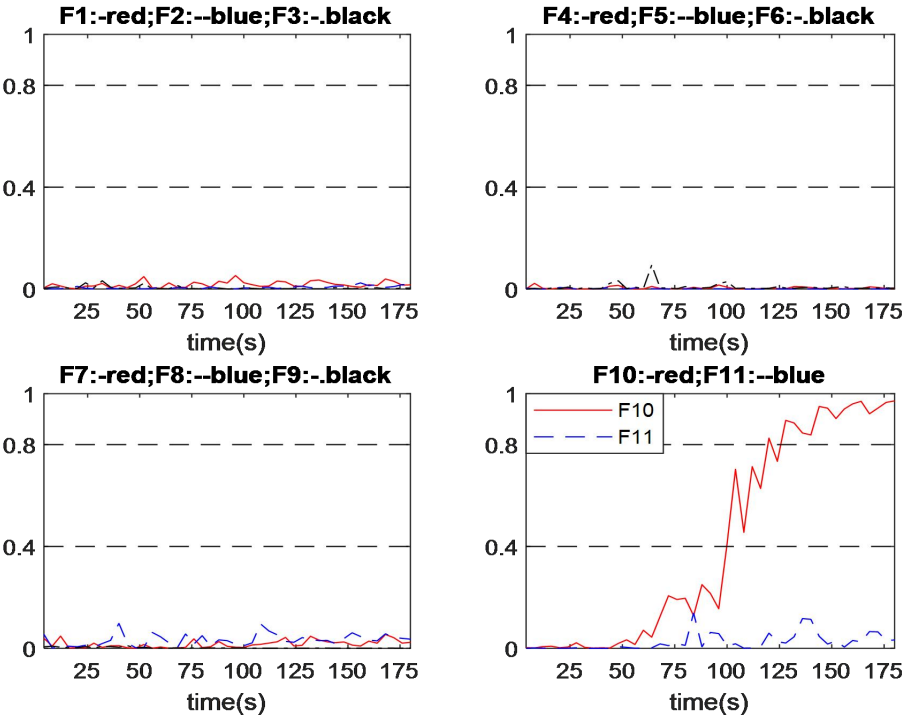

Figure 9.12 Output of fault no.10 diagnosis in incipient fault with baseline scheme with developing speed of No.1

9.4. Summary

This chapter 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 (scheme A), 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.

This scheme shows that Andrews function processing in fault diagnosis system can be combined with deep network. The diagnosis system using scheme F can achieve better performance than scheme A and conventional scheme. But the system structure is more complex, that may bring uncertainties and the time consumption in development is significantly increased. Future works will consider integration of Andrews function with alternative deep neural networks such as light-weight CNN for on-line process fault diagnosis.

Chapter 10. Conclusions and Suggestions for Future Works

10.1 Conclusions

This thesis proposed six intelligent fault diagnosis systems based on Andrews plot, neural network, and techniques that qualitative trend analysis, principal component analysis, autoencoder, multiple neural networks with information fusion, and convolutional neural network. All of the proposed fault diagnosis systems are applied to a CSTR system, where 11 faults in the form of abrupt faults or incipient faults are considered. In order to demonstrate the superiority of the proposed fault diagnosis schemes, a traditional fault diagnosis system based on neural network is also developed as a comparative study. The results indicate that all of the proposed process fault diagnosis schemes have enhanced the fault diagnosis performance.

The proposed schemes are summarised as follows.

Scheme A is an enhanced intelligent neural network based online process fault diagnosis system developed by integrating Andrews plot and neural network techniques. By using features extracted from Andrews plot as the inputs to a neural network, the diagnosis speed and reliability can be improved. A method for determining the important features in Andrews function is also proposed. In addition, the proposed data pre-processing method is highly effective in adjusting the high dimensional data to an appropriate size.

Scheme B is a neural network based fault diagnosis system developed through the fusion of Andrews function and qualitative trend analysis. The features from the original measurements are extracted using Andrews function with certain numbers of *t*-values. The extracted features are next converted into qualitative trend form, before fed into a neural network. Applications to a simulated CSTR process proved that the proposed diagnosis scheme can advance the fault diagnosis performance. Converting

features extracted from Andrews function into qualitative trend form could make the system more sensitive to faults and enhance the fault diagnosis performance.

