

# Fault Detection Method based on Incrementable Laplacian Eigenmaps and Normal Space

Liwei Feng\*, Yu Xing\*\*, \*\*\*, Shaofeng Guo\*\*\*, \*\*\*, Yifei Wu\*\*\*, \*\*\*,  
Guozhu Wang\*\*\*\*, Yuan Li\*

\* College of Science, Shenyang University of Chemical Technology, Shenyang, 110142, Liaoning, China  
(e-mail: fengliwei@syuct.edu.cn, li-yuan@mail.tsinghua.edu.cn).

\*\* College of Computer Science and Technology, Shenyang University of Chemical Technology,  
Shenyang, 110142, Liaoning, China (e-mail: xingyu69@163.com, 2574386416@qq.com,  
a2435208397@163.com)

\*\*\* Key Laboratory of Intelligent Technology for Chemical Process Industry of Liaoning Province,  
Shenyang, 110142, Liaoning, China

\*\*\*\* School of Cable Engineering, Henan Institute of Technology, Xinxiang, 453003, Henan, China  
(e-mail: wang.guo.zhu@163.com).

**Abstract:** Aiming at the difficulty of embedding new samples added and the loss question of information when using Laplacian Eigenmaps (LE) for process monitoring, a fault detection method based on Incremental Laplacian Eigenmaps and Normal Space (ILENS) is developed. After feature extraction of the training data using Laplacian Eigenmaps, the local projection matrix is constructed from the local information for embedding new samples. It ensures that normal samples can be embedded in the manifold of the training data in the feature space and that most fault samples can be separated from normal samples. Then Normal Space of the manifold is constructed based on the local information of the samples. In the Normal Space, a small number of fault samples falling into the low-dimensional manifold can be separated from normal samples. The ILENS method was compared with PCA, KPCA, FD-KNN, and RP-KNN through a numerical simulation process and a turbocharged spark-ignited engine system simulation process. The results show that ILENS possesses a higher fault detection rate compared to other classical methods.

**Keywords:** Process control, Fault detection, Laplacian Eigenmaps, Out-of-sample, Nonlinear process

## 1. INTRODUCTION

Currently, industrial production is growing in complexity and scale. Safety has become the focus of attention in the production process so fault detection technology based on process monitoring has been developed rapidly. Fault detection is strongly linked to modeling. In the field of modeling, successful performance has been achieved in many applications. Abramov et al. proposed the new opportunities model to monitor, analyze, and forecast COVID-19 (Abramov et al., 2023). Yurtkan et al. proposed a novel approach for predicting student success by using conventional feed-forward neural networks (Yurtkan et al., 2023). Zheng et al. proposed an improved extraction approach applied to the line structured light stripe center extraction process (Zheng et al., 2023). In the field of process monitoring, principal components analysis (PCA) and partial least squares (PLS) have been widely researched and developed as classical methods (Cao et al., 2003; Deng et al., 2016; Henseler et al., 2016).

PCA and PLS are suitable for processing data with a linear manifold structure but have limitations when the data exhibit nonlinearities, which real process data usually exhibit. The classical solutions for this problem are Kernel Principal Components Analysis (KPCA) (Lee et al., 2004), Kernel Partial Least Squares (KPLS) (Rosipal et al., 2001), Local Outlier Factor (LOF) (Breunig et al., 2000), fault detection-k-nearest neighbor rule (FD-KNN) (He et al., 2007), and so on.

KPCA and KPLS resolve nonlinear problems by projecting the data from the original space to a high-dimensional space through nonlinear mapping, then processing the data in the high-dimensional feature space. KPCA and KPLS have been widely studied and applied in the fault diagnosis of nonlinear processes (Fazai et al., 2019; Said et al., 2020; Navi et al., 2015; Sun et al., 2020). The kernel methods do not take into account the manifold structure of the data, which does not necessarily satisfy the Gaussian distribution in the feature space, leading to poor detection. LOF calculates the outlier factor for each sample point based on the data densities. It uses the outlier factor to identify the degree of outliers in the data, thus enabling the detection of faults. FD-KNN performs fault detection utilizing a statistic  $D^2$  constructed from sample nearest neighbor information. Both LOF and FD-KNN determine whether a sample is a fault point by the outlier degree of the sample, without considering whether the structure of the process data presents nonlinearity, therefore works well when monitoring nonlinear processes. LOF and FD-KNN perform well in many applications in the process monitoring domain (Guo et al., 2018; Kim et al., 2022), however, the methods do not involve dimensionality reduction and therefore will incur high computational costs when the dataset dimensionality is too high. In this regard, ZHOU et al. improved it by applying Random Projections to the FD-KNN method and proposed fault detection using Random Projections and k-Nearest Neighbor Rule (RP-KNN) (Zhou et al., 2014). RP-KNN first maps the data to low dimensions by

Random Projections, then performs fault detection based on the FD-KNN method, which effectively reduces the computational cost, but at the same time loses some accuracy.

Process data is usually located on a manifold with nonlinear feature. Traditional methods such as KPCA, LOF, FD-kNN, etc., although they can effectively deal with nonlinear data, ignore the manifold structure that exists within the data. To extract the features of the manifold structure present within the data, the Manifold Learning method can be used. Manifold Learning aims at discovering the intrinsic manifold structure of nonlinear high-dimensional data and extracting the low-dimensional manifold of the data. The classical methods in this field are Isometric Mapping (ISOMAP) (Tenenbaum et al., 2000), Locally Linear Embedding (LLE) (Roweis et al., 2000), Local Tangent Space Alignment (LTSA) (Zhang et al., 2004), Laplacian Eigenmaps (LE) (Belkin et al., 2003), and so on. Among them, LE has been studied most intensively and extensively (Li et al., 2019; Donoho et al., 2003; He et al., 2003; Jiang et al., 2017). LE possesses an advantage that other methods do not have, which is ideal robustness when outliers are present. LE constructs relationships between data in a localized perspective and reconstructs the local features by constructing a relationship graph, thus reflecting the inherent manifold structure of the data. However, LE has two defects for fault detection in process monitoring. The first one is that there is no projection matrix in the process of feature extraction by LE because of nonlinear mapping, and no way to directly embed the new samples to feature space. When a new sample is added, it needs to be incorporated into the training data and reprocessed, which will undoubtedly greatly increase the computational cost and workload. Additionally, when the new samples are outliers, it is not reasonable to use LE for feature extraction directly after merging new samples into the training data. Another defect is that there exists only one manifold feature space after LE processing, therefore some information must be lost after feature extraction, which will affect the accuracy of process monitoring.

