arXiv:2303.03324v3 [cs.LG] 9 Oct 2023

基于重构的状态空间模型进行时间序列异常检测

Fan Wang[1],Keli Wang[2,3],和 Boyu Yao[1]

[1]诺和诺德公司[2]中国铁道科学研究院研究生部[3]中国铁路检验认证中心有限公司

摘要

数字化的最新进展使各个领域都能够获取多变量时间序列数据，从而实现对运营情况的实时监测。在这些场景中识别异常数据模式并检测潜在故障十分重要，但也颇具挑战性。本文提出了一种针对时间序列数据的新型异常检测方法。该方法框架联合学习观测模型和动态模型，并根据正常样本估计模型不确定性。具体而言，采用基于长短期记忆（LSTM）的编码器-解码器来表示观测空间与状态空间之间的映射。通过利用向前和向后的时间信息，同时对状态的双向转移进行建模。对状态空间进行正则化，以约束正常样本的状态，并使用马氏距离评估异常程度。在合成数据集和真实世界数据集上的实证研究表明，该方法在异常检测任务中表现出优异的性能。

关键词：时间序列、神经网络、异常检测、状态空间模型

1 引言

时间序列数据的异常检测在金融、医疗保健和制造业等领域有着广泛的应用。异常通常是关键事件的重要信号，例如故障运行和健康状况恶化，因此，从数据角度捕捉此类信号至关重要。现实生活中的时间序列数据往往呈现复杂的模式，这给异常检测算法的方法带来了挑战。尤其是，高维性增加了提取有意义特征的难度，而这对算法性能至关重要；高度非线性动力学则进一步增加了识别系统状态的复杂性。

对一组随时间变化的测量值进行异常检测一直是一个活跃的研究领域[3]。通常，这一过程包括两个阶段：在训练阶段，利用历史数据建模，以学习时间序列的时间模式；在测试阶段，评估每个观测是否符合正常或异常模式。由于真实世界数据集通常缺少带标签的异常，而且异常可能表现出不可预测的数据行为，因此在这些情况下，训练集可能只包含正常运行的数据。异常检测方法可分为

基于聚类、基于距离、基于密度、基于孤立以及混合方法。传统机器学习方法，例如单类支持向量机 [15], 孤立森林 [14], 等，在从真实场景收集的数据呈现出日益增长的维度和复杂的动力学特性时，面临着诸多挑战。由于维度灾难以及难以理解时间关系，这些方法可能无法取得有竞争力的性能。与此同时，基于深度学习的方法因具备对复杂模式建模的能力而备受关注 [6] [4]。

2 相关工作

凭借更强的表达能力，基于深度学习的方法利用神经网络对数据背后的动力学进行建模，并且能够优于传统机器学习方法。例如，深度支持向量数据描述 [20] 是一种单类分类模型，其理念与单类支持向量机相似。对于实践中更常用的基于距离的模型，可以通过观测与估计值之间的差异来确定异常程度。例如，可以使用长短期记忆（LSTM）神经网络 [8] 根据过去的观测预测未来观测，预测误差则表明正常数据的时间关系是否遭到破坏。

近年来，基于重构的状态空间模型已成为热门研究主题。此类方法之一是联合学习观测值与隐状态之间的非线性映射，以及状态空间中的非线性转移；随后使用学得的模型进行推断。例如，[17] 中的作者针对时间序列数据提出了一种方法：使用编码器获取隐状态，以转移函数对状态演化进行建模，并通过解码器将隐状态转换到观测空间。该方法还会学习一个观测器函数，将状态、控制量和观测值传递到下一个状态。尽管该框架用于模型预测控制，但也可以扩展用于滤波。 [5] 中的作者提出了神经系统辨识与贝叶斯滤波（NSIBF），该方法采用类似的编码器-转移函数-解码器架构，用于异常检测。在测试阶段，该方法利用贝叶斯滤波方案递归地估计随时间变化的隐状态，其中测量函数和状态转移函数由学得的神经网络表示。与上述方法中用作编码器-解码器的全连接神经网络相比，循环神经网络（RNN），例如 LSTM，可以捕捉时间序列内部的时间关系。双向循环神经网络还可以联合学习时间序列两个方向的信息，从而更全面地表示底层动态。最后，对状态空间行为进行适当的正则化，并一致地定义异常程度，对于异常检测性能至关重要。

