

Chapter 8

Fault Identification Based on Local Feature Correlation



Industrial data variables show obvious high dimension and strong nonlinear correlation. Traditional multivariate statistical monitoring methods, such as PCA, PLS, CCA, and FDA, are only suitable for solving the high-dimensional data processing with linear correlation. The kernel mapping method is the most common technique to deal with the nonlinearity, which projects the original data in the low-dimensional space to the high-dimensional space through appropriate kernel functions so as to achieve the goal of linear separability in the new space. However, the space projection from the low dimension to the high dimension is contradictory to the actual requirement of dimensionality reduction of the data. So kernel-based method inevitably increases the complexity of data processing. For this reason, we have proposed another kind of nonlinear processing approach based on the manifold learning, a class of unsupervised model that seeks to describe data sets as low-dimensional manifold embedded in high-dimensional spaces. It characterizes the original data as a low-dimensional manifold to achieve the goal of nonlinear correlation processing. This strategy is consistent with the goal of dimensionality reduction. Furthermore, manifold learning fits the nonlinear correlation by means of piecewise linearization in an intuitive sense. It has significantly less complexity compared to the kernel mapping method.

This chapter carries out the pattern classification techniques for multivariate variables with strong nonlinear correlation and applies them to the fault identification of batch process. Two kinds of pattern classification methods are given in this chapter: (1) kernel exponential discriminant analysis (KEDA): this method addresses the nonlinear correlation properties among multi-variables at two levels, kernel mapping and exponential discrimination, respectively. It can significantly improve the classification accuracy compared with the traditional FDA method. (2) The fusion method is based on manifold learning and discriminant analysis: two different fusion strategies, local linear exponential discriminant analysis (LLEDA) and neighborhood-preserving embedding discriminant analysis (NPEDA), are given, respectively. Here

locally linear embedding (LLE) is a popular algorithm of manifold learning. They both combine the advantage of global discriminant analysis with the local structure preserving. LLEDA is a parallel strategy to find a trade-off projection vector between the local geometric structure preserving and the global data classification. NPEDA is a cascaded strategy whose dimensionality reduction process is implemented in two serial steps. The two methods emphasize the intrinsic structure of the data while utilizing the global discriminant information, so they have better classification than the traditional EDA method. Finally, a kind of hybrid fault diagnosis scheme is given for the complex industrial process, which consists of PCA-based fault detection, hierarchical clustering-based pre-diagnosis, and LLEDA-based final identification.

8.1 Fault Identification Based on Kernel Discriminant Exponent Analysis

8.1.1 Methodology of KEDA

The kernel exponent discriminant analysis (KEDA) is also a discriminative classification method, which aims to find a series of discriminant vectors that can transform the data into the kernel space and achieve the greatest separation between different types of data in the projection direction.

Consider the batch process data set with I batches, i.e.,

$$\mathbf{X}(k) = [\mathbf{X}^1(k), \mathbf{X}^2(k), \dots, \mathbf{X}^I(k)]^T,$$

where \mathbf{X}^i consists of n_i , $i = 1, \dots, I$ row vectors, and each row vector is a sample vector $\mathbf{X}_j^i(k)$, $j = 1, \dots, n_i$ acquired at time k and batch i . According to the analysis from equations (7.1)–(7.9) in Sect. 7.1.1, the optimization function of kernel Fisher discrimination analysis (KFDA) is given as follows,

$$\begin{aligned} \max J(\boldsymbol{\alpha}) &= \frac{\text{tr}(\boldsymbol{\alpha}^T \mathbf{K}_b \boldsymbol{\alpha})}{\text{tr}(\boldsymbol{\alpha}^T \mathbf{K}_w \boldsymbol{\alpha})} \\ &= \frac{\text{tr}(\boldsymbol{\alpha}^T (\mathbf{V}_b \boldsymbol{\Lambda}_b \mathbf{V}_b^T) \boldsymbol{\alpha})}{\text{tr}(\boldsymbol{\alpha}^T (\mathbf{V}_w \boldsymbol{\Lambda}_w \mathbf{V}_w^T) \boldsymbol{\alpha})}, \end{aligned} \quad (8.1)$$

where $\mathbf{K}_b = \mathbf{V}_b \boldsymbol{\Lambda}_b \mathbf{V}_b^T$ and $\mathbf{K}_w = \mathbf{V}_w \boldsymbol{\Lambda}_w \mathbf{V}_w^T$ are eigenvalue decompositions of between-class and within-class scatter matrices, respectively. $\boldsymbol{\Lambda}_b = \text{diag}(\lambda_{b1}, \lambda_{b2}, \dots, \lambda_{bn})$, and $\boldsymbol{\Lambda}_w = \text{diag}(\lambda_{w1}, \lambda_{w2}, \dots, \lambda_{wn})$ are the eigenvalues, $\mathbf{V}_b = (v_{b1}, v_{b2}, \dots, v_{bn})$, and $\mathbf{V}_w = (v_{w1}, v_{w2}, \dots, v_{wn})$ are the corresponding eigenvectors. The basic objective is to maximize the between-class distance and minimize the with-class distance simultaneously during the projection.

In order to improve the discrimination accuracy further, the discriminant function (8.1) is exponentiated:

$$\begin{aligned} \max J(\alpha) &= \frac{\text{tr}(\alpha^T (\mathbf{V}_b \exp(\mathbf{A}_b) \mathbf{V}_b^T) \alpha)}{\text{tr}(\alpha^T (\mathbf{V}_w \exp(\mathbf{A}_w) \mathbf{V}_w^T) \alpha)} \\ &= \frac{\text{tr}(\alpha^T \exp(\mathbf{K}_b) \alpha)}{\text{tr}(\alpha^T \exp(\mathbf{K}_w) \alpha)}. \end{aligned} \quad (8.2)$$

The optimization problem (8.2) is transferred to the following generalized eigenvalue problem:

$$\begin{aligned} \exp(\mathbf{K}_b) \alpha &= \Lambda \exp(\mathbf{K}_w) \alpha \\ \text{or} & \\ \exp(\mathbf{K}_w)^{-1} \exp(\mathbf{K}_b) \alpha &= \Lambda \alpha, \end{aligned} \quad (8.3)$$

where Λ is the eigenvalue and α is the corresponding eigenvector. The discriminant vectors are calculated from (8.3). Usually, the first two vectors, optimal, and suboptimal ones are selected for dimensionality reduction.

The within-class and between-class scatter matrices are exponentiated in KEDA. Consider the general property of exponential function, $e^x > x$ for any $x > 0$, so the scatter matrix of KEDA is greater than KFDA. It means KEDA has better discriminatory capability than KFDA. Moreover, if the amount of sample data is less than the number of variables, the rank of within-class scatter matrix is less than the dimension of variables. Now the within-class scatter matrix is singular, and its inversion does not exist. But both the within-class and between-class scatter matrices are exponentiated in KEDA. The exponentiated matrices must be full rank, so the singular problem caused by small samples is solved. Thus from this view, the KEDA method not only solves the small sample problem, but also efficiently classifies the sample data into different categories, which helps to improve the classification accuracy.

Let's consider the nonlinear mapping $\Phi(\mathbf{x}_k^i)$ of original sample \mathbf{x}_k^i and project it to the optimal and suboptimal discriminant directions, respectively. Then the eigenvalues $\mathbf{T}_i(k) = [T_{ik}^1, T_{ik}^2]^T$ and T_{ik}^2 are obtained, which represent the projection values in the optimal and suboptimal discriminant directions. Usually, the data in the same class shows the similar project eigenvalues in the direction of selected discrimination vectors. If the test data matches with the known fault class, it has maximum projection eigenvalue under this model, obviously nonzero. If the test data does not match with this class, the eigenvalue is small even close to zero. It is unrealistic to judge the data type simply based on the magnitude of eigenvalues. So difference degree D between two projection values $\mathbf{T}_i(k)$ and $\mathbf{T}_j(k)$ is defined as follows:

$$D_{i,j}(k) = 1 - \frac{(\mathbf{T}_i(k))^T \mathbf{T}_j(k)}{\|\mathbf{T}_i(k)\|_2 \|\mathbf{T}_j(k)\|_2}. \quad (8.4)$$

The smaller the difference D , the higher the model matched.

The KEDA-based fault classification and identification process for batch process is given as follows:

Step 1: Data preprocess. The three-dimensional data set $X(L \times J \times K)$ is batch-wise unfolded into two-dimensional data $X(LK \times J)$, normalized along the time in the batch cycle and variable-wise re-arranged.

Step 2: Kernel projection. The original data X is mapping to a high-dimensional feature space via a nonlinear kernel function, and the kernel sampling data $\xi_j^i = [K(x_1, x_j^i), K(x_2, x_j^i), \dots, K(x_n, x_j^i)]^T$ are obtained.

Step 3: KEDA modeling. The optimal kernel discriminant vectors are solved from the discriminant function equation (8.3). Project the sample data ξ_j^i to the selected kernel discriminant vectors and calculate the corresponding eigenvalues $T_i(k)$.

Step 4: Test calculation. The test sample $x_{j,new}(k)$ is collected and the corresponding eigenvalues $T_{i,new}(k)$ according to the known S classes model are calculated, respectively.

Step 5: Fault identification. The class of test data can be determined by calculating the difference degree between test sample and trained data (8.4).

8.1.2 Simulation Experiment

The proposed KEDA was used for fault identification in the penicillin fermentation process mentioned in Sect. 4.2. Here nine process variables were considered for monitoring and three faults are shown in Table 8.1. The data were generated by the penicillin simulator when the amplitude and time of fault are changed. A total of 40 batches were selected as the training data set: 10 batches for normal and known 3 faults. The KEDA method with Gaussian kernel function was used to find the optimal discriminant vectors for each type of model, and four different models were obtained.