In general, scheme B is successful. Although this scheme are not working for the limitations in scheme A, but the encouraging results of application to CSTR prove that proposed diagnosis scheme is feasible, and it can effectively subside the impact of uncertainties.

Scheme C is a neural network based fault diagnosis system developed through the fusion of Andrews function and dimensional reduction by principal component analysis. To overcome the problem of the uncertainty in selecting the features in Andrews function, a large number of *t*-values are used in the Andrews function and then the high-dimensional features are processed by using PCA to reduce the dimensional, before fed into a neural network. Compared with the baseline conventional neural network based fault diagnosis system and scheme A, scheme C gives better diagnosis performance on the CSTR system. System development integrating PCA to reduce the dimensional can reduce the uncertainty in selecting *t*-values in Andrews function.

In general, scheme C is successful. Aiming at the uncertainties caused by parameter selection in scheme A, the PCA dimensionality reduction technique is used for parameter selection, which has achieved good results and greatly reduced the time consumption for system development. This indicates that the dimensionality reduction technique has a great potential to overcome the impacts of uncertainties from parameter selection in Andrews function processing.

Scheme D is an intelligent fault diagnosis system based on the techniques of Andrews plot, autoencoder, and neural network. The core idea is same as scheme C, i.e. using dimensionality reduction to overcome the uncertainty in selecting *t*-values in Andrews function. This scheme using the encoder of autoencoder instead PCA to achieve the dimensionality reduction. Compared with the baseline conventional neural network based fault diagnosis system, scheme A, and scheme C, scheme D gives better diagnosis performance on the CSTR system. The reason that scheme D gives better

performance than scheme C is that autoencoder performs nonlinear dimension reduction whereas PCA only performs linear dimension reduction.

In general, the intelligent based fault diagnosis system integrating with Andrews plot and autoencoder is successful. Similar as scheme C, scheme D achieved better performance than scheme A and scheme C, and also the system developing speed up, albeit slower than C. This further suggests that the dimensionality reduction techniques have great potential to overcome the impacts of uncertainties from parameter selection in Andrews function processing.

Scheme E is a neural network based fault diagnosis system integrating Andrews function with multiple neural networks combined through information fusion. In order to guard against the non-robustness of an imperfect shallow neural network model, this work utilizes the bootstrap aggregation of 10 neural networks via information fusion. Applications to the CSTR process show very encouraging results. Both diagnosis speed and reliability are improved. This scheme dose not attempt to work on the limitations in feature selections in Andrews function. Instead, it tries to using multiple networks with information fusion to improve the performance of the classifier.

In general, neural network based fault diagnosis system integrating Andrews function with multiple neural networks combined through information fusion is successful. Both diagnosis speed and reliability are obviously improved. The time consumption on development of fault diagnosis system relatively increased, due to the more classifiers need training.

Scheme F is 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.

This scheme shows that Andrews function processing in fault diagnosis system can be combined with deep network. The diagnosis system using scheme F can achieve better performance than scheme A and conventional scheme. But, the system structure are more complex, that may bring uncertainties, and the time consumption are significantly increased. Future works will consider integration of Andrews function with alternative deep neural networks such as light-weight CNN for on-line process fault diagnosis.

The contributions of this thesis are summarized as follow:

1. It is first time the Andrews plot has been exploited and integrated with neural networks for fault detection and diagnosis. The features extracted by Andrews plot would help the subsequent fault diagnosis by neural networks.

2. A method for determining the important features in Andrews plot is proposed.

3. Reducing uncertainty associated with parameter selection in Andrews plot by integrating with principal component analysis, qualitative trend analysis, convolutional neural network, and autoencoder has been proposed.

10.2 Suggestions for Future Works

Although the proposed schemes are shown to be effective in process fault diagnosis and perform much better than the conventional neural network based fault diagnosis scheme, they still have some limitations. These limitations include the requirement of process data covering various process faults, some uncertainties in setting the parameters such as the number of features in Andrews function, and the time consumption of in system development. Some future works can be carried out to overcome these limitations.

In all the proposed schemes in this thesis, shallow neural networks (i.e. single hidden layer neural networks) are used as the classifiers. Deep neural network based

classifiers, such as deep belief network, could be investigated in the future for further improving the fault diagnosis performance.

Stacking multiple neural networks is used in scheme E and shows improved performance. Stacking multiple neural networks can improve the classification performance of single neural networks. Scheme E can be considered as the modification of scheme A by using stacked neural networks as classifier. All other schemes can be extended in the future by using stacked neural networks as classifier.