3 方法

3.1 问题陈述

令 $\{\boldsymbol{x^{(1)}}, \boldsymbol{x^{(2)}} \dots \boldsymbol{x^{(\tau)}}\}$ 表示信号变量的时间序列， $\{\boldsymbol{u^{(1)}}, \boldsymbol{u^{(2)}} \dots \boldsymbol{u^{(\tau)}}\}$ 表示相应的控制变量。由于异常通常表现为序列层面上的时间关系违背，我们考虑在时间 t 处长度为xl 的时间窗口：

$$\boldsymbol{x_t} = \{\boldsymbol{x^{(t-xl+1)}} \dots \boldsymbol{x^{(t-1)}}, \boldsymbol{x^{(t)}}\}, \tag{1}$$

类似地，窗口大小为ul 的相应控制序列为

$$\boldsymbol{u_t} = \{\boldsymbol{u^{(t-ul+1)}} \dots \boldsymbol{u^{(t-1)}}, \boldsymbol{u^{(t)}}\}, \tag{2}$$

因此，在训练和测试阶段，原始序列都会被转换为信号序列 $X = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots \boldsymbol{x_T}\}$和控制序列 $U = \{\boldsymbol{u_1}, \boldsymbol{u_2}, \dots \boldsymbol{u_T}\}$的窗口。每个样本 $\boldsymbol{x_t}(t = 1 \dots T)$都由一个二元变量 $y_t \in \{0, 1\}$标记，以表示其为正常（$y_t = 0$）还是异常（$y_t = 1$）。我们考虑这样一种情形：训练阶段没有可用的异常数据，模型基于上述形式的正常样本进行训练。在测试阶段，对于一个此前未见过的样本 $\boldsymbol{x_{t'}}$，通过应用训练好的模型，根据异常分数将其标记为 $\hat{}y_{t'} = 0$ 或 1。

对于动态系统，通常假设隐状态是对低维空间中观测的一种压缩表示。因此，对这类系统进行建模需要建立观测值与隐状态之间的映射，并描述状态空间内随时间发生的状态转移，即，

$$\begin{aligned} \boldsymbol{s_t} &= \Phi(\boldsymbol{s_{t-1}}, \boldsymbol{u_{t-1}}) + \boldsymbol{w_t} \\ \boldsymbol{x_t} &= \Theta(\boldsymbol{s_t}) + \boldsymbol{v_t}, \end{aligned} \tag{3}$$

其中， $\Phi(.,.)$、 $\Theta(.)$分别为状态转移函数和测量函数，而 $\boldsymbol{w_t}, \boldsymbol{v_t}$ 是相应的噪声，在时间 t处均值为零。 $\boldsymbol{s_t}$ 是时间 t处的隐状态，是对 $\boldsymbol{x_t}$的紧凑表示。当 $\Phi(.,.)$、 $\Theta(.)$为线性映射，且 $\boldsymbol{w_t}$、 $\boldsymbol{v_t}$ 服从高斯分布时，模型（3）即为广泛使用的卡尔曼滤波器 [11]。

3.2 双向动态状态空间模型

所提出模型的整体架构如图1所示，异常检测流程包含三个阶段[1]。在训练阶段，模型通过基于LSTM的编码器-解码器学习观测空间与状态空间之间的映射，并利用双向LSTM（$BiLSTM$） [7]联合学习前向和后向状态转移函数。具体而言，系统动力学建模如下:

$$\begin{aligned} \boldsymbol{s_t} &= E(\boldsymbol{x_t}) \\ \boldsymbol{x_t} &= D(\boldsymbol{s_t}) \\ \boldsymbol{s_{t+1}} &= F(\boldsymbol{s_t}, \boldsymbol{u_t}) = (f(\boldsymbol{s_t}) + \boldsymbol{u_t}+)/2 \\ \boldsymbol{s_{t-1}} &= B(\boldsymbol{s_t}, \boldsymbol{u_t}) = (f(\boldsymbol{s_t}) + \boldsymbol{u_t}-)/2 \\ \boldsymbol{u_t}+, \boldsymbol{u_t}- &= BiLSTM(\boldsymbol{u_t}), \end{aligned} \tag{4}$$

