

# 基于 TSNS-RAE 的多模态过程故障检测

郭小萍, 李志远, 李元

(沈阳化工大学 信息工程学院, 沈阳 110142)

**摘要:** 根据多模态工业生产数据的特点, 提出基于时空近邻标准化和鲁棒自编码器 (TSNS-RAE) 的故障检测方法; TSNS 处理数据时同时考虑了样本的时间近邻和空间近邻, 可以消除数据动态性和多模态特征; 相比于普通的自编码器, 鲁棒自编码器提升了模型的抗噪性和鲁棒性, 具有更好的提取非线性特征的能力; TSNS-RAE 模型将原始数据空间分成模型空间和残差空间两部分, 选择残差空间的 SPE 统计量作为监控统计量, 通过数值案例和青霉素实验来验证 TSNS-RAE 的可行性。

**关键词:** 多模态; 故障检测; 鲁棒自编码器; 时空近邻标准化; 青霉素生产过程

## Multi-Modal Process Fault Detection Based on TSNS-RAE

GUO Xiaoping, LI Zhiyuan, LI Yuan

(School of Information Engineering, Shen yang University of Chemical Technology, Shenyang 110142, China)

**Abstract:** Aiming at the characteristics of multimodal industrial processes, A fault detection method based on time-space nearest neighborhood standardization and robust autoencoder (TSNS-RAE) is proposed. The TSNS processes the data by considering both temporal and spatial neighbors of samples, thus eliminating the data dynamics and multimodal features. Compared with ordinary autoencoders, robust autoencoders improve the noise resistance and robustness of the model, and have better ability to extract the non-linear features. The TSNS-RAE model divides the original data space into the model space and residual space, and the SPE statistics of residual space are selected as monitoring statistics. Numerical cases and penicillin experiments are used to verify the feasibility of the TSNS-RAE.

**Keywords:** multimode; fault detection; robust autoEncoder; TSNS; penicilin production

## 0 引言

随着现代工业过程生产规模的不断扩大以及生产过程复杂性的不断提高, 为了保证整个生产过程的安全性和可靠性, 及时有效的故障检测变得越发的重要, 对其的研究也具有重大意义<sup>[1-2]</sup>。

现代工业生产过程大多是多模态的工业过程。多模态工业过程同时还具有动态性和非线性的特征。因此相较于普通单模态过程, 多模态过程的监控要更加复杂。芯片制造过程<sup>[3]</sup>, 青霉素发酵过程<sup>[4]</sup>和半导体生产过程<sup>[5]</sup>是最常见的多模态工业过程。

复杂多模态过程的监控, 可以从数据标准化方法和故障检测方法两个方面进行研究。

多模态过程的一种监控策略是选择合适的数据标准化方法。目前最常用的数据标准化方法是 Z-Score 标准化方法, 但是该方法并不适用于多模态过程。局部近邻标准化 (NS) 的数据处理方法<sup>[6-7]</sup>的提出有效的解决了 Z-Score 不能处理多模态数据的缺陷, NS 方法虽然可以将多模态数据合并成单模态, 但是忽略了样本的时序相关性, 在处理动

态过程时检测效果不佳。为了进一步完善 NS 在处理动态性生产过程的不足, 冯等提出了时空近邻标准化的数据处理方法<sup>[8]</sup>, 该标准化方法不仅将多模态数据融合为单模态数据, 同时还能消除前后数据之间的动态相关性, 进而将故障数据与正常数据更好的分离开, 达到更好的故障检测效果。

多模态过程的另一种监控策略是选择合适的故障检测方法, 主成分分析 (PCA)<sup>[9]</sup>、偏最小二乘方法 (PLS)<sup>[10]</sup>和独立成分分析 (ICA)<sup>[11]</sup>是最常用的故障检测方法。这些方法都是基于数据驱动的传统故障检测方法。它们都是线性学习的降维方法, 在处理非线性数据时效果不佳。

近年来, 深度学习方法在图像分类、目标检测、自动驾驶等领域都有着深入的研究。相对于传统的基于数据驱动的方法, 深度学习模型可以提取多变量数据更深层次的数据特征, 表现出强劲的处理各种非线性信息的能力<sup>[12]</sup>。常见的深度学习方法有卷积神经网络<sup>[13]</sup>, 循环神经网络<sup>[14]</sup>, 自动编码器<sup>[15]</sup>。其中自编码器的应用更为广泛。目前常用的自编码器模型有堆叠自编码器 (DAE), 鲁棒自编码器 (RAE), 正交自编码器 (QAE), 变分自编码器 (VAE)。

本文选择时空近邻标准化方法 (TSNS) 为数据的标准

收稿日期:2022-12-01; 修回日期:2023-01-04。

基金项目:国家自然科学基金资助项目(61490701,61673279);辽宁省教育厅重点实验室项目(LJ2020021)。

作者简介:郭小萍(1972-),女,博士,教授。

李元(1964-),女,博士,教授。

引用格式:郭小萍,李志远,李元. 基于 TSNS-RAE 的多模态过程故障检测[J]. 计算机测量与控制, 2023, 31(9): 22-28.