Experiment 1: Data classification Figures 8.1, 8.2, 8.3, and 8.4 show the classification comparison of KFDA and KEDA for penicillin data: normal data and three types of fault data. When the test data are different from the known four types, the projections are also separated from each other. But the KFDA shows weaker classification performance: some faults are closer together and the boundaries are not easily distinguishable, such as fault 3 data (red ★) and test fault data (black ■) in Figs. 8.1 and 8.3. However, the KEDA works better for classifying these data, and

Table 8.1 Description of the fault type of penicillin process

No.	Faults	Types
1	Bottom logistics decline	Step
2	Decreased power of the mixer	Step
3	Decreased airflow	Step

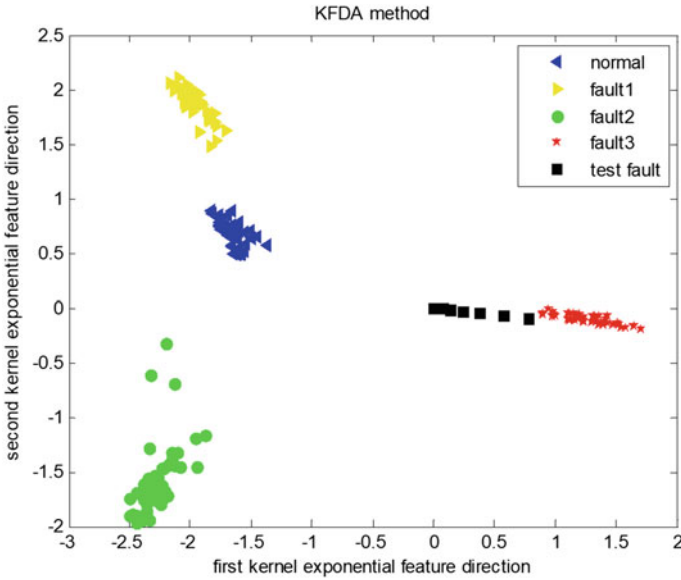


Fig. 8.1 Two-dimensional classification visualization: KFDA method

the red and black parts are classified clearly in Figs. 8.2 and 8.4. These plots show that the between-class and within-class distances have increased for different types of data in KEDA, but the between-class distance has increased by a larger magnitude than the within-class distance. So the different types of data can be better separated.

Experiment 2: Fault-type identification Let’s consider the testing data set, which also consists of the four types of data and an unknown fault data. Table 8.2 gives the eigenvalues of the four testing data calculated based on the KEDA model of fault 2. The eigenvalues are obtained by projecting the testing data to the selected optimal discriminant directions. If there is a large difference between the testing data and the training data, then the value of $\| \mathbf{u} - \mathbf{v} \|^2$ is large and the exponentiated Gaussian kernel function, $K(\mathbf{u}, \mathbf{v}) = \exp(-\| \mathbf{u} - \mathbf{v} \|^2 / (2\sigma)^2)$, is almost close to zero. However, sometimes the fault occurrence eigenvalues are not close to zero, as shown in Table 8.2. At this case, the eigenvalues of the test data need to be analyzed further.

It is impossible to show the values at any sampling instance, so we further analyze the statistical characterizes of eigenvalues projected to the optimal discrimination direction of known model. If the eigenvalue of testing data follows a normal distribution in a model, the testing data belongs to this kind of model. Conversely, if the eigenvalue does not follow a normal distribution, it means that the testing data does not match with this model. Figures 8.5, 8.6, and 8.7 give the statistical analysis of the testing data (normal, faults 1 and 3) in the known fault 3 model. The eigenvalue of fault 3 follows a normal distribution in the fault 3 model, while the normal data or fault 1 data do not follow a normal distribution.

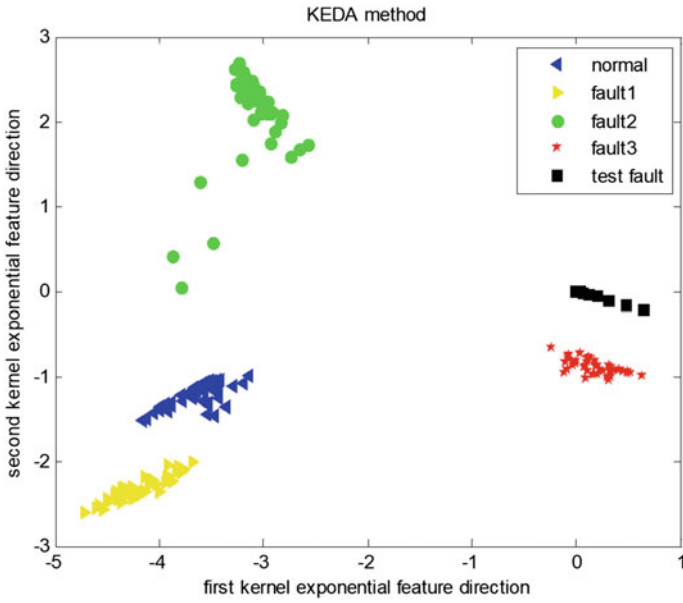


Fig. 8.2 Two-dimensional classification visualization: KEDA method

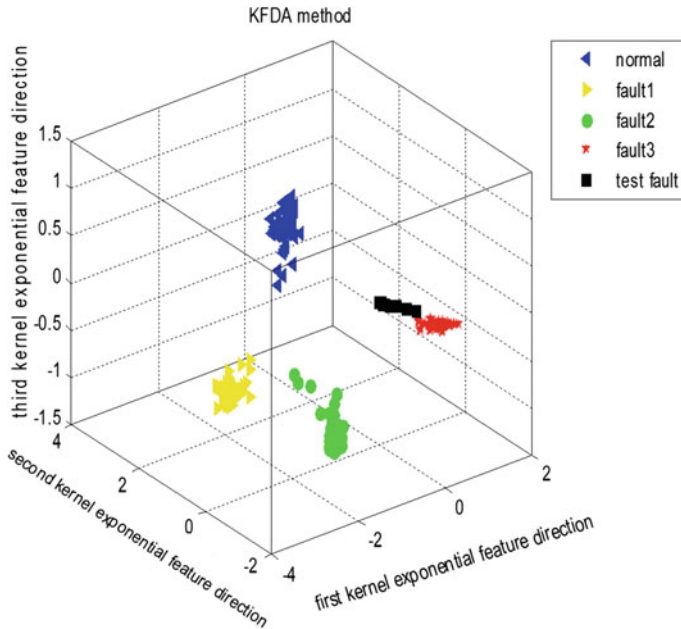


Fig. 8.3 Three-dimensional classification visualization: KFDA method

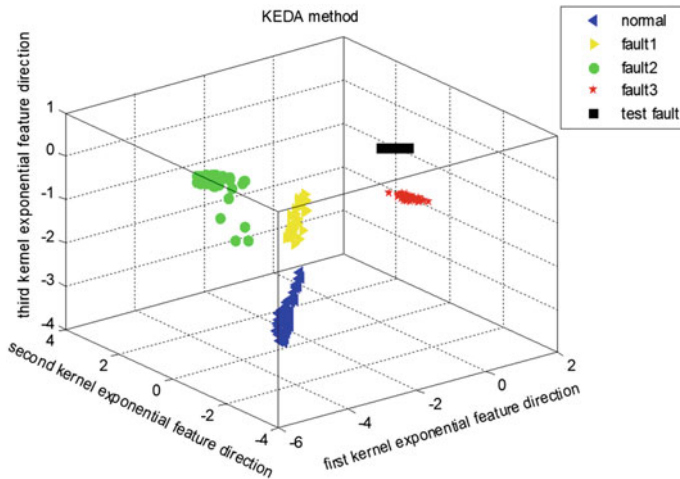


Fig. 8.4 Three-dimensional classification visualization: KEDA method

Table 8.2 The eigenvalues of test data in fault 2 model

Sampling instant	Eigenvalues of test data (T_k)				
	Normal	Fault 1	Fault 2	Fault 3	New fault
53	0.148	-0.148	-0.203	0	0
54	0.194	-0.194	0.0090	0	0
55	0.448	0	0.1660	0	0
56	0.187	0	0.1020	0	0
⋮	⋮	⋮	⋮	⋮	⋮
79	0.079	0	-0.024	0	0
80	0.103	0	-0.075	0	0
81	0.108	0	-0.084	0	0
82	0.041	0	-0.059	0	0

Moreover, the difference degree between test data and known model is used to determine the type of fault. The results are shown in Table 8.3. Since some of the test data have zero eigenvalues in the known model, and the denominators in the definition (8.4) are zero, the different degree cannot be calculated and expressed as “-”. The difference degree is small if the test data belongs to the known type model, and large if the test data does not belong to the model. It is found that the test data has the smallest different degree in the matching model.