Aiming at the difficulty of embedding new samples added by LE and the loss question of information of LE, a fault detection method based on Incrementable Laplacian Eigenmaps and Normal Space is developed in this paper. First, the low-dimensional manifold structure of the data is extracted by LE. Then the local projection matrix is constructed with the local information of the new sample to realize the embedding of it. After that, the local information of the sample is used to construct the Normal Space as a complement to the manifold feature space. The manifold feature space is monitored using the  $D^2$  statistic and the Normal Space is monitored using the Squared Prediction Error (SPE).

The paper is organized as follows. A review of Laplacian Eigenmaps is given in Section 1. In Section 2, the ILENS strategy is presented and analyzed. In Section 3, the superiority of using ILENS for process monitoring is illustrated by two examples, a numerical simulation process and a turbocharged spark-ignited engine system simulation process. Finally, we present our conclusions in Section 4.

## 2. LAPLACIAN EIGENMAPS

Laplacian Eigenmaps is a graph-based manifold learning method that constructs relationships between data from a local perspective. The method reconstructs the local structural features of the data in a low-dimensional space by expressing the adjacency graph between the data through the constructed adjacency matrix  $W \in \mathbb{R}^{n \times n}$ . For the original data  $X = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^T$ ,  $\mathbf{x}_i \in \mathbb{R}^m$ , denote the embedding result of  $X$  after LE processing as  $Y = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n)^T$ ,  $\mathbf{y}_i \in \mathbb{R}^d$ . Laplacian Eigenmaps expects points in the adjacency graph that are connected to each other to be as close as possible in the space after feature extraction. Therefore, the objective function of the optimization as

$$\min \frac{1}{2} \sum_{i,j} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{ij}, s.t. Y^T D Y = I \quad (1)$$

where  $I$  is the unit matrix. The adjacency matrix  $W$  reflects the degree of similarity between the samples, and it is constructed in two steps:

a) Construct the undirected graph  $G$  by connecting each sample to its  $k$  nearest samples;

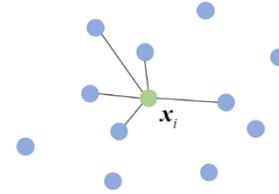


Fig. 1. Connect  $\mathbf{x}_i$  with its  $k$  nearest neighbors.

b) Determine the weights between the samples in  $G$ . If there is an edge between point  $\mathbf{x}_i$  and point  $\mathbf{x}_j$ , put the weights between them as

$$W_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\varepsilon}} \quad (2)$$

where parameter  $\varepsilon \in \mathbb{R}$  is a constant. Otherwise, put  $W_{ij} = 0$ . Another optional simplification is that if points  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are connected, then  $W_{ij} = 1$ , otherwise, put  $W_{ij} = 0$ .

Simplified objective function gives

$$\frac{1}{2} \sum_{i,j} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{ij} = \text{trace}(Y^T L Y) \quad (3)$$

where  $L \in \mathbb{R}^{n \times n}$  is the Laplacian matrix and calculated by

$$L = D - W \quad (4)$$

where matrix  $D \in \mathbb{R}^{n \times n}$  is the diagonal weight matrix and each of its entries is defined as

$$D_{ii} = \sum_{j=1}^n W_{ij} \quad (5)$$

the optimization problem for the Laplacian Eigenmaps becomes

$$\min \text{trace}(Y^T LY), s.t. Y^T DY = I \quad (6)$$

The constraint  $Y^T DY = I$  removes an arbitrary scaling factor in the embedding. The optimization problem is solved using the Lagrange multiplier method:

$$f(Y) = \text{tr}(Y^T LY) + \text{tr}[\Lambda(Y^T DY - I)] \quad (7)$$

$$\frac{\partial f(Y)}{\partial Y} = LY + L^T Y + D^T Y \Lambda + DY \Lambda = 0 \quad (8)$$

from the above get  $LY = -\Lambda DY$ , which becomes a generalized eigenvalue problem. Therefore, choose the matrix  $Y \in \mathbb{R}^{n \times d}$  of eigenvectors corresponding to the lowest  $d$  non-zero eigenvalues as the output after dimensionality reduction.

### 3. ILENS FAULT DETECTION METHOD

#### 3.1 Out-of-sample problem

The Out-of-sample problem often arises when applying LE to real-world data. There is no projection matrix between the original data and their low-dimensional embedding when processing datasets by LE, thus new samples cannot be added directly (Chen et al., 2010). After a new sample is added, to obtain its low-dimensional embedding, the new sample needs to be merged into the training samples and then reduce dimensionality again. The approach undoubtedly increases the computational cost, and when the new sample is an outlier since not on the manifold structure, it is not reasonable to embed it by LE even after merging it into the training samples. To address the out-of-sample problem, this paper proposes an Incrementable Laplacian Eigenmaps (ILE), the main idea is to find an approximate local transformation of the neighborhood of  $x_{new}$  from the original space to the low-dimensional feature space, and then apply this transformation to  $x_{new}$  in order to realize its embedding. Denote the neighborhood of  $x_{new}$  as  $X_{Ni} \in \mathbb{R}^{k \times m}$  and the data corresponding to  $X_{Ni}$  in the low-dimensional feature space as  $Y_{Ni} \in \mathbb{R}^{k \times d}$ . The next step is to find the approximate transformation of  $X_{Ni}$  to  $Y_{Ni}$ . Although the overall process is a nonlinear transformation process when feature extraction is performed on the sample set by LE, for localization, it can be approximated as a linear transformation. It is assumed that the local space where  $X_{Ni}$  is located can be approximately equal to  $Y_{Ni}$  after linear projection and similarity transformation. Denote  $X_{Ni}$  after this approximate transformation as

$$Y_{Ni}' = X_{Ni} T S R + b \quad (9)$$

where  $T \in \mathbb{R}^{m \times d}$  is the local projection matrix,  $S \in \mathbb{R}^{d \times d}$  is the scaling matrix,  $R \in \mathbb{R}^{d \times d}$  is the rotation matrix, and  $b \in \mathbb{R}^d$  is the vector representing the displacement. The  $Y_{Ni}'$  can be approximately equal to  $Y_{Ni}$ . This process can be specifically divided into four steps as shown in Fig. 2. Elaborate on each step below.