Data pre-processing part also has many improvement possibilities. Further studies on the trend analysis to overcome the uncertainties from Andrews function can be carried out. In addition, other mathematical techniques may have a chance to improve the diagnosis performance, such as slow feature analysis (Wiskott and Sejnowski, 2002), and rough sets theory (Pawlak, 1982).

In the six proposed schemes, historical process data covering normal and various faulty operating conditions are required. They may encounter problems when new faults occur. Future research can be conducted to integrate the proposed schemes with clustering techniques so that any new faults can be discovered as new clusters different from the known fault classes.

References

Abdi, H., 2007. Singular value decomposition (SVD) and generalized singular value decomposition. Encyclopedia of measurement and statistics, pp.907-912.

Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A. and Arshad, H., 2018. State-of-the-art in artificial neural network applications: A survey. Heliyon, 4(11), p.e00938.

Aguilar-López, R., 2003. Integral observers for uncertainty estimation in continuous chemical reactors: algebraic-differential approach. Chemical Engineering Journal, 93(2), pp.113-120.

Ahmad, Z. and Zhang, J., 2009. Selective combination of multiple neural networks for improving model prediction in nonlinear systems modelling through forward selection and backward elimination. Neurocomputing, 72(4-6), pp.1198-1204.

Ali, J.M., Hussain, M.A., Tade, M.O. and Zhang, J., 2015. Artificial Intelligence techniques applied as estimator in chemical process systems–A literature survey. Expert Systems with Applications, 42(14), pp.5915-5931.

Alter, O., Brown, P.O. and Botstein, D., 2000. Singular value decomposition for genome-wide expression data processing and modeling. Proceedings of the National Academy of Sciences, 97(18), pp.10101-10106.

Andrews, D.F., 1972. Plots of high-dimensional data. Biometrics, pp.125-136.

Balasubramanian, M., Schwartz, E.L., Tenenbaum, J.B., de Silva, V. and Langford, J.C., 2002. The isomap algorithm and topological stability. Science, 295(5552), pp.7-7.

Baldi, P. and Sadowski, P.J., 2013. Understanding dropout. Advances in neural information processing systems, 26, pp.2814-2822.

Baraldi, P., Podofillini, L., Mkrtchyan, L., Zio, E. and Dang, V.N., 2015. Comparing the treatment of uncertainty in Bayesian networks and fuzzy expert systems

used for a human reliability analysis application. Reliability Engineering & System Safety, 138, pp.176-193.

Bengio, Y., 2009. Learning deep architectures for AI. Now Publishers Inc.

Bengio, Y., Courville, A. and Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8), pp.1798-1828.

Bhat, N.V. and McAvoy, T.J., 1992. Determining model structure for neural models by network stripping. Computers & chemical engineering, 16(4), pp.271-281.

Bishop, C.M., 1995. Neural networks for pattern recognition. Oxford university press.

Bourlard, H. and Kamp, Y., 1988. Auto-association by multilayer perceptrons and singular value decomposition. Biological cybernetics, 59(4), pp.291-294.

Bouvrie, J., 2006. Notes on convolutional neural networks.

Caruana, R., Lawrence, S. and Giles, L., 2001. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. Advances in neural information processing systems, pp.402-408.

Chairez, I., Poznyak, A. and Poznyak, T., 2007. Reconstruction of dynamics of aqueous phenols and their products formation in ozonation using differential neural network observers. Industrial & engineering chemistry research, 46(18), pp.5855-5866.

Chen, C.T. and Chang, W.D., 1996. A feedforward neural network with function shape autotuning. Neural networks, 9(4), pp.627-641.

Chen, J. and Patton, R.J., 2012. Robust model-based fault diagnosis for dynamic systems (Vol. 3). Springer Science & Business Media.

Chiang, L., Lu, B. and Castillo, I., 2017. Big data analytics in chemical engineering. Annual review of chemical and biomolecular engineering, 8, pp.63-85.

Chiang, L.H., Russell, E.L. and Braatz, R.D., 2000. Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis. Chemometrics and intelligent laboratory systems, 50(2), pp.243-252.

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

Cho, J.H., Lee, J.M., Choi, S.W., Lee, D. and Lee, I.B., 2005. Fault identification for process monitoring using kernel principal component analysis. Chemical engineering science, 60(1), pp.279-288.