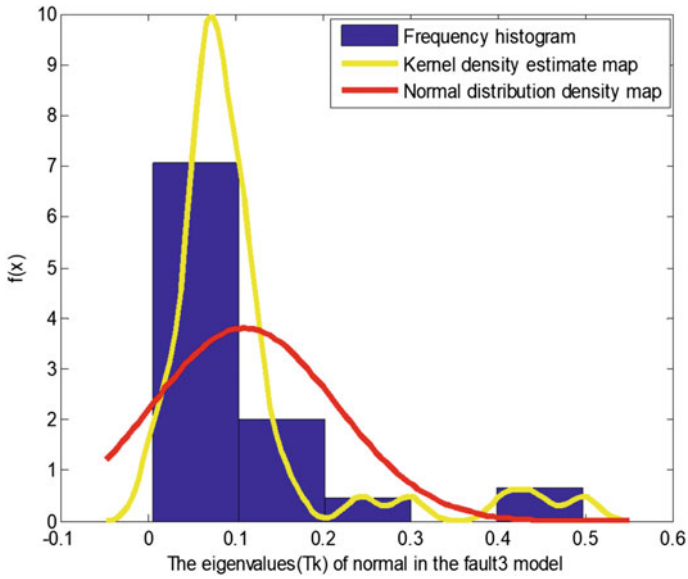


Fig. 8.5 The eigenvalues of test normal data in fault 3 model

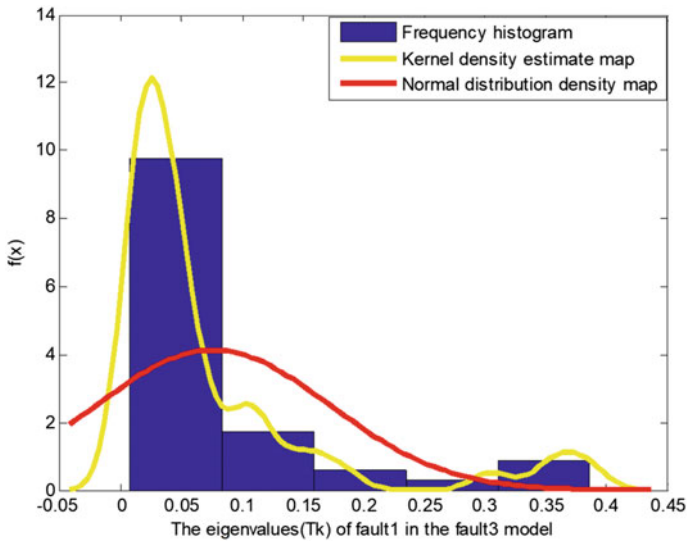


Fig. 8.6 The eigenvalues of test fault 1 data in fault 3 model

Fig. 8.7 The eigenvalues of test fault 3 data in fault 3 model

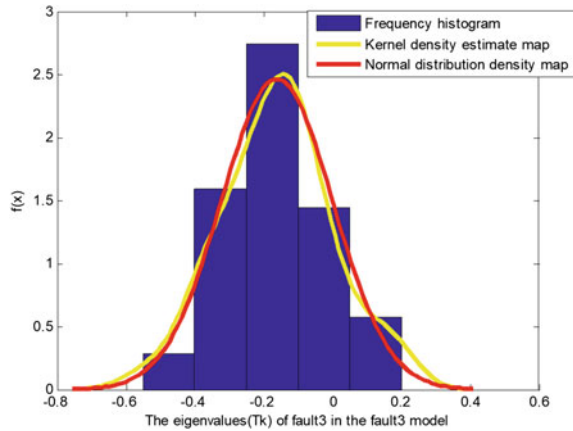


Table 8.3 The difference degree of test data in different models

Type of test data	Normal model	Fault 1 model	Fault 2 model	Fault 3 model
Normal	0.516679	0.669503	1.448272	1.630094
Fault 1	–	0.223966	–	1.578313
Fault 2	–	0.632128	0.550645	1.194915
Fault 3	–	–	–	0.553784
New fault	1.120218	–	–	1.137496

8.2 Fault Identification Based on LLE and EDA

The new dimensionality reduction approach based on the combination of EDA and LLE is proposed with two different combination performances, Local Linear Exponential Discriminant Analysis (LLEDA) and Neighborhood-Preserving Embedding Discriminant Analysis (NPEDA). This fusion idea combines the global discriminant analysis with local structure preservation during the dimensionality reduction process. LLEDA and NPEDA are solved by different optimization objectives, respectively, and the corresponding maximum values are derived to reduce the computational complexity. They both exhibit the good local preservation and global discrimination capabilities. The nonlinear analytics is transformed into an equivalent neighborhood holding problem based on the idea of piecewise linearization.

The main difference between the two methods is that LLEDA is a parallel strategy whereas NPEDA is a cascading strategy. LLEDA focuses on the global supervised discrimination balanced with local nonlinear dimensionality reduction. It finds a balanced projection vector between the local geometry and the data classification and results in an optimal subspace projection of the samples. When faults are difficult to distinguish, LLEDA method can improve the identification rate by adjusting the trade-off parameter between the global index and the local index. NPEDA is a

cascading strategy where the dimensionality reduction process is implemented in two successive steps: the first aims at maintaining the local geometric relationships and reconstructing each sample point using a linear weighted combination of nearest neighbors, the second at performing discriminant analysis on the reconstructed sample.

8.2.1 Local Linear Exponential Discriminant Analysis

The basic idea of LLEDA is to project the samples into the optimal discriminant space while maintaining the local geometric structure of the original data. The schematic diagram is shown in Fig. 8.8. LLEDA combines the advantages of LLE and EDA, which extracts the global classification information while compressing the dimensionality of the feature space without destroying local relationships. It finds a balance between global supervised discrimination and local preservation of nonlinearity through an adjusted trade-off parameter.

Consider the original data being mapped into a hidden space F via function A . An explicit linear mapping from X to Y , $Y = A^T X$ is constructed to circumvent the out-of-sample problem. The original LLE problem is written as follows:

$$\begin{aligned} \min \varepsilon(Y) &= \sum_{j=1}^n \left| y_j - \sum_{r=1}^k W_{jr} y_{jr} \right|^2 = \| Y(I - W) \|^2 \\ &= \text{tr}(Y(I - W)(I - W)^T Y^T) \\ &= \text{tr}(A^T X M X^T A). \end{aligned} \quad (8.5)$$

The LLEDA problem is proposed with the following objective function:

$$\max J(A) = \frac{\text{tr}(A^T \exp(S_b) A)}{\text{tr}(A^T \exp(S_w) A)} - \mu \cdot \text{tr}(A^T X M X^T A), \quad (8.6)$$

where μ is a trade-off parameter that balances the intrinsic geometry and global discriminant information. In general, (8.6) is equivalently transformed into an optimization problem with constraint,

$$\begin{aligned} \max J(A) &= \text{tr}(A^T \exp(S_b) A) - \mu \cdot \text{tr}(A^T X M X^T A) \\ \text{s.t. } &A^T \exp(S_w) A = I, \end{aligned} \quad (8.7)$$

where $A = [a_1, a_2, \dots, a_n]$. (8.7) is solved by introducing the Lagrangian multiplier:

$$L_1(a_i) = a_i^T (\exp(S_b) - \mu X M X^T) a_i + \theta(1 - a_i^T \exp(S_w) a_i), \quad (8.8)$$

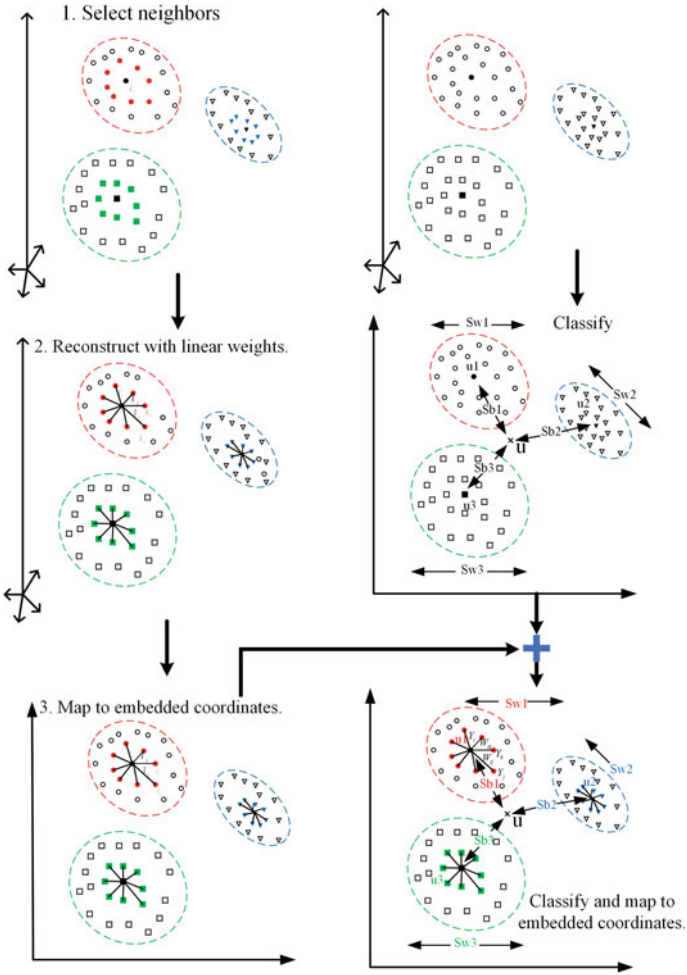


Fig. 8.8 The schematic diagram of LLEDA

where θ is Lagrangian multiplier. According to the zero gradient in $L_1(a_i)$ with respect to a_i , we have

$$\begin{aligned}
 (\exp(S_b) - \mu X M X^T) a_i &= \theta \exp(S_w) a_i \\
 \text{or} & \\
 (\exp(S_w)^{-1} (\exp(S_b) - \mu X M X^T) a_i &= \theta a_i,
 \end{aligned}
 \tag{8.9}$$

where θ is treated as a generalization eigenvalue. The discriminant matrix A is made up of the corresponding eigenvectors of the first d largest eigenvalues in (8.9).