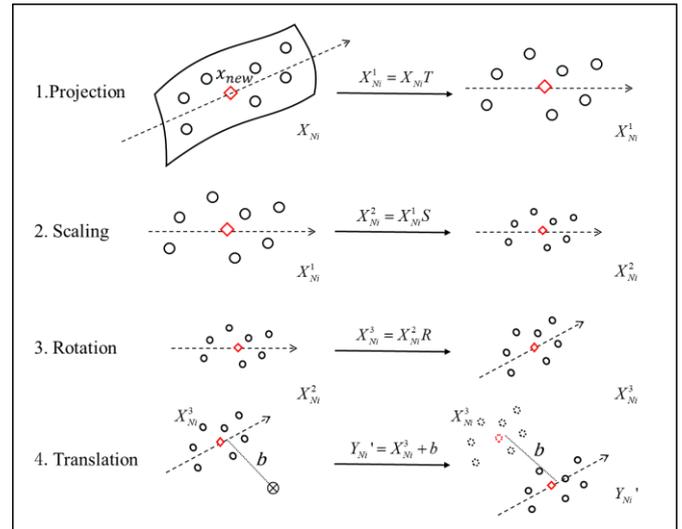


Fig. 2. Schematic diagram of an approximate transformation of a local space.

The first step is projection. Since we consider the localized transformation as a linear projection, the local projection matrix  $T$  can be obtained by the PCA method. Compute the covariance matrix for the neighborhood  $X_{Ni}$  through

$$C = \frac{1}{n-1} X_{Ni}^T X_{Ni} \quad (10)$$

the eigenvalue decomposition of the covariance matrix is then performed:  $C p_i = \lambda_i p_i$ . The eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_m$  arranged from largest to smallest and the  $P_{m \times m} = (p_1, p_2, \dots, p_m)$  consisting of the corresponding eigenvectors is obtained. Then split the matrix  $P$  according to the LE target space dimension  $d$ :

$$P_{m \times m} = (T_{m \times d}, O_{m \times q}) \quad (11)$$

where  $d + q = m$ . After obtaining the local projection matrix  $T$ , the first step of the transformation of  $X_{Ni}$  can be realized by

$$X_{Ni}^1 = X_{Ni} T \quad (12)$$

The second step is scaling. The scaling matrix  $S$  is constructed using the range of  $X_{Ni}^1$  and  $Y_{Ni}$  in each principal axis direction by

$$S = \text{diag}\left(\frac{r_1}{r_1'}, \frac{r_2}{r_2'}, \dots, \frac{r_d}{r_d'}\right) \quad (13)$$

where  $r_i$  is the range of  $Y_{Ni}$  in the direction of the  $i^{\text{th}}$  principal axis, and  $r_i'$  is the range of  $X_{Ni}^1$  in the direction of its  $i^{\text{th}}$  principal axis. After obtaining the scaling matrix  $S$ , the second step of transformation can be applied to  $X_{Ni}$  through

$$X_{Ni}^2 = X_{Ni}^1 S \quad (14)$$

The third step is rotation, which requires the construction of a rotation matrix  $R$ . Start by constructing  $V = (v_1, v_2, \dots, v_d)$

with unit vectors in each principal axis direction of  $Y_{Ni}$ . where  $v_i$  is the unit vector in the direction of the  $i^{th}$  principal axis of  $Y_{Ni}$ . The  $X_{Ni}^2$  has been processed using PCA, thus each principal axis direction of  $X_{Ni}^2$  is the direction of the coordinate axis. If wish to rotate  $X_{Ni}^2$  so that the directions of its principal axes coincide with the directions of the principal axes of  $Y_{Ni}$ , then it is sufficient to multiply  $X_{Ni}^2$  by the inverse matrix of  $V$ . It can be seen that the rotation matrix  $R = V^{-1}$ . The third step of the transformation of the matrix  $X_{Ni}$  can be performed by

$$X_{Ni}^3 = X_{Ni}^2 R \tag{15}$$

The last step is translation, and construct the translation  $b$  below. Use the center coordinates  $c_X$  of  $X_{Ni}^3$  and the center coordinates  $c_Y$  of  $Y_{Ni}$  for the construction of the displacement

$$b = c_X - c_Y \tag{16}$$

once the displacement  $b$  is obtained, the fourth step of the transformation of  $X_{Ni}$  can be performed by

$$Y_{Ni}' = X_{Ni}^3 + b \tag{17}$$

At this point, an approximate transformation of the neighborhood  $X_{Ni}$  of  $x_{new}$  from the original space to the low-dimensional feature space is obtained. By applying this approximate transformation to  $x_{new}$ , the embedding result  $y_{new}$  of  $x_{new}$  in the feature space can then be obtained by

$$y_{new} = x_{new} TSR + b \tag{18}$$

The embedding effect depends on  $k$ , the number of samples in the neighborhood. The following search for the optimal value of  $k$ . Use the LE method for the entire dataset  $X$  to obtain a low-dimensional representation  $Y$ , then divide  $X$  into two parts:  $X = [X_{train}, X_{new}]$ . And do the same put  $Y = [Y_{train}, Y_{new}]$ . The low dimensional embedding  $Y_{new}'$  of  $X_{new}$  is calculated by Eq. (18). Now denote the embedding error based on the root mean square error (RMSE) between the actual dataset  $Y_{new}$  and the estimated dataset  $Y_{new}'$ :

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (Y_{new}^i - Y_{new}'^i)^2}{n}} \tag{19}$$

where  $Y_{new}^i$  denotes the  $i^{th}$  sample in  $Y_{new}$ . The RMSE can reflect the magnitude of error between the estimated dataset  $Y_{new}'$  obtained by the above embedding method and the actual dataset  $Y_{new}$ , and the smaller the RMSE value, the better the effect of the new sample embedding.