Clevert, D.A., Unterthiner, T. and Hochreiter, S., 2015. Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289.

Cristianini, N. and Shawe-Taylor, J., 2000. An introduction to support vector machines and other kernel-based learning methods. Cambridge university press.

Damour, C., Benne, M., Boillereaux, L., Grondin-Perez, B. and Chabriat, J.P., 2010. NMPC of an industrial crystallization process using model-based observers. Journal of Industrial and Engineering Chemistry, 16(5), pp.708-716.

Dash, S., Rengaswamy, R. and Venkatasubramanian, V., 2003. Fuzzy-logic based trend classification for fault diagnosis of chemical processes. Computers & Chemical Engineering, 27(3), pp.347-362.

De Assis, A.J. and Maciel Filho, R., 2000. Soft sensors development for on-line bioreactor state estimation. Computers & Chemical Engineering, 24(2-7), pp.1099-1103.

De Silva, V. and Tenenbaum, J.B., 2003. Global versus local methods in nonlinear dimensionality reduction. Advances in neural information processing systems, pp.721-728.

Du, Y.G., del Villar, R. and Thibault, J., 1997. Neural net-based softsensor for dynamic particle size estimation in grinding circuits. International Journal of Mineral Processing, 52(2-3), pp.121-135.

E. J. Wegman and J. Shen, "Three-dimensional Andrews plots and the grand tour," Computing Science and Statistics, 25, 284–288, 1993.

Efron, B., & Tibshirani, R. (1993). An introduction to bootstrap. London: Chapman and Hall

Famili, A., Shen, W.M., Weber, R. and Simoudis, E., 1997. Data preprocessing and intelligent data analysis. Intelligent data analysis, 1(1), pp.3-23.

Frank, P.M., Ding, S.X. and Marcu, T., 2000. Model-based fault diagnosis in technical processes. Transactions of the Institute of Measurement and Control, 22(1), pp.57-101.

Garcfa, E. A., Frank, P. M. ,1996. On the relationship between observer and parameter identification based approaches to fault detection. IFAC Proceedings Volumes, 29(1), 6349-6353.

Gao, Z., Cecati, C. and Ding, S.X., 2015. A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches. IEEE transactions on industrial electronics, 62(6), pp.3757-3767.

Gao, Z., Cecati, C. and Ding, S.X., 2015. A survey of fault diagnosis and fault-tolerant techniques—Part II: Fault diagnosis with knowledge-based and hybrid/active-based approaches. IEEE Trans. Ind. Electron, 62(6), pp.3768-3774.

Ge, Z. and Song, Z., 2007. Process monitoring based on independent component analysis− principal component analysis (ICA− PCA) and similarity factors. Industrial & Engineering Chemistry Research, 46(7), pp.2054-2063.

Ge, Z., Song, Z. and Gao, F., 2013. Review of recent research on data-based process monitoring. Industrial & Engineering Chemistry Research, 52(10), pp.3543-3562.

Ge, Z., Song, Z., Ding, S.X. and Huang, B., 2017. Data mining and analytics in the process industry: The role of machine learning. Ieee Access, 5, pp.20590-20616.

Ge, Z., Yang, C. and Song, Z., 2009. Improved kernel PCA-based monitoring approach for nonlinear processes. Chemical Engineering Science, 64(9), pp.2245-2255.

Gertler, J. ,1995. Diagnosing parametric faults: from parameter estimation to parity relations. In Proceedings of 1995 American Control Conference-ACC'95 (Vol. 3, pp. 1615-1620). IEEE.

Germain, M., Gregor, K., Murray, I. and Larochelle, H., 2015, June. Made: Masked autoencoder for distribution estimation. In International Conference on Machine Learning (pp. 881-889). PMLR.

Gulcehre, C., Moczulski, M., Denil, M. and Bengio, Y., 2016, June. Noisy activation functions. In International conference on machine learning (pp. 3059-3068). PMLR.

Hagiwara, K. and Kuno, K., 2000, July. Regularization learning and early stopping in linear networks. In Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium (Vol. 4, pp. 511-516). IEEE.

Hastie, T., Tibshirani, R. and Friedman, J., 2009. The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

He, K., Zhang, X., Ren, S. and Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision (pp. 1026-1034).