8.2.2 Neighborhood-Preserving Embedding Discriminant Analysis

NPEDA is also to find a series of discriminative vectors and map the samples into a new space. The sample point is represented linearly by their neighbors to maintain the local geometry as much as possible during the projection process. The schematic diagram is shown in Fig. 8.9. NPEDA is a cascade strategy in which the dimensionality reduction process is divided into two successive steps, the first aiming at maintaining local geometric relationships and the second aiming at a discriminant analysis in which each sample point is reconstructed by a linearly weighted combination of its neighbors.

Rewrite the between-class scatter matrix S_b and the within-class scatter matrix S_w under the explicit linear mapping $Y = A^T X$:

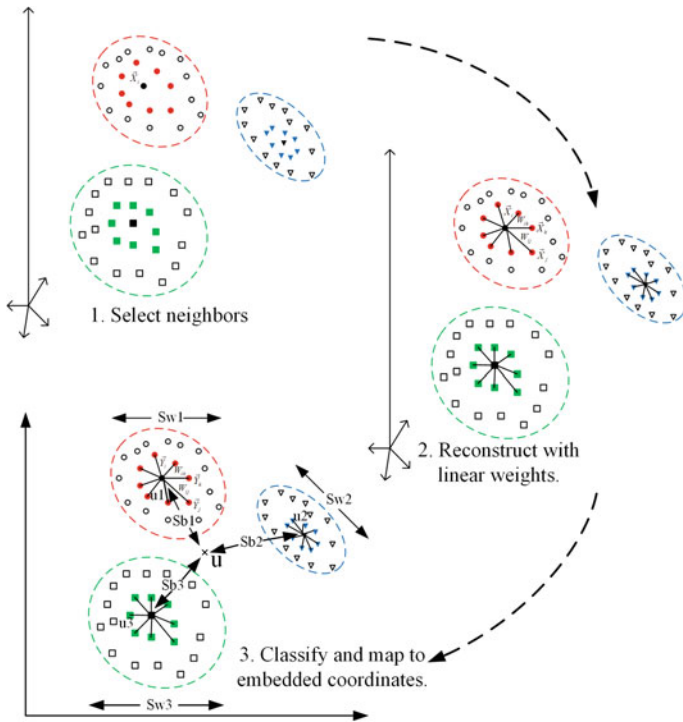


Fig. 8.9 The schematic diagram of NPEDA

$$\begin{aligned}
S_b &= \sum_{i=1}^c n_i (\bar{y}^i - \bar{y})^2 = \sum_{i=1}^c n_i (A^T \bar{x}^i - A^T \bar{x})^2 \\
&= A^T \left(\sum_{i=1}^c n_i (\bar{x}^i - \bar{x})(\bar{x}^i - \bar{x})^T \right) A \\
&= A^T \left(\sum_{i=1}^c \frac{1}{n_i} (\mathbf{x}_1^i + \cdots + \mathbf{x}_{n_i}^i) (\mathbf{x}_1^i + \cdots + \mathbf{x}_{n_i}^i)^T - 2n\bar{x}\bar{x}^T + n\bar{x}\bar{x}^T \right) A \\
&= A^T \left(\sum_{i=1}^c \sum_{j,k=1}^{n_i} \frac{1}{n_i} \mathbf{x}_j^i \mathbf{x}_k^{iT} - n_i \bar{x} \bar{x}^T \right) A \\
&= A^T (X B X^T - n\bar{x}\bar{x}^T) A \\
&= A^T X \left(B - \frac{1}{n} \mathbf{e} \mathbf{e}^T \right) X^T A,
\end{aligned} \tag{8.10}$$

where $\bar{x}^i = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{x}_j^i$, $\bar{y} = \frac{\sum_{i=1}^c n_i \bar{y}^i}{\sum_{i=1}^c n_i}$, $\bar{x} = \frac{\sum_{i=1}^c n_i \bar{x}^i}{\sum_{i=1}^c n_i} = \frac{1}{n} \sum_{i=1}^c n_i \bar{x}^i$; $\mathbf{e} = [1, 1, \dots, 1]^T$ with dimension n , and

$$B_{ij} = \begin{cases} \frac{1}{n_k} & \mathbf{x}_i \text{ and } \mathbf{x}_j \in k\text{-th class.} \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{aligned}
S_w &= \sum_{i=1}^c \sum_{j=1}^{n_i} (y_j^i - \bar{y}^i)^2 = \sum_{i=1}^c \sum_{j=1}^{n_i} (A^T \mathbf{x}_j^i - A^T \bar{x}^i)^2 \\
&= A^T \left(\sum_{i=1}^c \left(\sum_{j=1}^{n_i} (\mathbf{x}_j^i - \bar{x}^i) (\mathbf{x}_j^i - \bar{x}^i)^T \right) \right) A \\
&= A^T \left(\sum_{i=1}^c \left(\sum_{j=1}^{n_i} \mathbf{x}_j^i \mathbf{x}_j^{iT} - n_i \bar{x}^i \bar{x}^{iT} \right) \right) A \\
&= A^T \left(\sum_{i=1}^c \left(X_i X_i^T - \frac{1}{n_i} X_i (\mathbf{e}_i \mathbf{e}_i^T) X_i^T \right) \right) A \\
&= A^T \sum_{i=1}^c (X_i L_i X_i^T) A,
\end{aligned} \tag{8.11}$$

where $L_i = I - \frac{1}{n_i} \mathbf{e}_i \mathbf{e}_i^T$, I is unit matrix, and $\mathbf{e}_i = [1, 1, \dots, 1]^T$ with dimension n_i .

The discriminant vectors A^* are solved by the following optimization problem:

$$A^* = \arg \max \frac{|A^T X (B - \frac{1}{n} \mathbf{e} \mathbf{e}^T) X^T A|}{|A^T \sum_{i=1}^c (X_i L_i X_i^T) A|}. \tag{8.12}$$

Considering that the original data is reconstructed by its neighbors less than ε :

$$\sum_{j=1}^n \|\mathbf{x}_j - \sum_{r=1}^k \mathbf{W}_{jr} \mathbf{x}_{jr}\|^2 < \varepsilon,$$

where ε is a small positive number. \mathbf{W} is reconstruction mapping matrix such that $\sum_{r=1}^k \mathbf{W}_{ir} = 1$. Then

$$\left\| \mathbf{x}_i - \sum_{r=1}^k \mathbf{W}_{ir} \mathbf{x}_{ir} \right\|^2 = \left\| \sum_{r=1}^k (\mathbf{W}_{ir} \mathbf{x}_i - \mathbf{W}_{ir} \mathbf{x}_{ir}) \right\|^2 = \|\mathbf{Q}_i \mathbf{W}_i\|^2,$$

where $\mathbf{Q}_i = [\mathbf{x}_i - \mathbf{x}_{i1}, \mathbf{x}_i - \mathbf{x}_{i2}, \dots, \mathbf{x}_i - \mathbf{x}_{ir}]$.

Matrix \mathbf{W} can be solved by Lagrange multiplier.

$$L_2 = \frac{1}{2} \|\mathbf{Q}_i \mathbf{W}_i\|^2 - \lambda_i \left[\sum_{r=1}^k \mathbf{W}_{ir} - 1 \right]$$

$$\frac{\partial L_2}{\partial \mathbf{W}_i} = \mathbf{Q}_i^T \mathbf{Q}_i \mathbf{W}_i - \lambda_i \mathbf{E} = \mathbf{C}_i \mathbf{W}_i - \lambda_i \mathbf{E} = 0,$$

where $\mathbf{W}_i = \lambda_i \mathbf{C}_i^{-1} \mathbf{E}$, $\mathbf{C}_i = \mathbf{Q}_i^T \mathbf{Q}_i$, $\mathbf{E} = [1, 1, \dots, 1]^T$ with dimension k .

Considering

$$\sum_{r=1}^k \mathbf{W}_{ir} = \mathbf{E}^T \mathbf{W}_i = 1 \implies \mathbf{E}^T \lambda_i \mathbf{C}_i^{-1} \mathbf{E} = 1 \implies \lambda_i = (\mathbf{E}^T \mathbf{C}_i^{-1} \mathbf{E})^{-1},$$

we have

$$\mathbf{W}_i = \lambda_i \mathbf{C}_i^{-1} \mathbf{E} = \frac{\mathbf{C}_i^{-1} \mathbf{E}}{\mathbf{E}^T \mathbf{C}_i^{-1} \mathbf{E}}.$$

The sample point is reconstructed by the optimal weights \mathbf{W} , i.e., $\mathbf{x}_j = \sum_{r=1}^k \mathbf{W}_{jr} \mathbf{x}_{jr}$. It is linearly represented by its neighbors by maintaining the local geometry in the dimensionality reduction process. Substitute it into (8.12) and NPEDA optimization is revised as follows:

$$A^* = \arg \max_A \frac{\left| \mathbf{A}^T \exp \left((\sum_{r=1}^k \mathbf{W}_{ir} \mathbf{x}_{ir}) (\mathbf{B} - \frac{1}{n} \mathbf{e} \mathbf{e}^T) (\sum_{r=1}^k \mathbf{W}_{ir} \mathbf{x}_{ir})^T \right) \mathbf{A} \right|}{\left| \mathbf{A}^T \exp \left(\sum_{i=1}^c (\sum_{r=1}^k \mathbf{W}_{jr} \mathbf{X}_{jr}^i) \mathbf{L}_i (\sum_{r=1}^k \mathbf{W}_{jr} \mathbf{X}_{jr}^i)^T \right) \mathbf{A} \right|} \quad (8.13)$$

$$= \arg \max_A \frac{\left| \mathbf{A}^T \exp(\mathbf{S}_{nb}) \mathbf{A} \right|}{\left| \mathbf{A}^T \exp(\mathbf{S}_{nw}) \mathbf{A} \right|}.$$

Equation (8.13) is equivalently to solve the maximum eigenvalue of the generalized eigenvalue decomposition problem:

$$\begin{aligned} \exp(\mathbf{S}_{nb})\mathbf{A} &= \sigma \exp(\mathbf{S}_{nw})\mathbf{A} \\ \text{or} \\ \exp(\mathbf{S}_{nw})^{-1} \exp(\mathbf{S}_{nb})\mathbf{A} &= \sigma \mathbf{A}, \end{aligned} \quad (8.14)$$

where σ is the generalized eigenvalue and the linear transformation matrix \mathbf{A} of NPEDA is the eigenvector corresponding to the first d largest eigenvalues of $(\exp(\mathbf{S}_{nw}))^{-1} \exp(\mathbf{S}_{nb})$.