Fig. 3 shows the 3-D S-curve data widely used in manifold learning (Saul et al., 2003). The expression for the dataset is

$$\begin{cases} x_1 = \sin(\alpha) \\ x_2 \sim U(0,5) \\ x_3 = \text{sign}(\alpha)[\cos(\alpha) - 1] \end{cases} \tag{20}$$

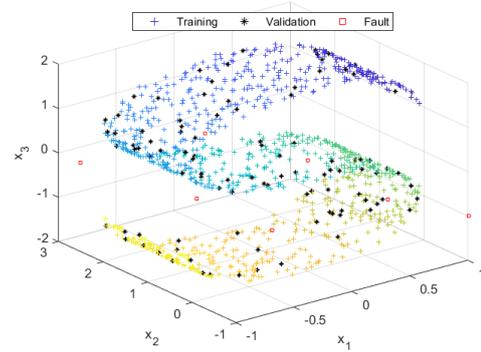


Fig. 3. The 3-D S-curve data original spatial scatter plot.

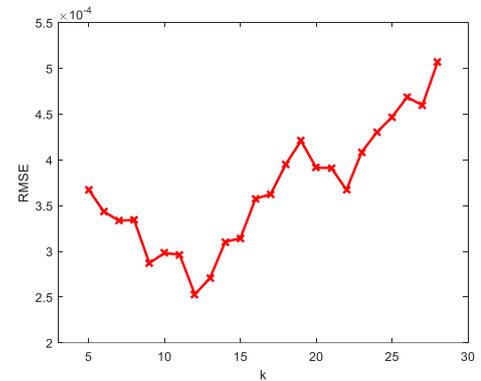


Fig. 4. RMSE diagram of 3-D S-curve training data.

where the latent variables  $\alpha$  satisfy  $\alpha \sim U(-1.5\pi, 1.5\pi)$ ,  $\text{sign}$  is a symbolic function that returns +1 when  $\alpha > 0$ , 0 when  $\alpha = 0$ , and -1 when  $\alpha < 0$ . Calculate the RMSE for different values of the parameter  $k$  using the training set. The results are shown in Fig. 4, the optimal value of  $k$  is 13 for this dataset. From Fig. 4, the RMSE is very small and the embedding error is close to zero when the value of  $k$  is in the right range, thus it can show that the proposed Incrementable Laplacian Eigenmaps method has an excellent performance when embedding new samples.

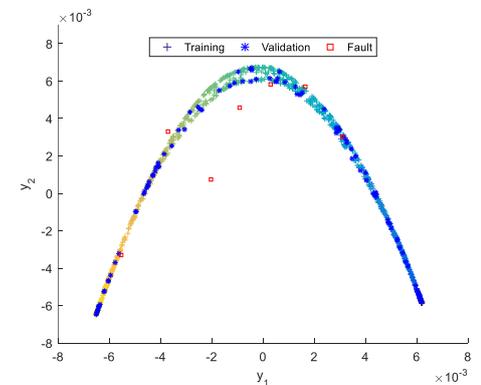


Fig. 5. The manifold feature space of 3-D S-curve data.

Perform feature extraction on the 3-D S-curve data by LE and embed the validation data and fault data in manifold feature space by Eq. (18). The obtained results are shown in Fig. 5, all the validation data can be correctly embedded into the feature extracted low-dimensional manifold, but for a few fault data, they are also incorrectly mixed into the normal samples, leading to missed reports during fault detection.

### 3.2 Normal Space

Since only one manifold feature space exists after the LE method processes the dataset, some information may be lost, and as can be seen from Fig. 5, a small number of fault samples fall into the normal samples after feature extraction. For this reason, this paper proposes the Normal Space, making it complementary to the manifold features space. After obtaining the local projection matrix  $T$  and the residual projection matrix  $O$  of the local space of the sample through Eq. (11), the  $O$  can be used to construct the Normal Space. The specific steps are as follows:

1) For a sample  $x_i$ , we find its  $k$  nearest neighbor composition  $X_{Ni}$ , and obtain the local projection matrix  $T$  and residual projection matrix  $O$  of  $X_{Ni}$  by Eq. (11).

2) Compute the residual projection  $z_i \in \mathbb{R}^q$  and  $Z_{Ni} \in \mathbb{R}^{k \times q}$  of the sample  $x_i$  with  $X_{Ni}$  in local space:

$$z_i = x_i O \quad (21)$$

$$Z_{Ni} = X_{Ni} O \quad (22)$$

3) Calculate the difference between  $x_i$  and the mean of  $X_{Ni}$ , and use this difference vector  $h_i \in \mathbb{R}^q$  as the mapping of the sample  $x_i$  in Normal Space by the expression

$$h_i = z_i - \frac{1}{k} \sum_{j=1}^k l_j \quad (23)$$

where  $l_j$  is the  $j^{\text{th}}$  sample of the data set  $Z_{Ni}$ . If the original data is 3-dimensional, the  $\|h_i\|$  can be regarded as the distance from the sample to the local manifold, as shown in Fig. 6.

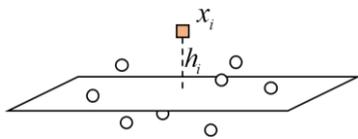


Fig. 6. Localized manifold.

Construct the Normal Space of the 3-D S-curve data by Eq. (23), as shown in Fig. 7, in which all the fault samples can be separated from the normal samples. After constructing the Normal Space, selecting the appropriate statistics is especially important, so it is necessary to analyze the characteristics of the distribution of data in the Normal Space. Plot the normality test of normal space.

The statistic is determined by the type of distribution of the data. If the data do not obey a normal distribution, it is necessary to use data fitting methods to obtain the distribution

obeyed by the samples in normal space, to determine the appropriate statistics.

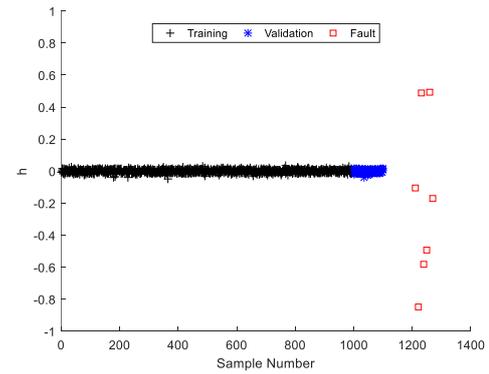


Fig. 7. 3-D S-curve data normal space.

Fig. 8 is the normality test diagram of the Normal Space of 3-D S-curve data. From the figure, it can be seen that the samples approximately obey a normal distribution in the Normal Space.