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Himmelblau, D.M., 2008. Accounts of experiences in the application of artificial neural networks in chemical engineering. Industrial & Engineering Chemistry Research, 47(16), pp.5782-5796.

Hinton, G.E. and Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. science, 313(5786), pp.504-507.

Hinton, G.E. and Zemel, R.S., 1994. Autoencoders, minimum description length, and Helmholtz free energy. Advances in neural information processing systems, 6, pp.3-10.

Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

Horn, J., 2001. Trajectory tracking of a batch polymerization reactor based on input–output-linearization of a neural process model. Computers & Chemical Engineering, 25(11-12), pp.1561-1567.

Huang, J. and Yan, X., 2017. Quality relevant and independent two block monitoring based on mutual information and KPCA. IEEE Transactions on Industrial Electronics, 64(8), pp.6518-6527.

Isermann, R., 1997. Supervision, fault-detection and fault-diagnosis methods—an introduction. Control engineering practice, 5(5), pp.639-652.

Isermann, R., 1998. On fuzzy logic applications for automatic control, supervision, and fault diagnosis. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 28(2), pp.221-235.

Jiang, Q., Yan, X. and Huang, B., 2015. Performance-driven distributed PCA process monitoring based on fault-relevant variable selection and Bayesian inference. IEEE Transactions on Industrial Electronics, 63(1), pp.377-386.

Jiang, Q., Yan, X. and Huang, B., 2019. Review and perspectives of data-driven distributed monitoring for industrial plant-wide processes. Industrial & Engineering Chemistry Research, 58(29), pp.12899-12912.

Joe Qin, S., 2003. Statistical process monitoring: basics and beyond. Journal of Chemometrics: A Journal of the Chemometrics Society, 17(8‐9), pp.480-502.

Kadlec, P., Gabrys, B. and Strandt, S., 2009. Data-driven soft sensors in the process industry. Computers & chemical engineering, 33(4), pp.795-814.

Kalogirou, S.A., 2003. Artificial intelligence for the modeling and control of combustion processes: a review. Progress in energy and combustion science, 29(6), pp.515-566.

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

Katare, S., Bhan, A., Caruthers, J.M., Delgass, W.N. and Venkatasubramanian, V., 2004. A hybrid genetic algorithm for efficient parameter estimation of large kinetic models. Computers & chemical engineering, 28(12), pp.2569-2581.

Kotsiantis, S.B., Zaharakis, I. and Pintelas, P., 2007. Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160(1), pp.3-24.

Kourti, T. and MacGregor, J.F., 1996. Multivariate SPC methods for process and product monitoring. Journal of quality technology, 28(4), pp.409-428.

Kramer, M.A., 1991. Nonlinear principal component analysis using autoassociative neural networks. AIChE journal, 37(2), pp.233-243.

Kresta, J.V., MacGregor, J.F. and Marlin, T.E., 1991. Multivariate statistical monitoring of process operating performance. The Canadian journal of chemical engineering, 69(1), pp.35-47.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, pp.1097-1105.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, pp.1097-1105.

Lashkarbolooki, M., Vaferi, B. and Mowla, D., 2012. Using artificial neural network to predict the pressure drop in a rotating packed bed. Separation Science and Technology, 47(16), pp.2450-2459.

LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553), pp.436-444.

LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), pp.2278-2324.

Lee, J.M., Qin, S.J. and Lee, I.B., 2006. Fault detection and diagnosis based on modified independent component analysis. AIChE journal, 52(10), pp.3501-3514.

Leong, W.C., Bahadori, A., Zhang, J. and Ahmad, Z., 2019. Prediction of water quality index (WQI) using support vector machine (SVM) and least square-support vector machine (LS-SVM). International Journal of River Basin Management, pp.1-8.

Li, C., Ye, H., Wang, G. and Zhang, J., 2005. A recursive nonlinear PLS algorithm for adaptive nonlinear process modeling. Chemical Engineering & Technology: Industrial Chemistry - Plant Equipment - Process Engineering - Biotechnology, 28(2), pp.141-152.

Li, G., Qin, S.J. and Zhou, D., 2010. Geometric properties of partial least squares for process monitoring. Automatica, 46(1), pp.204-210.

Maas, A.L., Hannun, A.Y. and Ng, A.Y., 2013, June. Rectifier nonlinearities improve neural network acoustic models. In Proc. icml (Vol. 30, No. 1, p. 3).