8.2.3 Fault Identification Based on LLEDA and NPEDA

In this section, the LLEDA and NPEDA methods are implemented for fault identification with monitoring flowchart, as shown in Fig. 8.10. The fault recognition rate (FCR) is introduced to test the identification effectiveness. FCR of fault model i is defined as the percentage of test data identified in this corresponding model out of the total number of samples tested:

$$\text{FCR}(i) = \frac{n_{i,identify}}{n_{all}} \times 100\%, \quad (8.15)$$

where $n_{i,identify}$ denotes the sample size identified as fault i and n_{all} denotes the sample size of all samples of fault i . The identification process is given as follows,

1. Process data are collected under the normal and faulty conditions, and standardized.
2. The between-class scatter matrix \mathbf{S}_b and the within-class scatter matrix \mathbf{S}_w are calculated by the LLEDA (or NPEDA) method, respectively.
3. The discriminant vector \mathbf{A} is obtained by maximizing the between class dispersion matrix \mathbf{S}_b and minimizing the with class dispersion matrix \mathbf{S}_w .
4. The discriminant function $g(x)$ of the online data x is observed by the projection of discriminant vector \mathbf{A} in the normal model:

$$\begin{aligned} g(x) &= -\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}^i)^T \mathbf{A} \left(\frac{1}{n_i - 1} \mathbf{A}^T \exp(\mathbf{S}_w^i) \mathbf{A} \right)^{-1} \mathbf{A}^T (\mathbf{x} - \bar{\mathbf{x}}^i) \\ &\quad + \ln(c) - \frac{1}{2} \ln \left[\det \left(\frac{1}{n_i - 1} \mathbf{A}^T \exp(\mathbf{S}_w^i) \mathbf{A} \right) \right]. \end{aligned} \quad (8.16)$$

If the value of the discriminant function exceeds the normal limitation, a fault occurs.

5. The fault type of online data can be determined when its posterior probability value is maximum. The posterior probability of data \mathbf{x} in fault c_i class is calculated as

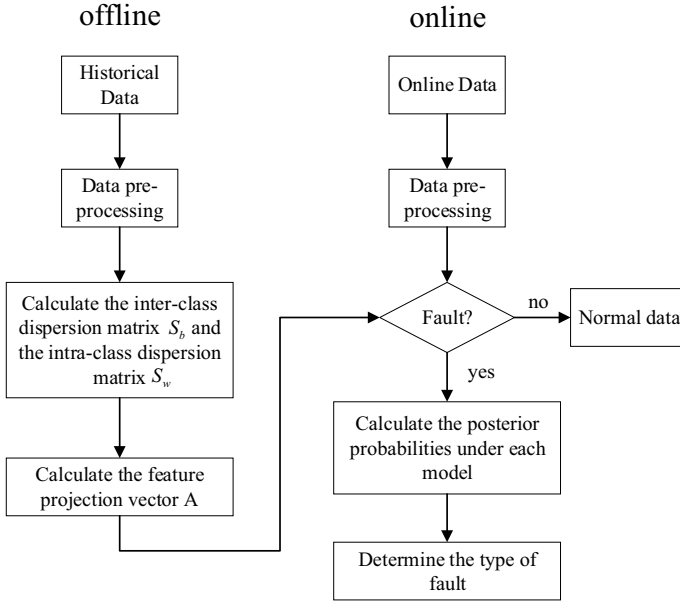


Fig. 8.10 Flowchart of fault identification with LLEDA and NPEDA methods

$$P(\mathbf{x} \in \mathbf{c}_i | \mathbf{x}) = \frac{P(\mathbf{x} | \mathbf{x} \in \mathbf{c}_i) P(\mathbf{x} \in \mathbf{c}_i)}{\sum_{i=1}^c P(\mathbf{x} | \mathbf{x} \in \mathbf{c}_i) P(\mathbf{x} \in \mathbf{c}_i)}, \quad (8.17)$$

where $P(\mathbf{x} \in \mathbf{c}_i)$ is the prior probability and $P(\mathbf{x} | \mathbf{x} \in \mathbf{c}_i)$ is the conditional probability density function of the sample \mathbf{x} :

$$P(\mathbf{x} | \mathbf{x} \in \mathbf{c}_i) = \frac{\exp[-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}^i)^T \mathbf{A} \mathbf{P}_q \mathbf{A}^T (\mathbf{x} - \bar{\mathbf{x}}^i)]}{(2\pi)^{\frac{m}{2}} [\frac{1}{n_i - 1} \mathbf{A}^T (\sum_{\mathbf{x} \in \mathbf{c}_i} (\mathbf{x} - \bar{\mathbf{x}}^i)(\mathbf{x} - \bar{\mathbf{x}}^i)^T) \mathbf{A}]^{\frac{1}{2}}}, \quad (8.18)$$

where $\mathbf{P}_q = [\frac{1}{n_i - 1} \mathbf{A}^T (\sum_{\mathbf{x} \in \mathbf{c}_i} (\mathbf{x} - \bar{\mathbf{x}}^i)(\mathbf{x} - \bar{\mathbf{x}}^i)^T) \mathbf{A}]^{-1}$.

8.2.4 Simulation Experiment

Multi-classification methods, FDA, EDA, LLE+FDA, LLEDA, and NPEDA, were carried to evaluate the classification performance in TE simulation platform. TE operation lasted for 48h, with faults occurring in the 8thh and sampled every 3 min. 400 training data were selected for building the classification model and 400 testing data for evaluating the performance of the model. Three different types of faults were considered: faults 2, 8, and 13. Fault 2 refers to a step change in the B component feed

with the A/C feed ratio remaining constant. Fault 8 refers to a random change in the A, B and C feed component variables. Fault 13 refers to a slow drift in the reaction dynamics. Here faults 8 and 13 are difficult identified due to its random variation and slow drift. The training and testing data for the three types of faults were projected onto the first and second eigenvectors, respectively, by different methods and the classification results are shown in Fig. 8.11.

Table 8.4 shows the identification rate for faults 2, 8, and 13 under different classification methods. Here the number of discrimination directions, i.e., reduction order, is considered from 1 to 10. It is shown that the identification rates are improved with increasing the number of discrimination vectors. The recognition rate for fault 2 is high, almost close to 100%. The recognition rate for faults 8 and 13 gradually increases as the number of discrimination vectors increases. NPEDA and LLEDA show higher recognition rates on faults 2, 8, and 13, compared with other methods, such as FDA and LLE+EDA.

Figure 8.12 shows the posterior probability values for the different test data under the LLEDA and NPEDA methods. The larger posteriori probability values mean the higher possibility of the test data belong to this category. Furthermore, the diagnostic results are related to the classification capability. If the classification performance is good, higher identification rate is achieved.

8.3 Cluster-LLEDA-Based Hybrid Fault Monitoring

8.3.1 Hybrid Monitoring Strategy

Generally, the data collected from an actual industrial process are unlabeled and initially undiagnosed. It is worth noting that the LLEDA method performs well in fault identification, but it is a supervised algorithm that requires the known classification of the historical data set. To overcome this problem, the supervised LLEDA method is extended into an unsupervised learning method by introducing the cluster analysis method. The cluster method can obtain the fault data category information which is input to LLEDA modeling module as a prior. To make better use of the proposed cluster-LLEDA classification method, a hybrid fault monitoring strategy is given, as shown in Fig. 8.13.

Figure 8.13 indicates that the hybrid fault monitoring strategy is mainly divided into three parts, **historical data analysis**, **fault model library establishment**, and **online detection and fault identification**. First, the historical data of industrial processes is roughly detected by PCA to label the fault data. Then hierarchical clustering technique is used to classify the process data detected as fault into different types. The model library is established for all fault types by LLEDA, which further extracts the fault features and obtain fine identification. Finally, the online detection and fault identification are realized.