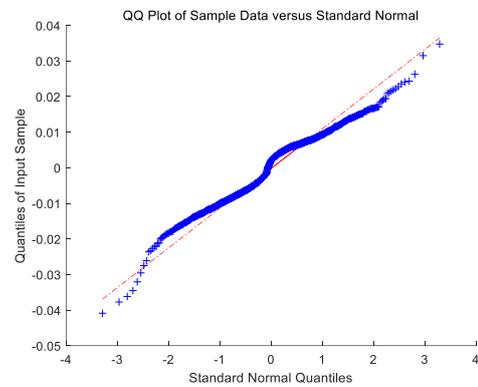


Fig. 8. Normality test of 3-D S-curve data in normal space.

Therefore, in this space, the samples can be monitored by SPE statistics. The SPE statistic is calculated as

$$SPE_i = \|h_i\|^2 \quad (24)$$

### 3.3 ILENS

Obtain the manifold feature space and Normal Space of the training data by the ILE and Eq. (23). For the new samples, embedding is done by Eq. (18). In the manifold feature space, Use the  $D^2$  statistic for monitoring, which is constructed by the equation

$$D_i^2 = \frac{1}{t} \sum_{j=1}^t \bar{d}_{i,j}^2 \quad (25)$$

where  $t$  is the number of neighbors,  $\bar{d}_{i,j}^2$  denotes squared Euclidean distance from sample  $x_i$  to its  $j^{\text{th}}$  nearest neighbor. In the Normal Space, the SPE statistic is used for monitoring.

The specific monitoring steps of the ILENS method are as follows:

1) Model building

a) For the training samples, construct the undirected graph  $G$  by connecting each sample to its  $k$  nearest samples and determine the weights of the edges. If sample  $x_i$  and sample  $x_j$  are connected, then set the weights  $W_{ij}$  by Eq. (2). Otherwise, put  $W_{ij} = 0$ ;

b) Compute eigenvalues and eigenvectors for the generalized eigenvector problem  $Lf = \lambda Df$ , where  $D$  is the diagonal weight matrix, constructed by Eq. (5), and  $L = D - W$  is the Laplacian matrix. Then choose the eigenvectors corresponding to the lowest  $d$  non-zero eigenvalues as the output in manifold feature space;

c) Calculate the mapping of the training samples in Normal Space by Eq. (21) to Eq. (23);

d) Calculate the  $D^2$  statistics and the  $SPE$  statistics of the training samples by Eq. (25) and Eq. (24), respectively;

e) Based on confidence  $\alpha$ , the control limits  $D_\alpha^2$  and  $SPE_\alpha$  of  $D_i^2$  and  $SPE_i$  are calculated using kernel density estimation, respectively.

## 2) Fault detection

a) For a new sample  $x_{new}$ , project it into the manifold feature space by Eq. (18) and the Normal Space by Eq. (21) to Eq. (23).

b) Calculate the  $D_{new}^2$  statistics and the  $SPE_{new}$  statistics for  $x_{new}$  by Eq. (25) and Eq. (24), respectively;

c) If  $D_{new}^2 < D_\alpha^2$  and  $SPE_{new} < SPE_\alpha$ , then  $x_{new}$  is categorized as a normal sample, otherwise it is a fault sample.

## 4. EXAMPLE EXPERIMENT

In this section, through two examples, a nonlinear numerical simulation process and a turbocharged spark-ignited engine system simulation process to verify the effectiveness of the ILENS. And compared the ILENS method with PCA, KPCA, FD-KNN, and RP-KNN methods.

### 4.1 Numerical simulation process.

The numerical simulation process is shown in equation (26), which has six monitored variables

$$\begin{cases} x_1 = u \cos(u) + e_1 \\ x_2 = 20w + e_2 \\ x_3 = u \sin(u) + e_3 \\ x_4 = 5w + e_4 \\ x_5 = 0.5u + w + e_5 \\ x_6 = 3w + e_6 \end{cases} \quad (26)$$

where  $u$  and  $v$  are the latent variables, satisfy  $u \sim U(1.5\pi, 4.5\pi)$  and  $w \sim U(0,1)$  respectively,  $e$  are noise variables with standard deviation of 0.01. According to Eq. (26), 2000 normal samples are generated as training data for modeling, and 100 normal samples are generated for

validation. Generate 10 fault samples at different locations near the manifold structure for detection by setting the values of monitoring variables, where faults 2, 3, and 4 are weak faults closer to the manifold structure of training data. The distribution of the generated samples in the first three-dimensional is shown in Fig. 9.

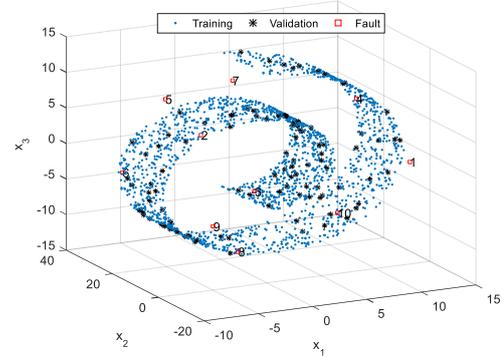


Fig. 9. Raw data scatter plot in the first three dimensions.

PCA, FD-KNN, RP-KNN, KPCA, and ILENS methods were used for fault detection in the process, respectively. PCA and KPCA choose the number of principal component subspaces (PCs) as 3 according to the cumulative contribution rate of 85%. The KPCA bandwidth  $\sigma=0.001$ . The number of nearest neighbors  $k$  of FD-KNN and RP-KNN is 8 and the parameter  $L$  of RP-KNN is 2. The dimension of the manifold feature space of ILENS is 2 and the dimension of the Normal Space is chosen to be 4, other parameters are set to  $\varepsilon=5$ ,  $k=13$ , and  $t=8$ . The confidence limits of the above five methods were set to 97%. Fig. 10 to 14 shows the detection diagrams of the five methods.