Magni, J. F., & Mouyon, P. ,1994. On residual generation by observer and parity space approaches. IEEE Transactions on Automatic Control, 39(2), 441-447.

Maldonado, S., Weber, R. and Famili, F., 2014. Feature selection for high-dimensional class-imbalanced data sets using support vector machines. Information sciences, 286, pp.228-246.

Maltarollo, V.G., Honório, K.M. and da Silva, A.B.F., 2013. Applications of artificial neural networks in chemical problems. Artificial neural networks-architectures and applications, pp.203-223.

Mao, K.Z., 2004. Orthogonal forward selection and backward elimination algorithms for feature subset selection. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 34(1), pp.629-634.

Martin, E.B., Morris, A.J. and Zhang, J., 1996. Process performance monitoring using multivariate statistical process control. IEE Proceedings-Control Theory and Applications, 143(2), pp.132-144.

Maurya, M.R., Paritosh, P.K., Rengaswamy, R. and Venkatasubramanian, V., 2010. A framework for on-line trend extraction and fault diagnosis. Engineering Applications of Artificial Intelligence, 23(6), pp.950-960.

Maurya, M.R., Rengaswamy, R. and Venkatasubramanian, V., 2007. Fault diagnosis using dynamic trend analysis: A review and recent developments. Engineering Applications of artificial intelligence, 20(2), pp.133-146.

Mc Loone, S. and Irwin, G., 2001. Improving neural network training solutions using regularisation. Neurocomputing, 37(1-4), pp.71-90.

Mesbah, A., Huesman, A.E., Kramer, H.J. and Van den Hof, P.M., 2011. A comparison of nonlinear observers for output feedback model-based control of seeded batch crystallization processes. Journal of Process Control, 21(4), pp.652-666.

Miletic, I., Quinn, S., Dudzic, M., Vaculik, V. and Champagne, M., 2004. An industrial perspective on implementing on-line applications of multivariate statistics. Journal of Process Control, 14(8), pp.821-836.

Muralidharan, V. and Sugumaran, V., 2012. A comparative study of Naïve Bayes classifier and Bayes net classifier for fault diagnosis of monoblock centrifugal pump using wavelet analysis. Applied Soft Computing, 12(8), pp.2023-2029.

N. H. Spencer, "Investigating data with Andrews plots," Social Science Computer Review, 21(2), 244–249, 2003.

Negnevitsky, M., 2005. Artificial intelligence: a guide to intelligent systems. Pearson education.

Patton, R.J., 1994. Robust model-based fault diagnosis: the state of the art. IFAC Proceedings Volumes, 27(5), pp.1-24.

Patton, R.J., Frank, P.M. and Clark, R.N. eds., 2013. Issues of fault diagnosis for dynamic systems. Springer Science & Business Media.

Paul, A. and Venkatasubramanian, S., 2014. Why does deep learning work?-a perspective from group theory. arXiv preprint arXiv:1412.6621.

Pawlak, Z., 1982. Rough sets. International journal of computer & information sciences, 11(5), pp.341-356.

Porru, G., Aragonese, C., Baratti, R. and Servida, A., 2000. Monitoring of a CO oxidation reactor through a grey model-based EKF observer. Chemical engineering science, 55(2), pp.331-338.

Prakash, J. and Senthil, R., 2008. Design of observer based nonlinear model predictive controller for a continuous stirred tank reactor. Journal of Process control, 18(5), pp.504-514.

Priddy, K.L. and Keller, P.E., 2005. Artificial neural networks: an introduction (Vol. 68). SPIE press.

Qin, S.J. and Chiang, L.H., 2019. Advances and opportunities in machine learning for process data analytics. Computers & Chemical Engineering, 126, pp.465-473.

Qin, S.J., 2012. Survey on data-driven industrial process monitoring and diagnosis. Annual reviews in control, 36(2), pp.220-234.

Qin, S.J., 2014. Process data analytics in the era of big data.

R. N. Khattree and D. N. Naik, "Andrews plots for multivariate data: Some new suggestions and applications," Journal of Statistical Planning and Inference, 100(2), 411–425, 2002.

Raich, A. and Çinar, A., 1997. Diagnosis of process disturbances by statistical distance and angle measures. Computers & chemical engineering, 21(6), pp.661-673.

Ringnér, M., 2008. What is principal component analysis?. Nature biotechnology, 26(3), pp.303-304.