The procedure of historical data analysis part is summarized as follows:

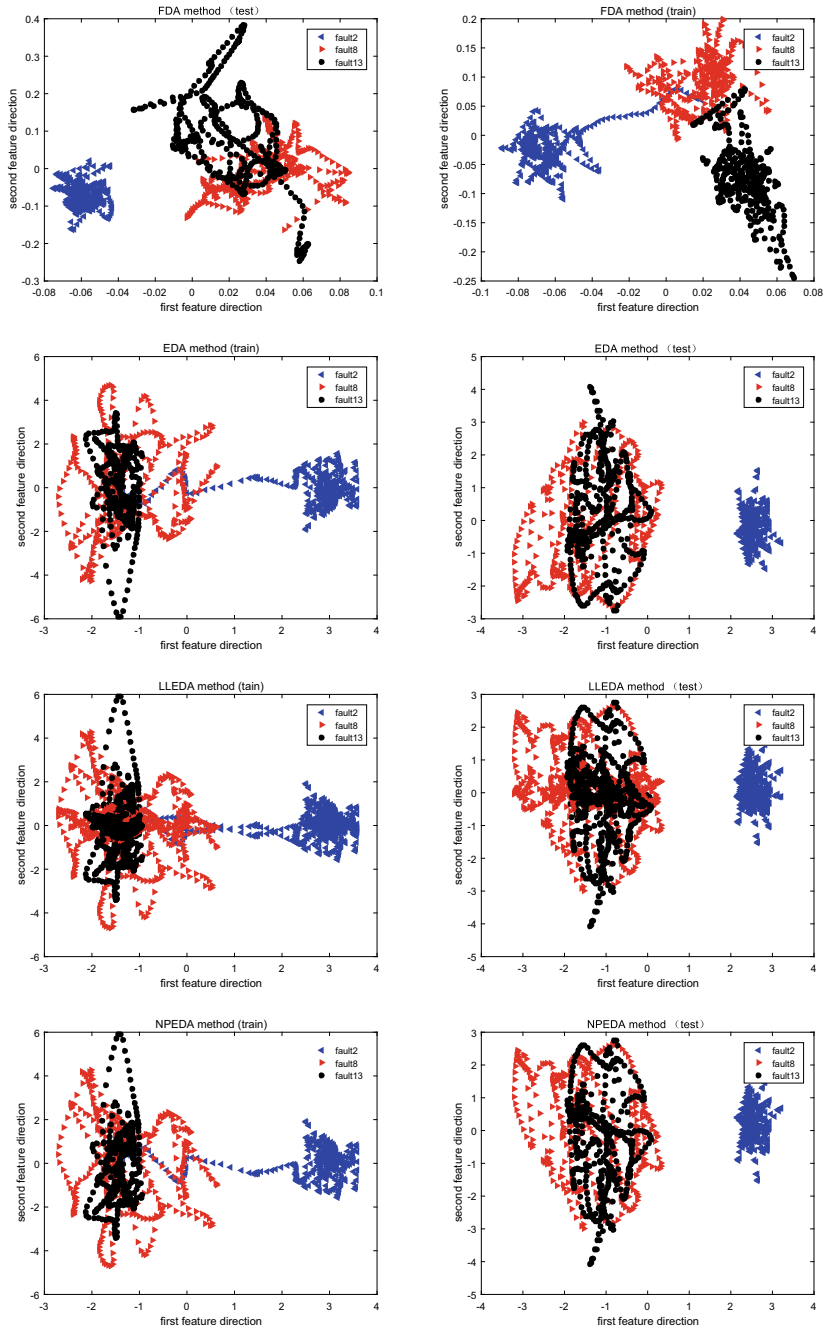


Fig. 8.11 Projection of different fault data on the first two feature vectors

Table 8.4 Comparison of identification rate for faults 2, 8, and 13

Reduction order	Fault No.	FDA	EDA	LLE+FDA	LLEDA	NPEDA
1	Fault 2	1	1	1	1	1
	Fault 8	0.4425	0.2125	0.4625	0.2125	0.2125
	Fault 13	0.415	0.6875	0.4175	0.6875	0.6875
2	Fault 2	1	1	1	1	1
	Fault 8	0.3525	0.475	0.48	0.4175	0.475
	Fault 13	0.36	0.6325	0.3475	0.6875	0.6325
3	Fault 2	1	1	1	1	1
	Fault 8	0.4375	0.67	0.3825	0.5975	0.67
	Fault 13	0.29	0.55	0.3375	0.6275	0.55
4	Fault 2	1	1	1	0.9925	1
	Fault 8	0.47	0.8325	0.425	0.705	0.8325
	Fault 13	0.2825	0.6575	0.295	0.565	0.6575
5	Fault 2	1	1	995	1	1
	Fault 8	0.625	0.8825	0.4875	0.815	0.8825
	Fault 13	0.53	0.6375	0.3025	0.5975	0.6325
6	Fault 2	1	1	1	1	1
	Fault 8	0.664	0.9325	0.62	0.895	0.9325
	Fault 13	0.5125	0.7225	0.25	0.6225	0.7225
7	Fault 2	1	1	9925	1	1
	Fault 8	0.695	0.8925	0.6	0.9125	0.8925
	Fault 13	0.49	0.7425	0.2425	0.725	0.7425
8	Fault 2	1	1	9825	1	1
	Fault 8	0.7275	0.88	0.7075	0.885	0.88
	Fault 13	0.4775	0.74	0.2275	0.7125	0.74
9	Fault 2	1	1	0.99	1	1
	Fault 8	0.745	0.88	0.6575	0.89	0.88
	Fault 13	0.49	1	0.995	1	1
10	Fault 2	0.99	1	0.995	1	1
	Fault 8	0.7625	0.8725	0.5825	0.8825	0.8725
	Fault 13	0.47	0.735	0.225	0.7125	0.735

1. Collect and standardize the normal process data from the DCS historical database.
2. Analyze the collected process data by PCA to extract the independent principle components, establish PCA model of the normal operation, and calculate the statistics of the data.
3. Calculate the statistics T^2 and SPE and their control limit.

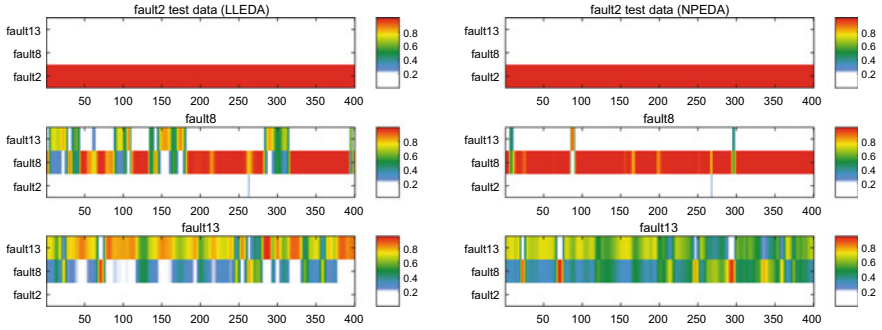


Fig. 8.12 Diagnosis results of faults 2, 8, and 13 by LLEDA and NPEDA methods

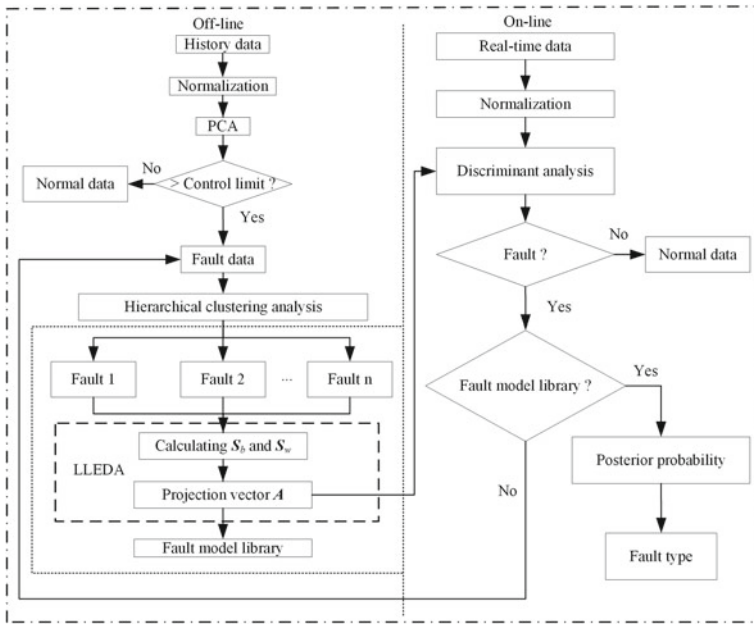


Fig. 8.13 Hybrid fault detection and diagnosis information process

The procedure of fault model library establishment is summarized as follows:

1. Perform hierarchical clustering analysis on the abnormal operation data and divide them into different fault categories.
2. Calculate the between-class and within-class scatter matrices S_b and S_w , find the corresponding projection vector A based on LLEDA method, and establish the fault model library for all fault classes.

The procedure of online detection and fault identification is summarized as follows:

1. Sample the real-time data and standardize it.
2. Perform the discriminant analysis based on LLEDA method, project the sample data to the projection direction, and extract the feature vector.
3. Project the sample data to the projection vector A based on the normal model and judge the current operation is normal or abnormal by observing whether the discriminant function exceeds the limit.
4. If a fault occurs, calculate the posterior probability in each fault model to identify the fault type. If the sample data is not in the existing fault category, this new fault will be modeled and introduced into the fault model library.

Clustering Analysis The hierarchical clustering algorithm is more widely used and has the advantages of simple calculation, fast and easy to obtain similar results, without knowing the number of clusters in advance (Saxena et al. 2017). The clustering starts with n samples each as a class, specifies the distance between samples and the clustering between classes. Then the two closest classes are merged into a new class, and the distance between the new class and the other classes are calculated. Repeat the merging process between the two closest classes, and the number of classes is reduced by one after each merging. The merging will stop until all samples are merged into one class or a certain condition is met.

The class is denoted by G in the cluster analysis. Suppose class G has m samples denoted by the column vector $x_i (i = 1, 2, \dots, m)$, d_{ij} is the distance between x_i and x_j , and D_{KL} is the distance between two different categories G_K and G_L . The squared distance D_{KL}^2 between G_K and G_L is defined as follows:

$$D_{KL}^2 = \frac{1}{n_K n_L} \sum_{x_i \in G_K, x_j \in G_L} d_{ij}^2. \quad (8.19)$$

The recursive formula for between-class squared clustering is

$$D_{ML}^2 = \frac{n_K}{n_M} D_{KJ}^2 + \frac{n_K}{n_M} D_{LJ}^2. \quad (8.20)$$

The inconsistency coefficient Y is used to determine the final number of clusters c . Here Y is a matrix of $(n - 1) \times 4$, where the first column is the mean of all link lengths (i.e., merging class distances) involved, the second column is the standard deviation of all the related link lengths, the third column is the number of related links, and the fourth column is the inconsistency coefficient.