The detection results of PCA are shown in Fig. 10. The PCA method is unfavorable for the detection of nonlinear data. The  $T^2$  statistic has four faults that are not detected, and the SPE statistic is not detected at all. Fig. 11 shows the fault detection diagram of the KPCA method. KPCA has a better effect on fault detection for nonlinear data, but it still fails to separate faults that deviate from the manifold structure to a lesser extent from the normal samples after mapping the original data into a high dimensional space using the kernel function. Therefore, again, not all faults are detected. The detection diagram of FD-KNN is shown in Fig. 12, FD-KNN has a better effect on the nonlinear data processing, and most of the faults can be successfully detected, but faults 2, 3, and 4 are not detected. This is mainly due to the proximity of these three faults to the manifold structure, so the  $D^2$  statistic is not able to detect these three weak faults effectively. RP-KNN suffers from the same problem as FD-KNN leading to unsuccessful detection of faults 2, 3, and 4. Moreover, the random projection matrix of RP-KNN has uncertainty, which incorrectly projects faults 8 and 9 into normal samples, and thus faults 8 and 9 are also missed.

Fig. 14 shows the fault detection diagram of the ILENS method, all faults can be effectively detected. The weak faults 2, 3, and 4 are still very well detected. Although these faults are close to the normal samples, they are still some distance away from the manifold structure in which the sample set is

located. Therefore, in the Normal Space, these faults can be separated from the normal samples and detected by the SPE statistic.

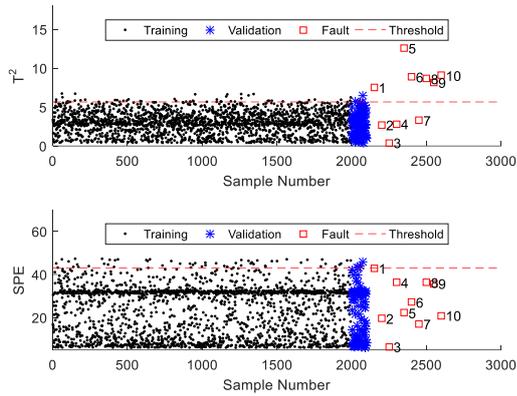


Fig. 10. Detection diagrams of the PCA.

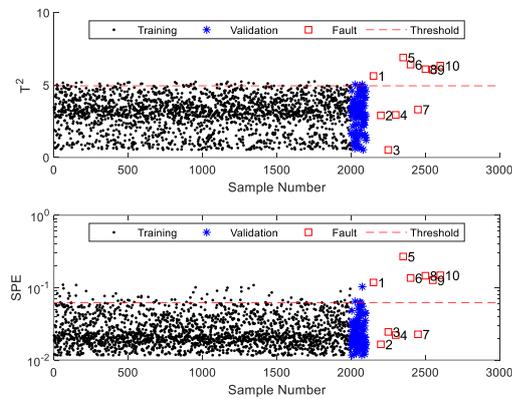


Fig. 11. Detection diagrams of the KPCA.

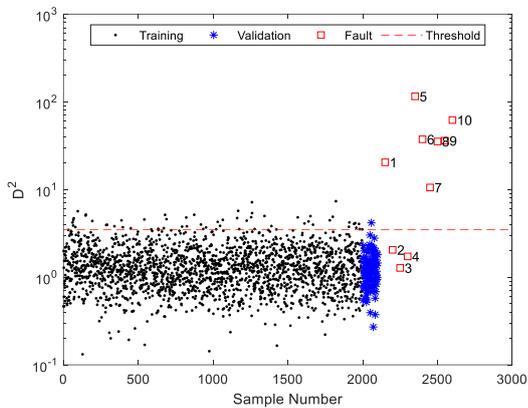


Fig. 12. Detection diagrams of the FD-KNN.

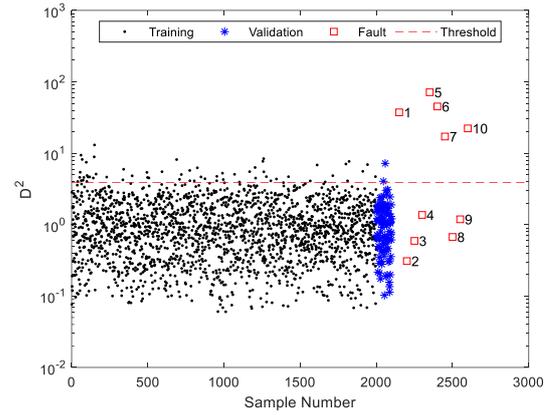


Fig. 13. Detection diagrams of the RP-KNN.

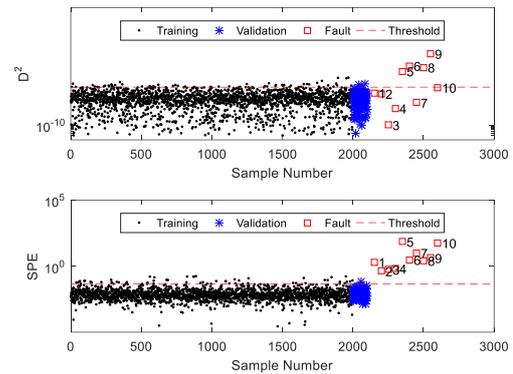


Fig. 14. Detection diagrams of the ILENS.

#### 4.2 TCSI simulation process

The realistic simulation testbed of a turbocharged spark-ignited engine system (TCSI) is an open-source analog simulation system designed and developed by Ng et al. through Matlab/Simulink (Ng et al., 2020). The user interacts with the test stand through a graphical interface, where the engine can be simulated using four industry-standard driving cycle procedures, including the Worldwide Harmonized Light Vehicle Test Procedure (WLTP), the New European Driving Cycle (NEDC), the Extra-Urban Driving Cycle (EUDC), and the U.S. Environmental Protection Agency Federal Test Procedure (FTP-75). The simulation system also enables the user to induce multiple faults and better understand the impact of faults on engine performance. Researchers can generate relevant data through this simulation system to develop and compare current and future fault diagnosis methods. The simulation system is stable and has been used in several fields for research (Stoumpos et al., 2022; Duan et al., 2023).

There are 9 variables in the TCSI simulation process, this paper selects all 9 variables as monitoring variables, and the relevant information of the variables is shown in Table 1.

**Table 1. Process Variables.**

No.	Variables
1	Compressor temperature (K)
2	Compressor pressure (Pa)
3	Intercooler temperature (K)
4	Intercooler pressure (Pa)
5	Intake manifold temperature (K)

6	Intake manifold pressure (Pa)
7	Air filter mass flow (kg/s)
8	Engine torque (N·m)
9	Exhaust manifold pressure (Pa)

The TCSI simulation system was run to obtain a normal batch of data for modeling, and the WLTP program was selected for the simulation test. The simulation process was 1800 seconds with a sampling interval of 1.0 seconds, and all remaining parameters used the system defaults.