化方法, 选择鲁棒自编码器 (RAE) 模型为故障的检测方法, 将两种方法进行结合, 提出基于 TSNS-RAE 的故障检测方法来对多模态工业生产进行故障检测。

## 1 数据的预处理

### 1.1 标准差标准化 (Z-Score)

在进行故障检测之前, 需要对采集的数据集进行预处理。针对工业过程的特点选择最合适的预处理方法来对数据进行标准化, 对之后的故障检测有着十分重要的意义。

目前默认最常使用的数据预处理方法是 Z-Score 标准化方法。Z-Score 标准化也称为标准差标准化, 该方法是基于全体数据特征的均值和标准差对数据进行的标准化。Z-Score 的步骤如下: 对于样本集  $X = \{x_1, x_2, \dots, x_n\}$ ,  $X \in \mathbf{R}^{n \times m}$ , 其中  $x_i$  是特征变量  $x_i = \{x_i^1, \dots, x_i^j, \dots, x_i^m\}$ ,  $i = 1, 2, \dots, n, j = 1, 2, \dots, m$ 。利用式 (1) 和式 (2) 计算特征  $x_i$  的均值  $\mu_i$  和标准差  $\sigma_i$ , 然后利用式 (3) 将特征  $x_i$  标准化为  $\bar{x}_i$ 。

$$\mu_i = \frac{1}{m} \sum_{j=1}^m x_i^j \quad (1)$$

$$\sigma_i = \sqrt{\frac{1}{m} \sum_{j=1}^m (x_i^j - \mu_i)^2} \quad (2)$$

$$\bar{x}_i = \frac{x_i - \mu_i}{\sigma_i} \quad (3)$$

Z-score 标准化使样本集的每个特征具有均值为 0 标准差为 1 的特点, 消除了数据量纲的影响, 使不同变量的方差 (离散程度) 相同。普通的 Z-Score 标准化方法在处理多模态数据时, 无法将多模态数据转换成单模态数据, 导致检测效果不佳。

### 1.2 邻域标准化 (NS)

为改善 Z-Score 方法处理多模态数据时的不足。Ma 等提出了邻域标准化方法 (NS, neighborhood standardized), 该方法使用最近邻样本的近邻集标准化当前样本, NS 的步骤如下: 首先寻找样本  $x$  的最近邻样本  $x^1$ , 其次寻找  $x^1$  的前  $N$  近邻样本集, 计算此近邻集的均值  $m(N(x^{(1)}))$  和标准差  $s(N(x^{(1)}))$ , 最后使用式 (4) 对  $x$  进行标准化:

$$\bar{x} = \frac{x - m(N(x^{(1)}))}{s(N(x^{(1)}))} \quad (4)$$

邻域标准化方法 (NS) 虽然可以将多模态数据转换成单模态数据, 但是该方法只考虑了空间上的近邻信息, 忽略了时间上的近邻信息。所以 NS 在处理动态过程的工业数据时, 往往忽略了数据的动态性, 无法消除之前时刻样本对后面时刻样本的影响。无法保证所用的近邻样本应来自于样本的相近时刻。

### 1.3 时空近邻标准化 (TSNS)

为了进一步改善 NS 方法的不足, 冯等提出了 TSNS 方法。TSNS 的步骤如下: 首先在训练集中寻找样本  $x$  在时间方向上的前  $N$  近邻样本集  $N(x) = \{x_1^{(x)}, \dots, x_i^{(x)}, \dots, x_N^{(x)}\}$ , 再寻找  $x_i^{(x)}$  在空间方向上的前  $K$  近邻样本集  $N(x_i^{(x)}) = \{x_1^{(x)}, \dots, x_f^{(x)}, \dots, x_K^{(x)}\}$ , 使用式 (5) 和 (6) 计

算此样本集的均值  $m(N(x_i^{(x)}))$  和标准差  $s(N(x_i^{(x)}))$ , 最后使用式 (7) 将  $x$  进行标准化处理得到标准化之后的样本。

$$m(N(x_i^{(x)})) = \frac{1}{K} \sum_{f=1}^K x_f^{(x)} \quad (5)$$

$$s(N(x_i^{(x)})) = \sqrt{\frac{1}{K-1} \sum_{f=1}^K (x_f^{(x)} - m(N(x_i^{(x)})))^2} \quad (6)$$

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N \frac{x - m(N(x_i^{(x)}))}{s(N(x_i^{(x)}))} \quad (7)$$

TSNS 方法对数据进行标准化时不仅单纯使用空间近邻信息, 还充分考虑时间序列信息的影响, 使用来自于相近时刻的样本近邻集。经过 TSNS 处理之后的样本在消除数据多模态的同时也消除了动态性, 能够实现故障与正常样本在空间上的分离。为之后对过程数据进行故障检测奠定基础。

## 2 基于鲁棒自编码器 (RAE) 的故障检测

### 2.1 普通自编码器 (AE) 原理

普通的自动编码器 (AE) 是一种常用的降维神经网络, 经常用于数据的降维和特征提取。它分为编码器和解码器两部分, 其结构如图 1 所示。在对原始数据进行数据降维处理和特征提取时, 需保持输出层数据维数与输入层输入维数相同, 而中间隐藏层数据维数则可以根据实际情况而定。

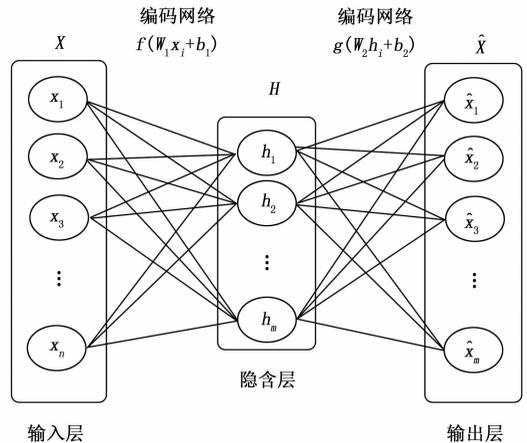


图 1 自编码器网络结构示意图

自编码器由编码和解码两步构成, 输入层原始数据经过编码过程进行降维处理, 在隐藏层得到数据的特征, 然后对隐藏层特征进行解码在输出层得到原始数据的重构。自编码器模型结构如下式:

$$h_i = f(W_1 x_i + b_1) \quad (8)$$

$$r(x_i) = \hat{x}_i = g(W_2 h_i + b_2) \quad (9)$$

式中,  $i = 1, 2, 3, \dots, I$  是采样时间,  $x_i \in \mathbf{R}^{J \times 1}$  是第  $i$  个样本,  $h_i \in \mathbf{R}^{K \times 1}$  是原始输入  $x_i$  的特征。  $W_1 \in \mathbf{R}^{K \times J}$  和  $b_1 \in \mathbf{R}^{K \times 1}$  是输入层到隐藏层之间的投影权重和偏置。  $r(x_i) \in \mathbf{R}^{J \times 1}$  是对提取特征  $h_i$  的解码或者说是输入数据  $x_i$  的重构。  $W_2 \in \mathbf{R}^{J \times K}$  和  $b_2 \in \mathbf{R}^{J \times 1}$  是隐藏层到输出层的投影权重和偏置。