For the links obtained by the k th merging class, the inconsistency coefficient is calculated as follows:

$$Y(k, 4) = \frac{(Z(k, 3) - Y(k, 1))}{(Y(k, 2))}, \quad (8.21)$$

where the input $Z_{(n-1) \times 3}$ is a matrix of systematic clustering trees. Under the condition that guarantees the number of classes as small as possible, the change of the inconsistency coefficient determines the final value of classes number.

8.3.2 Simulation Study

The experiment uses the Tennessee Eastman (TE) process to evaluate the effectiveness of the proposed hybrid method.

Experiment 1: Failure Initial Screening and Classification The TE data set was first detected by the PCA method, and the fault detection results are shown in Fig. 8.14, the final T^2 and SPE statistics obtained were 0.4951 and 0.6882, respectively. The specific detection is shown in Table 8.5. The results show that the recognition rate of faults 1, 2, 6, 7, 8, 12, 13, 14, 17, and 18 is high, and the recognition rate of other faults is low. This indicates that the significant faults can be detected, while the potential faults cannot be detected.

Therefore, PCA-based fault detection methods can only coarsely split the data set and detect significant faults. Potential faults can be identified with a high fault identification rate only in the case of known fault categories. In the coarse separation stage of historical data, the fault data can be identified not only by PCA method, but also by improved PCA or other fault detection methods to further improve the identification rate.

After the historical data analysis, the fault data set is collected and clustered into different fault classes by using the hierarchical clustering method. According to the inconsistency coefficient, the final number of fault classes is 10. As the fault type is in a large number, it is difficult to display the classified fault data together in a tree diagram. As example, we select the faults 1, 2, and 6 to demonstrate the clustering effect of the hierarchical cluster analysis algorithm. Fault 1 is a step change in the A/C feed ratio with component B remaining unchanged, while fault 2 is a step change

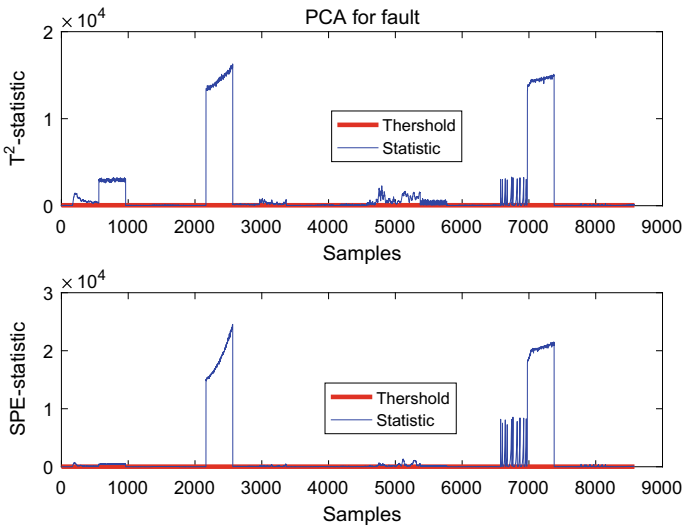


Fig. 8.14 Fault detection based on PCA

Table 8.5 Fault recognition rate based on PCA

Fault No.	T ²	SPE	Fault No.	T ²	
Fault 1	0.995	0.9988	Fault 12	0.9875	0.99
Fault 2	0.9825	0.9925	Fault 23	0.9513	0.9625
Fault 3	0.0225	0.2675	Fault 14	0.9988	1
Fault 4	0.41	1	Fault 15	0.0488	0.2625
Fault 5	0.2625	0.5025	Fault 16	0.2325	0.6937
Fault 6	0.99	1	Fault 17	0.8013	0.975
Fault 7	1	1	Fault 18	0.8912	0.9375
Fault 8	0.975	0.9825	Fault 19	0.0675	0.5913
Fault 9	0.0362	0.235	Fault 20	0.3738	0.735
Fault 10	0.4163	0.7638	Fault 21	0.3775	0.6687
Fault 11	0.5212	0.8163			

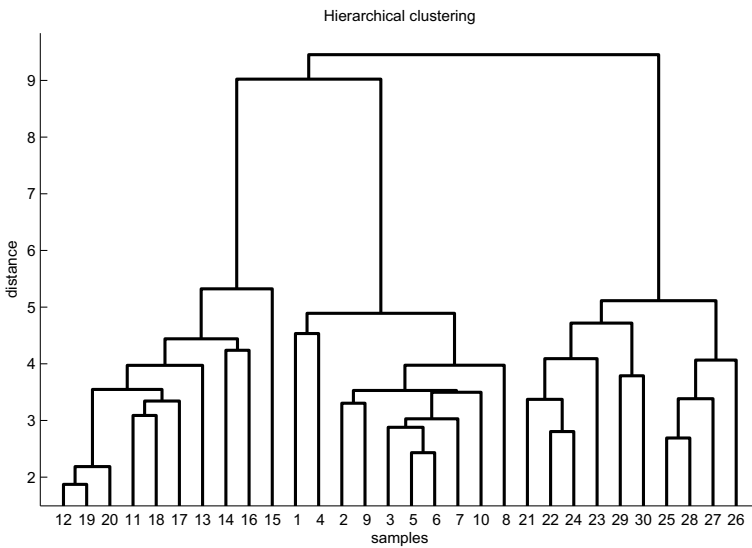


Fig. 8.15 Hierarchical cluster analysis

in component B with the A/C ratio remaining unchanged. Fault 6 is a step change in the feed loss of A. The hierarchical clustering tree diagram is given Fig. 8.15. The final number of categories is three according to the inconsistency coefficient, which is consistent with the actual classification.

Now the fault data have been divided into 10 classes by hierarchical cluster analysis. Obviously, the dimension is high and its visualization effect is poor. In order to improve the visualization effect and reflect the change trend and the interrelationship between each variable at the same time, the parallel coordinate visualization method