Ripley, B.D., 2007. Pattern recognition and neural networks. Cambridge university press.

Roweis, S.T. and Saul, L.K., 2000. Nonlinear dimensionality reduction by locally linear embedding. science, 290(5500), pp.2323-2326.

Russell, E.L., Chiang, L.H. and Braatz, R.D., 2012. Data-driven methods for fault detection and diagnosis in chemical processes. Springer Science & Business Media.#

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 Sensing. 9(5), 427, 2017.

S. R. Kulkarni and S. R. Paranjape, “Use of Andrews' function plot technique to construct control curves for multivariate process,” Commun. Statist. Theor. Meth, 13(20), 2511–2533, 1984.

Sakthivel, N.R., Nair, B.B., Elangovan, M., Sugumaran, V. and Saravanmurugan, S., 2014. Comparison of dimensionality reduction techniques for the fault diagnosis of mono block centrifugal pump using vibration signals. Engineering Science and Technology, an International Journal, 17(1), pp.30-38.

Sarle, W.S., 1994. Neural networks and statistical models.

Scholkopf, B. and Smola, A.J., 2018. Learning with kernels: support vector machines, regularization, optimization, and beyond. Adaptive Computation and Machine Learning series.

Senthil, R., Janarthanan, K. and Prakash, J., 2006. Nonlinear state estimation using fuzzy Kalman filter. Industrial & engineering chemistry research, 45(25), pp.8678-8688.

Shah, S.L., Patwardhan, R. and Huang, B., 2002. Multivariate controller performance analysis: methods, applications and challenges. In AIChE symposium series (pp. 190-207). New York; American Institute of Chemical Engineers; 1998.

Shang, C., Yang, F., Huang, D. and Lyu, W., 2014. Data-driven soft sensor development based on deep learning technique. Journal of Process Control, 24(3), pp.223-233.

Shang, W., Sohn, K., Almeida, D. and Lee, H., 2016, June. Understanding and improving convolutional neural networks via concatenated rectified linear units. In international conference on machine learning (pp. 2217-2225). PMLR.

Sharma, R., Singh, K., Singhal, D. and Ghosh, R., 2004. Neural network applications for detecting process faults in packed towers. Chemical engineering and processing: process intensification, 43(7), pp.841-847.

Shlens, J., 2014. A tutorial on principal component analysis. arXiv preprint arXiv:1404.1100.

Simani, S., Fantuzzi, C. and Patton, R.J., 2003. Model-based fault diagnosis techniques. In Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques (pp. 19-60). Springer, London.

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

Stephanopoulos, G. and Han, C., 1996. Intelligent systems in process engineering: A review. Computers & Chemical Engineering, 20(6-7), pp.743-791.

Sutter, J.M. and Kalivas, J.H., 1993. Comparison of forward selection, backward elimination, and generalized simulated annealing for variable selection. Microchemical journal, 47(1-2), pp.60-66.

Velardi, S.A., Hammouri, H. and Barresi, A.A., 2009. In-line monitoring of the primary drying phase of the freeze-drying process in vial by means of a Kalman filter based observer. Chemical Engineering Research and Design, 87(10), pp.1409-1419.

Venkatasubramanian, V. and Chan, K., 1989. A neural network methodology for process fault diagnosis. AIChE Journal, 35(12), pp.1993-2002.

Venkatasubramanian, V., 2019. The promise of artificial intelligence in chemical engineering: Is it here, finally?. AIChE Journal, 65(2), pp.466-478.

Venkatasubramanian, V., 2019. The promise of artificial intelligence in chemical engineering: Is it here, finally?. AIChE Journal, 65(2), pp.466-478.

Venkatasubramanian, V., Rengaswamy, R. and Kavuri, S.N., 2003. A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies. Computers & chemical engineering, 27(3), pp.313-326.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N. and Yin, K., 2003. A review of process fault detection and diagnosis: Part III: Process history based methods. Computers & chemical engineering, 27(3), pp.327-346.

Venkatasubramanian, V., Rengaswamy, R., Yin, K. and Kavuri, S.N., 2003. A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. Computers & chemical engineering, 27(3), pp.293-311.

Wang, D., 2016. K-nearest neighbors based methods for identification of different gear crack levels under different motor speeds and loads: Revisited. Mechanical Systems and Signal Processing, 70, pp.201-208.