公式 (8) 和 (9) 中  $f(i), g(i)$  是隐藏层和输出层的激活函数, 根据实际工业生产过程的特点, 激活函数的选择如下:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (10)$$

$$g(x) = x \quad (11)$$

Sigmoid 函数  $f(x) = \frac{1}{1 + e^{-x}}$  是一种非线性函数, 选择它作为编码过程的激活函数, 可以更好的提取过程数据非线性特征。恒等函数  $g(x) = x$  是一种线性函数, 选择它作为解码过程的激活函数可以保证解码出的重构数据范围是和原始数据一样的范围, 方便之后损失函数的计算<sup>[16]</sup>。

一个好的自编码器模型在提取数据特征时, 应保持输出端重构数据  $r(x)$  与输入端原始数据  $X$  的差异尽可能的小。这个差异就叫作损失函数就是我们需要最小化的目标函数。普通自编码器的目标函数如下:

$$L = \frac{1}{I} \sum_{i=1}^I \|r(x_i) - x_i\|_2^2 \quad (12)$$

式中,  $x_i$  是第  $i$  个样本数据点,  $r(x_i)$  是第  $i$  个样本的重构值。通过累积贡献率方法选择好好隐藏层的节点数  $K$ , 设置好训练次数之后, 可以选择 Adam 优化算法来对编码器模型进行训练, 当损失函数值慢慢下降并最后收敛趋于稳定之后, 编码器的训练阶段完成。得到最小化的损失函数和自编码器模型参数值。我们可以将此训练好的编码器用于之后的故障检测之中。

自编码器通过提取数据的深层特征来表示原始的过程数据。自编码器模型训练的效果与模型复杂度和待训练的数据集的质量有着密切的联系。当训练集数据量较少或训练集包含噪声的时候, 训练出来的模型容易出现过拟合现象, 即测试集数据在模型上的表现效果远差于训练集在模型上的表现效果。为了提高训练数据集质量, 在训练之前需要把正常数据与噪声隔离开。但是单纯的把噪声和冗余从训练集数据中剔除是十分困难的做法。

### 2.2 鲁棒自编码器 (RAE) 原理

为了避免编码器因为训练数据量较少或噪声干扰在模型训练时出现的过拟合情况, 对普通的自编码器 (AE) 进行改造, 提出鲁棒自编码器 (RAE) 的概念。

鲁棒自编码器相比于普通编码器, 能够将被噪声污染的原始数据进行编码解码并且很好的提取数据特征和输出原始数据的重构。这样的编码器更有利于处理复杂多变的过程数据。

因此在训练模型之前可以人为的给训练集加上高斯噪声, 达成对数据的损坏, 编码器如果可以从人为损坏的训练数据集提取出数据特征, 并且重构出原始数据, 那么这个编码器就便具有识别噪声的能力, 具有抗噪性, 有更好的鲁棒性, 从而提高模型的泛化能力。

RAE 就是通过对输入数据进行一定程度的损坏, 通过训练来将其还原为原始数据空间来学习数据中关键特征的自编码器。

RAE 模型的示意图如图 2 所示, 和普通自编码器有着相似的结构<sup>[17]</sup>。

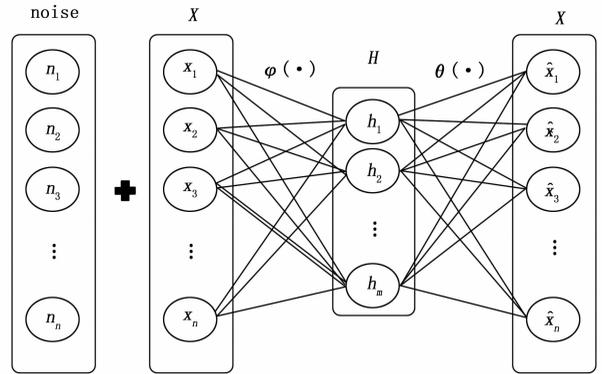


图 2 鲁棒自编码器示意图

RAE 与普通 AE 的区别就是训练阶段时在原始数据基础上人为加入高斯噪声  $\epsilon$ , 用加入噪声之后的数据来作为模型输入数据, 而模型的训练目标依旧是未加入噪声的原始数据, 鲁棒自编码器的目标函数如式 (13):

$$L_{RAE} = \frac{1}{I} \sum_{i=1}^I \|r(x_i + \epsilon_i) - x_i\|_2^2 \quad (13)$$

在理想情况下, 公式 (13) 可以被等效成一个加了正则化项的自编码器:

$$L_{RAE} = \int \|r(x + \epsilon) - x\|_2^2 p(x) dx = \int \left( \|r(x) - x\|_2^2 + \sigma_\epsilon^2 \left\| \frac{\partial r(x)}{\partial x} \right\|_F^2 \right) p(x) dx \quad (14)$$

式中,  $\sigma_\epsilon^2$  为所加高斯噪声  $\epsilon$  的方差,  $r(x)$  是对输入样本的重构,  $p(x)$  是原始数据的概率密度函数。