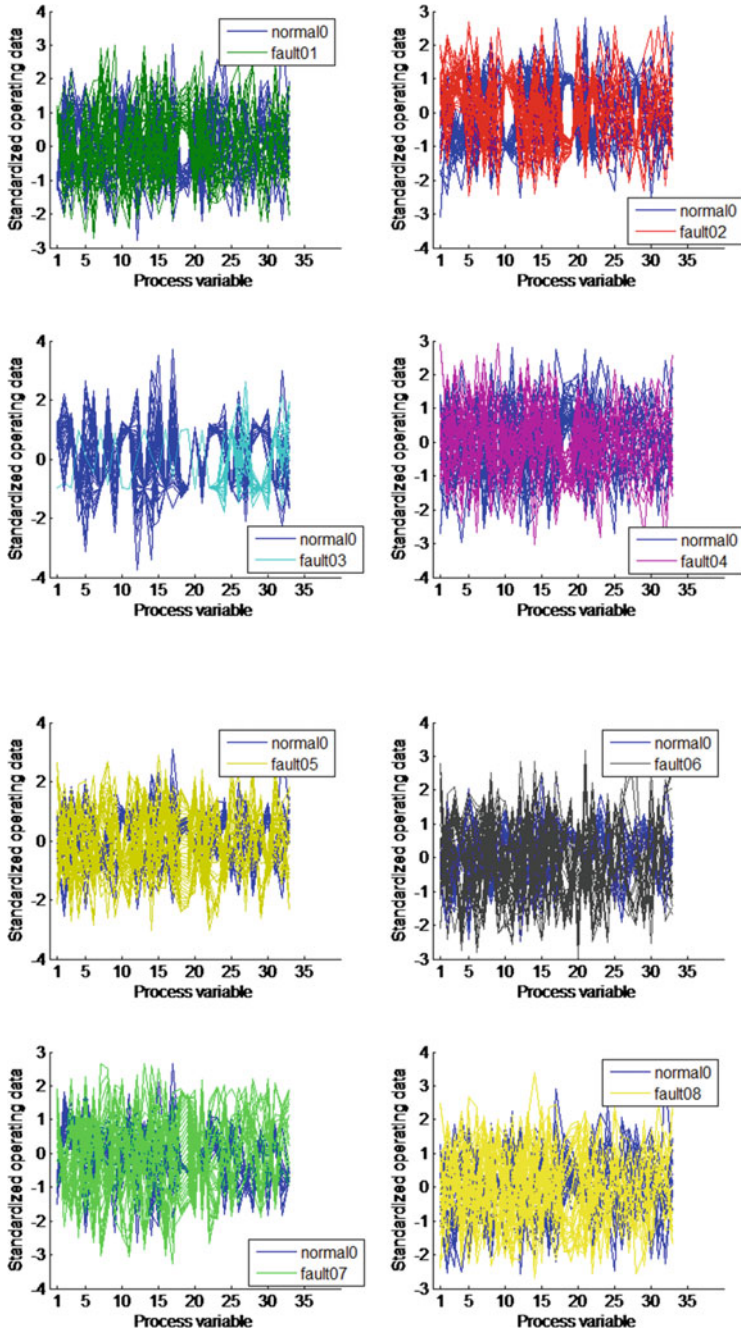


Fig. 8.16 Parallel coordinate visualization of fault data

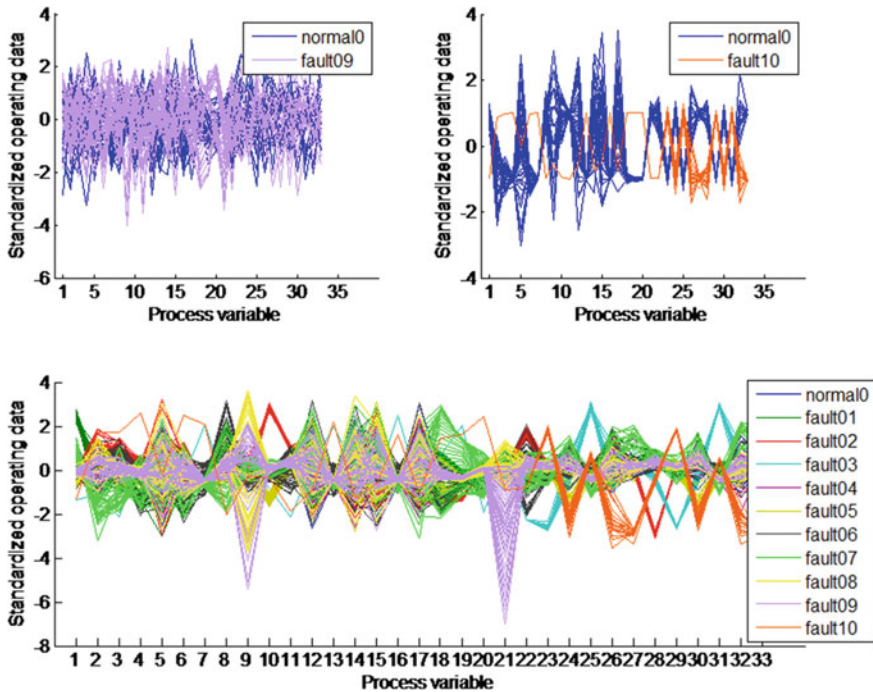


Fig. 8.16 (continued)

is selected. It is a visualization technique that allows the high-dimensional variables to be represented by a series of axes parallel to each other. The value of the variables is corresponding to the positions on the axes.

The visualization results for each type of fault data are shown in Fig. 8.16. The blue dash in each subplot indicates the normal data and the other color dashes indicate different fault data. Since each variable in the TE data has a corresponding actual physical meaning, the type of fault can be judged by comparing the other color dashes with the blue dash in each variable. These faults can be labeled for establishing the fault model library.

Experiment 2: LLEDA-based Fault Identification The fault identification method used here is LLEDA, which increases the distance between different classes and improves the classification ability even if fault samples are small. Here faults 4, 8, and 13 are selected as example to show the identification results. Fault 4 is a minor fault, which is manifested in the step change of the inlet temperature of the reactor cooling water, but the other 50 variables are still in a stable state, and the change is less than 2% compared with the normal data. Fault 13 refers to the slow drift of reactor kinetic constants when the fault occurs, which will cause a violent reaction of each variable, and the final product G is always in a fluctuating state. Fault 8 refers

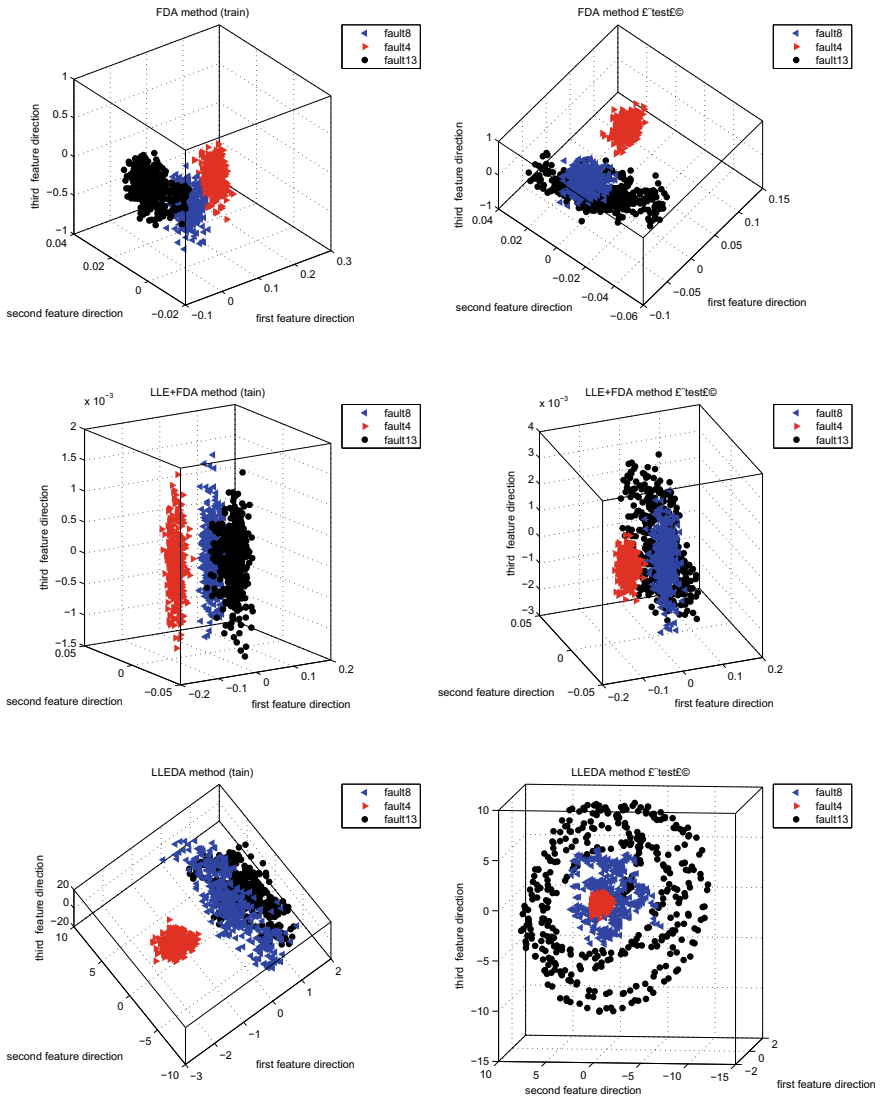


Fig. 8.17 Projection of different fault data on feature vectors

to the change of random variables of A, B, and C feed ingredients when the fault occurs.

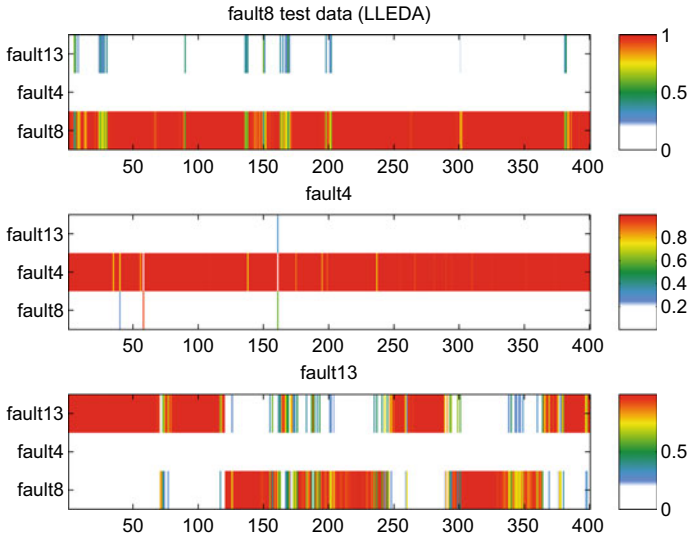


Fig. 8.18 Diagnosis results of fault 4, 8, and 13 by LLEDA methods

To better observe the classification in spatial structure, the training data and testing data of the three faults are projected onto the first three feature vectors by different methods. The classification results are shown in Fig. 8.17.

Figure 8.18 shows the posterior probability values of different test data by LLEDA method under different models. The posterior probability values are larger when the samples belong to category i . The colored bars indicate the diagnostic result, i.e., probability values, in which color bar from bottom to top is corresponding to the probability values 0–1 (white indicates that the probability of identification is 0 and red indicates that the probability value of identification is 1.) In this way, the fault identification results are visualized. The diagnosis result is related to the classification ability. The better classification performance leads to a higher fault recognition rate. Here fault 13 is in poor classification owing to the small number of feature vectors. The recognition rate of faults can be improved by increasing the number of feature vectors.

8.4 Conclusion

This chapter presents three discriminant analysis methods, KEDA, LLEDA and NPEDA, that can handle nonlinearities and avoid small sample data problems. Nor-

mal and faulty data models are developed, and these models are used to check whether abnormal behavior occurs, and variance-based performance metrics are used to identify the type of data tested. Especially, two new supervised dimensionality reduction methods, LLEDA and NPEDA, are proposed which combines the advantages of local linear embedding and exponential discriminant analysis methods, taking into account both global and local information. The nonlinear data is piecewise linearized by maintaining the internal structure during the extraction of the eigenvalues. They overcome the singularity problem of within-class scatter matrices, and therefore show good performance for the small sample problem.

Furthermore, the hybrid process monitoring and fault identification algorithm is proposed in this chapter, which effectively combines the PCA initial detection, the classification of hierarchical clustering, and the discriminative analysis of LLEDA. This hybrid method ensures the monitoring and diagnosis is performed directly on the collected data without a priori knowledge.

Reference

Saxena A, Prasad M, Gupta A, Bharill N, Patel O, Tiwari A, Er M, Ding W, Lin C (2017) A review of clustering techniques and developments. *Neurocomputing* 267:664–681

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