By TCSI simulation system 7 faults are generated for detection and Table 2 shows the information description of the faults. Performed fault detection of the TCSI simulation process by

**Table 2. Fault information description.**

Fault	Description	Faults setting
F <sub>PaF</sub>	Loss of pressure in the air filter	20-kPa pressure drop
F <sub>WaF</sub>	Air leakage between the air filter and the compressor	20% of flow through leakage
F <sub>We</sub>	Air leakage between the compressor and the intercooler	20% of flow through leakage
F <sub>WiC</sub>	Air leakage between the intercooler and the throttle	20% of flow through leakage
F <sub>Wth</sub>	Air leakage after the throttle in the intake manifold	20% of flow through leakage
F <sub>Xth</sub>	Throttle position actuator error	Fault leading to 20% flow error
F <sub>YTic</sub>	Intercooler temperature sensor fault	20-K offset

**Table 3. Fault detection rate (%).**

No.	Fault	PCA		KPCA		FD-KNN	RP-KNN	ILENS	
		T <sup>2</sup>	SPE	T <sup>2</sup>	SPE	D <sup>2</sup>	D <sup>2</sup>	D <sup>2</sup>	SPE
1	f <sub>PaF</sub>	88.72	100	86.21	100	100	99.83	72.76	100
2	f <sub>WaF</sub>	11.34	98.44	12.56	98.00	98.39	91.94	60.20	97.55
3	f <sub>We</sub>	36.19	100	36.63	100	100	98.72	69.93	100
4	f <sub>WiC</sub>	4.61	37.69	5.06	25.35	52.25	48.25	11.45	98.17
5	f <sub>Wth</sub>	23.18	100	23.51	100	100	100	93.55	100
6	F <sub>Xth</sub>	3.06	44.30	3.72	39.08	42.13	42.02	11.17	71.04
7	F <sub>YTic</sub>	9.84	99.94	10.84	99.94	99.94	99.39	24.68	100

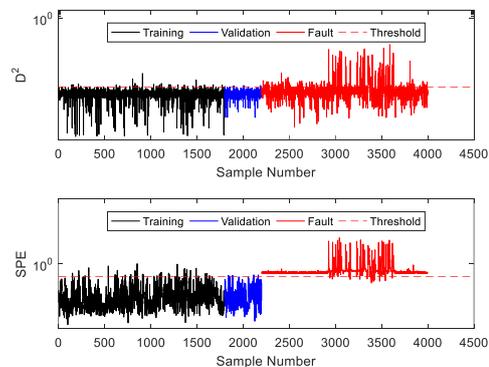
As shown in Table 3, the ILENS has a high detection rate for all faults. Especially for faults 4 and 6, which are the faults with small deviations, the detection rate of ILENS is significantly higher compared to other methods. For the two weak faults, PCA, KPCA, FD-KNN, and RP-KNN are all not detected well. PCA is less effective in processing nonlinear data and has a lower fault detection rate. Although KPCA can extract the nonlinear structure of the data, it is still unable to separate the faults from the normal samples after mapping the data to a high-dimensional space through the kernel function for faults with a small deviation, and thus is not effective in detecting faults 4 and 6. The control limits generated by the FD-KNN do not separate weak faults from normal samples and provide poor detection of weak faults. RP-KNN suffers from the same problem in this regard, and due to the randomness of the generation of the random projection matrix of RP-KNN, it is easy to categorize the fault data into normal data with a low detection rate.

For all 7 faults, ILENS has the high detection rate. For faults 4 and 6, which have smaller deviations, ILENS has a significantly higher detection rate compared to other methods. Taking fault 4 as an example, this fault produces violent fluctuations between 720 seconds and 1440 seconds, and the

PCA, KPCA, FD-KNN, RP-KNN, and ILENS methods. Both PCA and KPCA choose the number of PCS as 1 according to the cumulative contribution rate of 85%. The KPCA bandwidth  $\sigma=0.0002$ . Both FD-KNN and RP-KNN set the number of nearest neighbors  $k = 5$ , and the parameter  $L$  of RP is set to 2. The dimension of the manifold feature space of ILENS is chosen to be 2 from the manifold characteristics, and the dimension of the Normal Space is chosen to be 7, other parameters of the ILENS are set to  $\varepsilon = 32$ ,  $k = 17$ , and  $t = 5$ . Table 3 shows the fault detection results of the TCSI simulation process by PCA, KPCA, FD-KNN, RP-KNN, and ILENS methods.

fault in this period fluctuates more. This phase accounts for about 40% of the total time, and the magnitude of the fault outside this period is small and difficult to detect, so the detection rate of all four traditional detection methods is around 40%, which is a low detection rate.

Fig. 16 shows the distribution of normal data and fault 4 in two of the dimensions of the Normal Space.



**Fig.15 Fault fWic detection by ILENS**

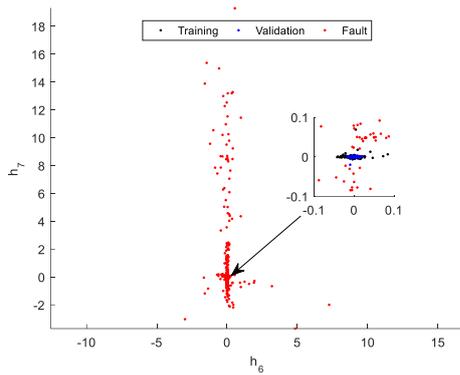


Fig.16. Distribution of data in two of the dimensions of the Normal Space.

Normal samples are located in the manifold structure that exists within the process data, with values close to zero in all dimensions in the Normal Space. In the case of weak faults, although the magnitude of the deviation is small, the fault samples are farther away from the manifold structure existing inside the process data, so the values in each dimension in the Normal Space are away from zero, and the faults can be effectively detected by the SPE statistic. The detection rate of ILENS for fault 4 is shown in Fig. 15, which shows that ILENS not only has a good detection effect when the fault fluctuation is large but also still has a high detection rate in the stage of small fault fluctuation. The total detection rate for fault 4 reaches 98.78%, which is much higher than the other four methods.