RAE 模型的训练过程与 AE 类似。分别选择 sigmoid 函数和恒等函数作为 AE 神经网络的激活函数。通过累积贡献率方法选择好隐藏层的节点个数  $K$ , 设置好训练次数。选择 Adam 算法作为优化器来加速神经网络的训练。将原始训练数据  $x$  加入噪声之后送入 AE 进而形成 RAE 模型。RAE 模型随着训练次数的增加, 损失函数越来越小, 编码器神经网络的权值参数与偏置参数也在不断的更新, 当损失函数的值慢慢下降并最后收敛趋于稳定之后, RAE 的训练阶段完成。得到最小化的损失函数和 RAE 模型参数值。

RAE 模型训练完成之后, 原始训练数据  $x$  被分成输入数据的重构  $r^*(x)$  和重构误差  $\xi^*(x)$  两部分。分别对应于 RAE 的模型空间和残差空间。如式 (15) 所示:

$$x = r^*(x) + \xi^*(x) \quad (15)$$

$r^*(x)$  和  $\xi^*(x)$  分别对应于输入向量  $x$  的输出重构值和残差。在训练阶段,  $x$  在模型空间的投影就是模型输出  $r^*(x)$ , 而在残差空间的投影就是模型重构误差  $\xi^*(x)$ , 与测量噪声  $\epsilon$  成正比关系。

### 2.3 RAE 故障检测方法

RAE 模型训练完成之后, 将未加入噪声的原始数据  $x$  送入离线训练好的 RAE 模型中得到输出的重构  $r^*(x)$ 。训练好的 RAE 模型将原始数据空间分为模型空间和残差空

间, 其中模型空间对输入变化具有鲁棒性, 而残差空间对输入变化敏感。所以在残差空间选择 SPE 统计量作为故障监测统计量。RAE 模型的训练数据 SPE 统计量为

$$SPE_R = (r^*(x) - x)^T (r^*(x) - x) \quad (16)$$

选择核密度估计法 (KDE) 来计算残差空间统计量  $SPE_R$  的阈值  $SPE\_lim$ , 这个阈值就是之后测试阶段判断是否存在故障的控制限。

### 3 基于 TSNS-RAE 的故障检测方法

将 TSNS 与 RAE 进行结合, 形成 TSNS-RAE 故障检测方法。TSNS-RAE 故障检测流程图如图 3 所示。TSNS-RAE 模型故障检测主要分为离线建模和在线监测两个阶段。

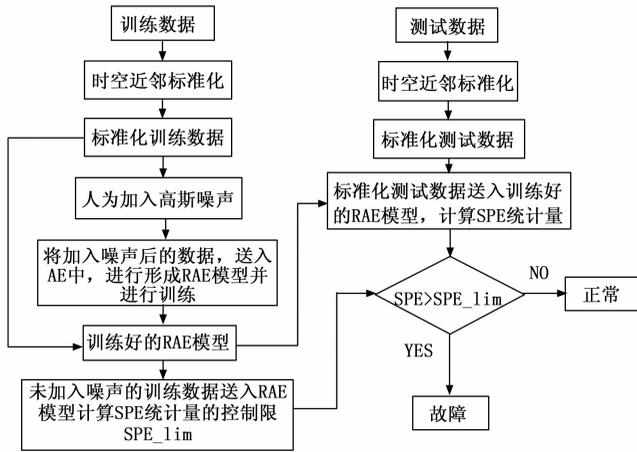


图 3 TSNS-RAE 故障检测流程图

离线建模阶段和在线监测的步骤如下。

离线建模过程:

- 1) 对正常工况运行下的原始数据进行时空近邻标准化处理得到标准样本  $X$ 。
- 2) 对时空近邻标准化之后的标准样本  $X$  人为加上方差是  $\sigma_e^2$  的高斯噪声, 得到污染数据  $\hat{X} = X + E, \sigma_e^2$  一般取 0.5 以下的值。
- 3) 将  $\hat{X}$  送入 AE 进而形成 RAE 模型, 对 RAE 进行训练, 当训练过程的损失函数最后收敛于一个最小值时, 完成 RAE 模型的训练, 得到训练好的 RAE 模型。
- 4) 将未加入噪声污染的标准样本  $X$  送入训练好的 RAE 模型之后, 在残差空间计算重构误差从而求得  $SPE$  统计量。
- 5) 利用核密度估计方法求  $SPE$  统计量的控制限  $SPE\_lim$ 。

在线监测过程:

- 1) 从在线过程获得新的测试集数据并运用 TSNS 进行预处理得到  $X_i$ 。
- 2) 将  $X_i$  直接送入离线过程训练好的 RAE 模型并获得输出的重构值  $r^*(X_i)$ 。
- 3) 计算在线监测时的  $SPE$  统计量并检查其是否超过离线过程设定的控制限  $SPE\_lim$ 。若超过控制限就表示发生

故障, 若没有超限则表示正常。

### 4 数值案例仿真

使用一个动态非线性的数值案例过程来进行仿真。通过  $z$ -score, NS, TSNS 这 3 种标准化方法的纵向对比来证明 TSNS 预处理对多模态过程检测有着重要作用。数值案例过程有两个输入变量  $\theta, t$  和两个输出变量  $x_1, x_2$ , 模型如下:

$$\begin{cases} x_1 = e^{at} \cos\theta + e_1, \\ x_2 = e^{at} \sin\theta + e_2, \\ \theta: 0 \sim 4\pi, t: 0 \sim 4\pi, \end{cases} \quad (17)$$

其中:  $a$  为过程参数, 本文取为 0.3,  $e_1$  和  $e_2$  为随机噪声。过程正常运行产生 500 个样本作为建模数据。再次, 让过程正常运行, 但从  $2\pi$  时刻起在变量  $t$  上添加幅值为 -3 的阶跃型扰动信号, 共产生 500 个样本作为测试数据。分别使用  $z$ -score 标准化方法, NS 标准化方法, 和 TSNS 标准化方法对训练数据和测试数据进行标准化处理, 其中 NS 标准化的近邻数  $N=4$ , TSNS 标准化的时间近邻数  $K=4$ , 空间近邻数  $N=5$ , 数据经过 3 种方法标准化之后的结果如图 4 所示。