Wang, Y., Si, Y., Huang, B. and Lou, Z., 2018. Survey on the theoretical research and engineering applications of multivariate statistics process monitoring algorithms: 2008–2017. The Canadian Journal of Chemical Engineering, 96(10), pp.2073-2085.

Watanabe, K., Matsuura, I., Abe, M., Kubota, M. and Himmelblau, D.M., 1989. Incipient fault diagnosis of chemical processes via artificial neural networks. AIChE journal, 35(11), pp.1803-1812.

Wen, L., Li, X., Gao, L. and Zhang, Y., 2017. A new convolutional neural network-based data-driven fault diagnosis method. IEEE Transactions on Industrial Electronics, 65(7), pp.5990-5998.

Wise, B.M., Ricker, N.L., Veltkamp, D.F. and Kowalski, B.R., 1990. A theoretical basis for the use of principal component models for monitoring multivariate processes. Process control and quality, 1(1), pp.41-51.

Wiskott, L. and Sejnowski, T.J., 2002. Slow feature analysis: Unsupervised learning of invariances. Neural computation, 14(4), pp.715-770.

Wu, H. and Zhao, J., 2018. Deep convolutional neural network model based chemical process fault diagnosis. Computers & chemical engineering, 115, pp.185-197.

Xu, B., Wang, N., Chen, T. and Li, M., 2015. Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853.

Zarei, J. and Poshtan, J., 2010. Design of nonlinear unknown input observer for process fault detection. Industrial & Engineering Chemistry Research, 49(22), pp.11443-11452.

Zhang, J. and Morris, A.J., 1998. A sequential learning approach for single hidden layer neural networks. Neural networks, 11(1), pp.65-80.

Zhang, J. and Roberts, P.D., 1991. Process fault diagnosis with diagnostic rules based on structural decomposition. Journal of Process Control, 1(5), pp.259-269.

Zhang, J. and Roberts, P.D., 1992. On-line process fault diagnosis using neural network techniques. Transactions of the Institute of Measurement and Control, 14(4), pp.179-188.

Zhang, J., 1999. Developing robust non-linear models through bootstrap aggregated neural networks. Neurocomputing, 25(1-3), pp.93-113.

Zhang, J., 1999. Inferential estimation of polymer quality using bootstrap aggregated neural networks. Neural networks, 12(6), pp.927-938.

Zhang, J., 2001, June. Inferential feedback control of distillation composition based on PCR and PLS models. In Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148) (Vol. 2, pp. 1196-1201). IEEE.

Zhang, J., 2001. Developing robust neural network models by using both dynamic and static process operating data. Industrial & engineering chemistry research, 40(1), pp.234-241.

Zhang, J., 2006. Improved on-line process fault diagnosis through information fusion in multiple neural networks. Computers & chemical engineering, 30(3), pp.558-571.

Zhang, J., Expert Systems in On-line Process Control and Fault Diagnosis, PhD Thesis, City University, London, 1991.

Zhang, J., Martin, E.B. and Morris, A.J., 1996. Fault-detection and diagnosis using multivariate statistical techniques. Chemical engineering research & design, 74(1), pp.89-96.

Zhang, J., Martin, E.B., Morris, A.J. and Kiparissides, C., 1997. Inferential estimation of polymer quality using stacked neural networks. Computers & Chemical Engineering, 21, pp.S1025-S1030.

Zhang, J., Roberts, P.D. and Ellis, J.E., 1990. Fault diagnosis of a mixing process using deep qualitative knowledge representation of physical behaviour. Journal of Intelligent and Robotic Systems, 3(2), pp.103-115.

Zhang, Z. and Friedrich, K., 2003. Artificial neural networks applied to polymer composites: a review. Composites Science and technology, 63(14), pp.2029-2044.

Zhao, S.J., Zhang, J. and Xu, Y.M., 2006. Performance monitoring of processes with multiple operating modes through multiple PLS models. Journal of process Control, 16(7), pp.763-772.

Zhao, S.J., Zhang, J., Xu, Y.M. and Xiong, Z.H., 2006. Nonlinear projection to latent structures method and its applications. Industrial & engineering chemistry research, 45(11), pp.3843-3852.

Zou, H. and Hastie, T., 2005. Regularization and variable selection via the elastic net. Journal of the royal statistical society: series B (statistical methodology), 67(2), pp.301-320.