## 5. CONCLUSIONS

Aiming at the difficulty of embedding new samples added and the loss question of information when using Laplacian Eigenmaps for process monitoring, a fault detection method based on Incrementable Laplacian Eigenmaps and Normal Space (ILENS) is developed.

Both theoretical analysis and experimental results show that the ILENS has a higher detection rate compared with other traditional methods when performing fault detection, which is of reference significance for fault diagnosis in industrial production processes. In the process of industrial production, the accuracy of fault detection directly affects the maintenance efficiency of the equipment as well as the production of the enterprise. As a new fault detection method with good effect, the ILENS can be applied to industrial production to detect faults in time when they occur and maintain production safety.

## ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial supports by the National Science Foundation of China (62273242), Henan Science and Technology R&D Program Joint Fund (21ZD4GA028).

## REFERENCES

Abramov S., Travin, S., Duca, G., and Precup R. (2023). New Opportunities Model for Monitoring, Analyzing and Forecasting the Official Statistics on Coronavirus Disease Pandemic. *Romanian Journal of Information Science and Technology*, 26(1), 48-63.

- Breunig, M. M., Kriegel, H. P., Ng, R. T., and Sander, J. (2000, May). LOF: identifying density-based local outliers. *In Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 93-104.
- Belkin, M., and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6), 1373-1396.
- Cao, L. J., Chua, K. S., Chong, W. K., Lee, H. P., and Gu, Q. M. (2003). A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine. *Neurocomputing*, 55(1-2), 321-336.
- Chen, C., Zhang, L., Bu, J., Wang, C., and Chen, W. (2010). Constrained Laplacian Eigenmap for dimensionality reduction. *neurocomputing*, 73(4-6), 951-958.
- Deng, X., Tian, X., Chen, S., and Harris, C. J. (2016). Nonlinear process fault diagnosis based on serial principal component analysis. *IEEE transactions on neural networks and learning systems*, 29(3), 560-572.
- Donoho, D. L., and Grimes, C. (2003). Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10), 5591-5596.
- Duan, H., Yin, X., Kou, H., Wang, J., Zeng, K., and Ma, F. (2023). Regression prediction of hydrogen enriched compressed natural gas (HCNG) engine performance based on improved particle swarm optimization back propagation neural network method (IMPSON-BPNN). *Fuel*, 331, 125872.
- Fazai, R., Mansouri, M., Abodayeh, K., Nounou, H., and Nounou, M. (2019). Online reduced kernel PLS combined with GLRT for fault detection in chemical systems. *Process Safety and Environmental Protection*, 128, 228-243.
- Guo, J., Wang, X., and Li, Y. (2018). Application of KNN in fault isolation of chemical production process. *Application Research of Computers*, 35(04), 1117-1121.
- He, Q. P., and Wang, J. (2007). Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes. *IEEE transactions on semiconductor manufacturing*, 20(4), 345-354.
- Henseler, J., Ringle, C. M., and Sarstedt, M. (2016). Testing measurement invariance of composites using partial least squares. *International marketing review*, 33(3), 405-431.
- He, X., and Niyogi, P. (2003). Locality preserving projections. *Advances in neural information processing systems*, 16.
- Jiang, Q., Zhu, Q., Wang, B., and Guo, L. (2017). Nonlinear machine fault detection by semi-supervised Laplacian Eigenmaps. *Journal of Mechanical Science and Technology*, 31, 3697-3703.
- Kim, M., Jung, S., Kim, B., Kim, J., Kim, E., Kim, J., and Kim, S. (2022). Fault Detection Method via k-Nearest Neighbor Normalization and Weight Local Outlier Factor for Circulating Fluidized Bed Boiler with Multimode Process. *Energies*, 15(17), 6146.
- Lee, J. M., Yoo, C., and Lee, I. B. (2004). Fault detection of batch processes using multiway kernel principal component analysis. *Computers & chemical engineering*, 28(9), 1837-1847.

- Li, B., Li, Y. R., and Zhang, X. L. (2019). A survey on Laplacian eigenmaps based manifold learning methods. *Neurocomputing*, 335, 336-351.
- Navi, M., Davoodi, M. R., and Meskin, N. (2015). Sensor fault detection and isolation of an industrial gas turbine using partial kernel PCA. *IFAC-PapersOnLine*, 48(21), 1389-1396.
- Ng, K. Y., Frisk, E., Krysanter, M., and Eriksson, L. (2020). A realistic simulation testbed of a turbocharged spark-ignited engine system: A platform for the evaluation of fault diagnosis algorithms and strategies. *IEEE Control Systems Magazine*, 40(2), 56-83.
- Rosipal, R., and Trejo, L. J. (2001). Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of machine learning research*, 2(Dec), 97-123.
- Roweis, S. T., and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500), 2323-2326.
- Said, M., Abdellafou, K. B., and Taouali, O. (2020). Machine learning technique for data-driven fault detection of nonlinear processes. *Journal of Intelligent Manufacturing*, 31, 865-884.
- Sun, H., Guo, Y., and Zhao, W. (2020). Fault detection for aircraft turbofan engine using a modified moving window KPCA. *IEEE Access*, 8, 166541-166552.
- Saul, L. K., and Roweis, S. T. (2003). Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of machine learning research*, 4(Jun), 119-155.
- Stoumpos, S., and Theotokatos, G. (2022). A novel methodology for marine dual fuel engines sensors diagnostics and health management. *International Journal of Engine Research*, 23(6), 974-994.
- Tenenbaum, J. B., Silva, V. D., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500), 2319-2323.
- Yurtkan, K., Adalier, A., and Umut, T. E. K. G. (2023). Student Success Prediction Using Feedforward Neural Networks. *Romanian Journal of Information Science and Technology*, 26(2), 121-136.
- Zhou, Z., Wen, C., and Yang, C. (2014). Fault detection using random projections and k-nearest neighbor rule for semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing*, 28(1), 70-79.
- Zhang, Z., and Zha, H. (2004). Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM journal on scientific computing*, 26(1), 313-338.
- Zheng, Q., Liu, Q., Zhu, P., Shi, H., and Li, X. (2023). WELD LINE STRUCTURED LIGHT STRIPE CENTER EXTRACTION APPROACH BASED ON DATA DIMENSIONALITY REDUCTION. *Proceedings of the Romanian Academy, Series A*, 24(1), 93-102.