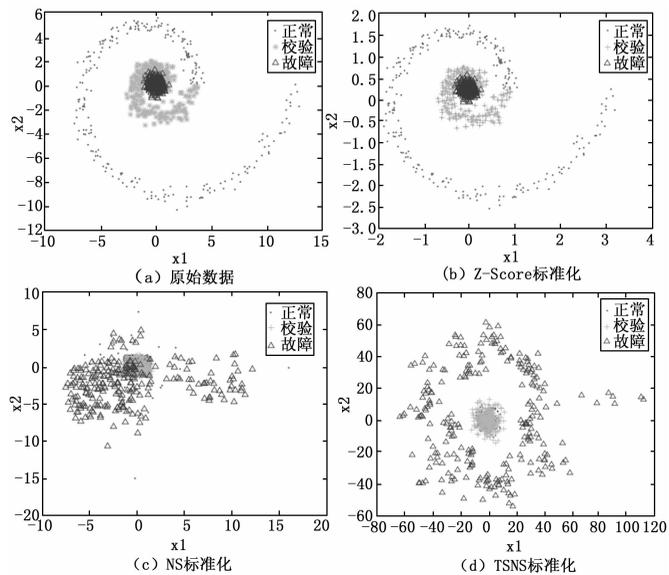


图 4 3 种标准化方法对多模态数值例子的处理结果

通过对图三中的几种标准化方法相比较可以发现, 普通的  $Z$ -Score 标准化方法无法将故障数据与正常数据进行分离。NS 标准化方法只考虑空间上的近邻, 忽略时间上的近邻, 无法消除过程动态性, 所以一部分故障可以与正常样本分离, 而另一部分故障则没有分离出来。TSNS 标准化方法同时考虑了时间和空间上的近邻, 并且可以完全将故障样本与正常样本实现分离。

完成了对训练数据的标准化处理之后, 我们给标准化的数据额外加上高斯噪声送入 AE 中形成 RAE 模型并进行训练, 隐藏层的维数设置为 1, 隐藏层的激活函数选择 sigmod 函数, 输出层激活函数选取恒等函数。优化算法选择

为 Adam 优化算法, 训练次数设置为 1 000 次。用训练数据训练好 RAE 模型之后, 根据核密度估计法在残差空间求 SPE 统计量的控制限  $SPE\_lim$ , 然后将校验数据和故障数据合并当成测试数据集进行标准化处理之后送入 RAE 模型, 得到测试集的 SPE 统计量并和  $SPE\_lim$  进行比较进而监测故障, 为了方便观测图形, 我们算 SPE 相关统计量时按照以 10 为底取其对数。3 种标准化方法的故障检测图如图 5 所示。

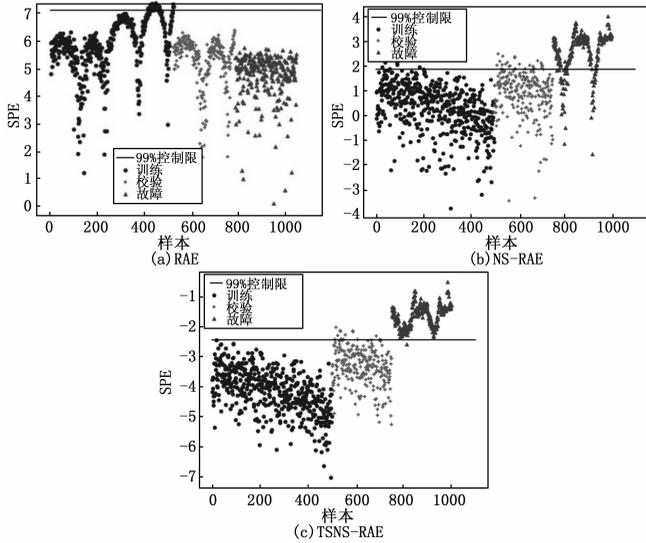


图 5 3 种方法对多模态数值例子故障检测结果

表 1 多模态数值例子的检测结果对比

| 实验方法  | RAE | NS-RAE | TSNS-RAE |
|-------|-----|--------|----------|
| 统计量   | SPE | SPE    | SPE      |
| 检测率/% | 0   | 80.6   | 99.6     |
| 误报率/% | 0   | 15.2   | 5.2      |

表 1 是 RAE、NS-RAE、TSNS-RAE 三种方法对多模态数值例子的故障检测误报率和检测率的结果对比。从表 1 中可以看出, Z-Score 标准化之后的 RAE 检测结果最差, 虽然没有误报, 但是也没有检测出任何故障。NS-RAE 故障检测效果较 RAE 有一定的提升。而 TSNS-RAE 方法的故障检测效果比 RAE 和 NS-RAE 方法的故障检测效果都要好, 通过这 3 种方法的纵向对比可得知 TSNS-RAE 方法有较低的误报率和更高的检测率, 更好地验证了 TSNS 在处理复杂多模态数据时的优越性。

### 5 青霉素生产过程数据仿真

青霉素生产发酵过程是一种具有时变性, 不同操作阶段动态有不同特性, 线性化程度不同以及非高斯性特点的典型的的多模态生产过程。青霉素发酵过程中的代谢变化分为菌体生长, 青霉素合成和菌体自溶 3 个阶段<sup>[18]</sup>。图 6 给出了青霉素发酵过程工艺流程图。

本文采用 pensim 仿真平台生产青霉素数据, 基于

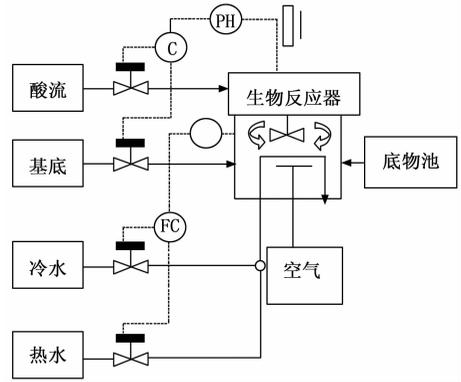


图 6 青霉素发酵过程工艺流程图

RAE, NS-RAE, TSNS-RAE 算法进行故障检测。pensim 仿真平台是由 Cinar 教授为首的研究小组创建开发的一款仿真软件, 专为研究青霉素发酵过程而设计。pensim 仿真平台在故障检测研究中有着十分广泛的应用<sup>[19]</sup>。同时由于其平台数据具有多模态过程数据的特点, 为验证深度学习方法在多模态过程故障检测与诊断方面带来了很多便利。该平台有 5 个输入变量控制发酵过程参数变化, 9 个过程变量是菌体合成及生长中产生的, 5 个质量变量影响青霉素的产量<sup>[20]</sup>。设定仿真时间为 400 h, 采样间隔为 0.5 h, 先在正常工况下产生一组数据作为离线训练样本。青霉素发酵过程可以在通气率、搅拌功率和底物流速率这 3 个变量上引入干扰, 分别有斜坡和阶跃两种干扰模式, 通过调整两种模式的幅度、干扰引入时间和终止时间来产生故障。本文采取了青霉素发酵过程中的 12 个主要变量进行实验, 同时设置了 6 种不同类型的故障作为待检测样本来验证所提 TSNS-RAE 算法的有效性, 故障类型如表 2 所示。

表 2 青霉素生产过程的 6 种故障类型

| 故障编号 | Fault1 | Fault2 | Fault3 | Fault4 | Fault5 | Fault6   |
|------|--------|--------|--------|--------|--------|----------|
| 故障变量 | 1      | 1      | 2      | 2      | 3      | 3        |
| 故障类型 | 阶跃     | 斜坡     | 阶跃     | 斜坡     | 阶跃     | 斜坡       |
| 故障幅值 | 1.5%   | 1L/h   | 1%     | 2 L/h  | 1%     | 0.01 L/h |
| 引入时间 | 150    | 150    | 150    | 150    | 150    | 150      |
| 结束时间 | 400    | 400    | 400    | 400    | 400    | 400      |

基于数据驱动的故障检测最常用的检测方法是主成分分析 (PCA) 算法和 K 近邻 (KNN) 算法。将这两种检测方法和 TSNS 相结合, 形成 TSNS-PCA 和 TSNS-KNN 算法。选用 TSNS-RAE 算法来进行故障检测, 并将其检测结果与 TSNS-PCA, TSNS-KNN 进行横向比较。TSNS 方法的时间近邻个数  $N=5$ , 空间近邻个数  $K=5$ 。这些故障检测方法均以置信水平为 99% 来确定监测模型的控制限; 其中 PCA 依据 85% 的累计贡献率来选取数据降维之后的主元个数。KNN 算法的近邻个数  $k=5$ 。RAE 模型依据 85% 的累计贡献率选取编码器隐藏层节点个数为 6; 激活函数依旧选择 Sigmoid 函数和恒等函数, 选择 Adam 算法为优化器加速 RAE 的训练。RAE 的训练次数设置为 1 500。完成编码器的建模与故障检

测之后。我们选择故障 3 和故障 6 的检测结果来作图, 为了更好的图形显示效果, SPE 相关统计量按照以 10 为底取对数。检测效果图如图 7 和图 8 所示。测试所用的不同方法的故障检测率和误报率均列在表 3 和表 4 中。

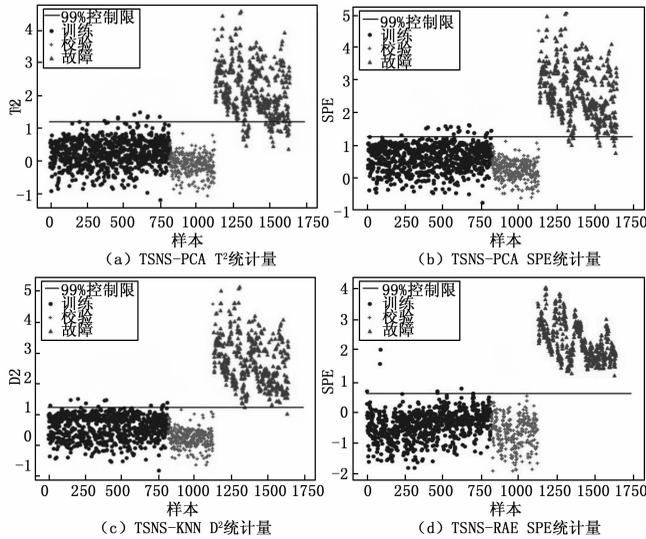


图 7 不同方法对故障 3 的检测结果图

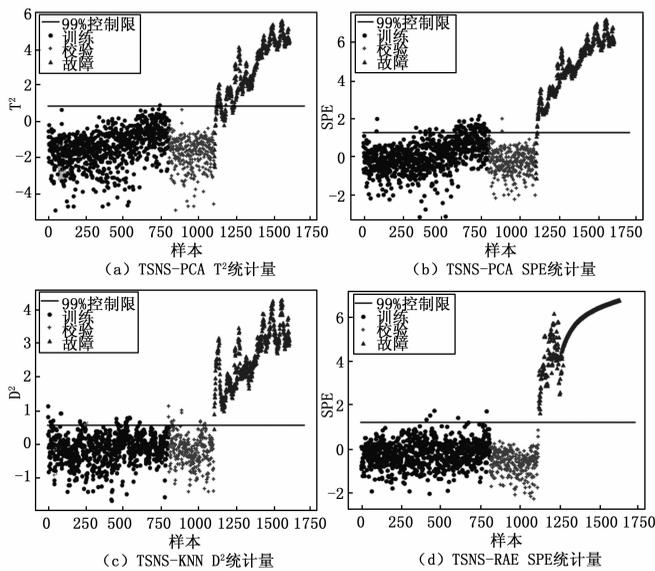


图 8 不同方法对故障 6 的检测结果图

表 3 不同方法的故障检测率/%

| 检测方法   | TSNS-PCA |      | TSNS-KNN | TSNS-RAE |
|--------|----------|------|----------|----------|
| 统计量    | $T^2$    | SPE  | $D^2$    | SPE      |
| Fault1 | 87.6     | 98.8 | 98.4     | 99.1     |
| Fault2 | 85.0     | 98.2 | 100      | 100      |
| Fault3 | 90.0     | 96.8 | 99.4     | 100      |
| Fault4 | 83.4     | 94.6 | 99.0     | 99.6     |
| Fault5 | 90.2     | 98.0 | 98.2     | 100      |
| Fault6 | 87.4     | 98.4 | 99.8     | 100      |

表 4 不同方法的故障误报率 %

| 检测方法   | TSNS-PCA |      | TSNS-KNN | TSNS-RAE |
|--------|----------|------|----------|----------|
| 统计量    | $T^2$    | SPE  | $D^2$    | SPE      |
| Fault1 | 0.33     | 0.66 | 1.66     | 0.66     |
| Fault2 | 0        | 0.33 | 0.33     | 0        |
| Fault3 | 0.33     | 0.33 | 0.33     | 0.33     |
| Fault4 | 0.33     | 0.66 | 1.66     | 0.33     |
| Fault5 | 0.33     | 0.66 | 0.66     | 0.33     |
| Fault6 | 0        | 0.66 | 3.66     | 0        |

PCA 作为最常用的故障检测方法, 将过程数据分成降维之后的主成分空间和残差空间, 分别对应  $T^2$  统计量和 SPE 统计量, PCA 原本只适用于单模态正态分布的数据, TSNS 方法与 PCA 的结合消除了青霉素生产过程的多模态和动态性提升了 PCA 的检测效果。但是 PCA 无法考虑过程的非线性, 所以 TSNS-PCA 的检测率与 TSNS-KNN 和 TSNS-RAE 相比效果不是很好。由故障 3 和故障 6 的实验结果图可以更直观的看出检测率较低。

KNN 检测方法使用过程数据样本与其前  $k$  个近邻样本的距离平方和  $D^2$  作为统计量, KNN 方法考虑了工业过程的非线性, 与 TSNS 方法结合之后对于多模态故障检测有着不错的检测效果, 在检测率上会比 PCA 方法更好一点, 误报率较高, 故障 1, 故障 4 和故障 6 有着很高的误报率。由故障 6 的检测结果图可以更直观的看出校验数据存在着误报。结合检测率与误报率, 整体的检测效果不如 TSNS-RAE。

RAE 作为深度学习方法, 与传统的基于数据驱动的方法相比, 有着更好的特征提取和数据降维效果, 训练模型的损失函数越小, RAE 对原始数据的重构越效果好。同时 RAE 模型有着抗噪性和鲁棒性。对青霉素数据使用 TSNS-RAE 方法进行故障检测, 基本都有较好的故障检测效果。由表 3 可知, 故障 2, 故障 3, 故障 5 和故障 6 的 SPE 统计量的检测率都达到了 100%, 故障 1 和故障 4 的 SPE 统计量的检测率分别达到 99.1% 和 99.6%。由表 4 可知整体的误报率都比较低。因此结合整体的检测率与误报率, 通过对 TSNS-PCA、TSNS-KNN、TSNS-RAE 三种方法进行横向对比可知 TSNS-RAE 对于复杂多模态过程有着更好的检测效果。

## 6 结束语

根据多模态工业生产数据的特点, 本文提出了一种基于 TSNS-RAE 的故障检测方法。通过数值案例, 将 TSNS 方法与常用的 Z-Score 方法、NS 方法进行对比, 突出 TSNS 方法在预处理复杂多模态数据时的优势。通过青霉素生产过程实验, 将 TSNS-RAE 与 TSNS-PCA、TSNS-KNN 进行横向对比, 结合检测率与误报率, 突出了 TSNS-RAE 在处理复杂多模态工业过程的优越性。

自动编码器除了鲁棒自编码器之外, 还有堆叠自编码器, 正交自编码器, 变分自编码器等。每一种编码器都有

各自的优势, 今后的实验可以针对不同的工业过程将 TSNS 方法与其他编码器模型进行结合进行故障检测。

参考文献:

[1] ZHENG D L, ZHOU L, SONG Z H. Kernel Generalization of Multi-Rate Probabilistic Principal Component Analysis for Fault Detection in Nonlinear Process [J]. IEEE/CAA Journal of Automatica Sinica, 2021, 8 (8): 1465 - 1476.

[2] ZHOU B, GU X. Multi-block statistics local kernel principal component analysis algorithm and its application in nonlinear process fault detection- ScienceDirect [J]. Neurocomputing, 2020, 376: 222 - 231.

[3] QIN Y, ZHAO C, GAO F. Multi-stage Process Analysis and Modelling Based Online Monitoring for ChipPackaging Process [J]. IFAC-Papers On Line, 2015, 48 (28): 993 - 998.

[4] 齐咏生, 王 普, 高学金. 基于核主元分析-主元分析的多阶段间歇过程故障监测与诊断 [J]. 控制理论与应用, 2012, 29 (6): 754 - 764.

[5] HE Q P, WANG J, ZHAO J. Fault Detection Using the k nearest Neighbor Rule for Semiconductor Manufacturing Processes [J]. IEEE Transactions on Semiconductor Manufacturing, 2007, 20 (4): 345 - 354.

[6] MA H, HU Y, SHI H B. A novel local neighborhood standardization strategy and its application in fault detection of multimode processes [J]. Chemometrics & Intelligent Laboratory Systems, 2012, 118 (7): 287 - 300.

[7] MA H, HU Y, SHI H B. Fault detection and identification based on the neighborhood standardized local outlier factor method [J]. Industrial & Engineering Chemistry Research, 2013, 52 (6): 2389 - 2402.

[8] 冯立伟, 李 元, 张 成, 等. 基于时空近邻标准化和局部离群因子的复杂过程故障 [J]. 控制理论与应用, 2020, 37 (3): 651 - 657.

[9] 南 林, 唐 翔, 王佩硕, 等. 基于 LabWindows/CVI 的数据采集与显示系统 [J]. 工业控制计算机, 2018, 31 (8): 6 - 7.

[10] 夏 磊. 基于 LabWindows/CVI 的分布式大气数据系统测试平台设计与实现 [D]. 成都: 电子科技大学, 2019.

[11] 孙晓阳, 苏新彦, 姚金杰, 等. 基于 LabWindows /CVI 的警犬状态监测系统 [J]. 国外电子测量技术, 2018, 37 (11): 109 - 113.

[12] 陈 娟, 张 建, 胡贵明. 基于 LabWindows/CVI 的导弹舵机测控系统设计 [J]. 计算机测量与控制, 2017, 25 (6): 86 - 88 .

[13] 倪笑宇, 徐 军, 戴美魁, 等. LabWindows/CVI 多线程技术在油门测试软件中的应用 [J]. 自动化与仪表, 2015, 30 (5): 62 - 64.

[14] 陶小亮, 牛 振. LabWindows/CVI 多线程技术在舵机测试软件中的应用 [J]. 中国测试, 2011, 37 (1): 81 - 83.

[15] 高 珺. 基于横向脉动阵列的 RNN 前向传播硬件加速实现 [D]. 南京: 南京大学, 2021: 34 - 38.

[9] 周东华, 李 钢, 李 元. 数据驱动的工业过程故障诊断技术 [M]. 北京: 科学出版社, 2011.

[10] KRESTA J V, MACGREGOR J F, MARLIN T E. Multivariate Statistical Monitoring of Process Operating Performance [J]. Canadian Journal of Chemical Engineering, 1991, 69 (1): 35 - 47.

[11] 陈国金, 梁 军, 钱积新. 独立元分析方法及其在化工过程监控和故障诊断中的应用 [J]. 化工学报, 2003, 54 (10): 1474 - 1477.

[12] LIU Y, FENG X, ZHOU Z. Multimodal video classification with stacked contractive autoencoders [J]. Signal Processing, 2016, 120 (4): 761 - 766.

[13] 王连云, 陶洪峰, 徐 琛, 等. 基于多层训练干扰的 CNN 轴承故障诊断 [J]. 控制工程, 2022, 29 (9): 1653 - 1657.

[14] 刘奕辰, 范 成, 刘旭媛, 等. 基于循环神经网络的冷水机组故障诊断方法 [J]. 建筑科学, 2022, 38 (8): 161 - 170

[15] 苏靖涵, 张 潇. 基于深度迁移自编码器的变工况下滚动轴承故障诊断方法 [J]. 计算机测量与控制, 2021, 29 (7): 85 - 90.

[16] 蒋 立, 宋执环. 基于自编码器模型的非线性过程检测 [D]. 杭州: 浙江大学, 2017, 10.

[17] 冯倩玉, 翁正新. 基于降噪正交自编码器的 TE 过程故障诊断 [D]. 上海: 上海交通大学, 2018, 5.

[18] PENG K X, MA L, ZHANG K. Review of Quality-related Fault Detection and Diagnosis Techniques for Complex Industrial Processes [J]. Acta Automatica Sinica, 2017, 43 (3): 349 - 365.

[19] 刘 毅, 王海清. Pensim 仿真平台在青霉素发酵过程的应用研究 [J]. 系统仿真学报, 2006 (12): 3524 - 3527.

[20] XIAO Y. Process Monitoring Based on Wavelet Transform Kernel Principal Component Analysis and Multiple Support Vector Machines [J]. Chinese Journal of Scientific Instrument, 2011, 31 (3): 558 - 564.

[16] 李金凤, 黄纬然, 赵雨童, 等. 基于 Kintex-7 FPGA 的 DDR3 SDRAM 高速访存控制器优化与实现 [J]. 现代电子技术, 2021, 44 (20): 112 - 116.

[17] 赵 航, 钱 博. 基于双 FIFO 乒乓操作的应用数据缓冲电路研究 [J]. 网络安全技术与应用, 2020 (12): 60 - 62.

[18] 李光春, 苏沛东, 杜世通, 等. 基于 FPGA 的视频采集及实时显示系统设计 [J]. 计算机测量与控制, 2015, 23 (11): 3810 - 3812.

[19] 黄纬然. 基于 FPGA 及 DDR3 的高速存储系统设计与实现 [D]. 沈阳: 沈阳化工大学, 2021.

[20] 王炳友, 党 博, 党瑞荣, 等. 基于 FPGA 的多通道出砂振动信号缓存系统研究 [J]. 石油化工应用, 2021, 40 (4): 98 - 104.

[21] 文生平, 张施华, 陈志鸿, 等. 基于模糊 PID 在线熔融指数仪测控系统设计 [J]. 中国测试, 2018, 44 (3): 97 - 101.

[22] 徐 托, 瞿少成, 王 安, 等. 基于模糊 PID 的室内空气质量测控系统 [J]. 电子测量技术, 2022, 45 (7): 62 - 67.