其中， $\boldsymbol{x_t}$、 $\boldsymbol{u_t}$ 和 $\boldsymbol{s_t}$ 分别为观测到的信号序列和控制序列，以及时间 t处的隐状态向量。模型训练的损失函数为

$$\begin{aligned} L = \sum_{t=2}^{T-1} &\alpha_1 ||\boldsymbol{x_{t-1}} - \hat{\boldsymbol{x}}_{\boldsymbol{t-1}}||^2 + \alpha_2 ||\boldsymbol{x_t} - \hat{\boldsymbol{x}}_{\boldsymbol{t}}||^2 + \alpha_3 ||\boldsymbol{x_{t+1}} - \hat{\boldsymbol{x}}_{\boldsymbol{t+1}}||^2 \\ &+ \beta_1 ||\boldsymbol{s_{t-1}} - \hat{\boldsymbol{s}}_{\boldsymbol{t-1}}||^2 + \beta_2 ||\boldsymbol{s_t}||^2 + \beta_3 ||\boldsymbol{s_{t+1}} - \hat{\boldsymbol{s}}_{\boldsymbol{t+1}}||^2, \end{aligned} \tag{5}$$

$$\begin{aligned} &\boldsymbol{s_t} = E(\boldsymbol{x_t}); \hat{\boldsymbol{s}}_{\boldsymbol{t-1}} = B(\boldsymbol{s_t}, \boldsymbol{u_t}); \hat{\boldsymbol{s}}_{\boldsymbol{t+1}} = F(\boldsymbol{s_t}, \boldsymbol{u_t}) \\ &\hat{\boldsymbol{x}}_{\boldsymbol{t-1}} = D(\hat{\boldsymbol{s}}_{\boldsymbol{t-1}}); \hat{\boldsymbol{x}}_{\boldsymbol{t}} = D(\boldsymbol{s_t}); \hat{\boldsymbol{x}}_{\boldsymbol{t+1}} = D(\hat{\boldsymbol{s}}_{\boldsymbol{t+1}}), \end{aligned} \tag{6}$$

[1]代码可在 https://github.com/DeepTSAD/BDM 获取

其中 $\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3 > 0$ 表示各项的权重。 $E(\boldsymbol{x_t})$ 是编码函数，由 LSTM 编码器实现。 $D(\boldsymbol{s_t})$ 是解码函数，由 LSTM 解码器实现 [16]。$F(\boldsymbol{s_t}, \boldsymbol{u_t})$ 和 $B(\boldsymbol{s_t}, \boldsymbol{u_t})$ 分别是前向和后向转移函数，它们通过以下方式联合学习。与本文其余部分的 [5], 类似，控制序列 $\boldsymbol{u_t}$ 表示长度为 ul 的滑动窗口中的信号变量和控制变量的并集。 $BiLSTM(\boldsymbol{u_t})$ 联合学习 $\boldsymbol{u_t}+$ 和 $\boldsymbol{u_t}-$，即前向和后向的隐向量； $f(\boldsymbol{s_t})$ 由全连接神经网络实现；最后， $F(\boldsymbol{s_t}, \boldsymbol{u_t})$ 和 $B(\boldsymbol{s_t}, \boldsymbol{u_t})$ 分别由 $(f(\boldsymbol{s_t}) + \boldsymbol{u_t}+)/2$ 和 $(f(\boldsymbol{s_t}) + \boldsymbol{u_t}-)/2$ 实现，其中 $f(.)$ 和 $BiLSTM(.)$ 使用相同的激活函数。

式（5）中的前三项是观测空间中的重构误差。注意，$\hat{}\boldsymbol{x_{t-1}}$ 和 $\hat{}\boldsymbol{x_{t+1}}$ 既是编码器-解码器的输出，也是转移函数的输出。这样做是为了避免解码器放大 $F(.,.)$ 和 $B(.,.)$ 的误差。由于同时考虑了两个方向的动力学，编码器-解码器对能够学习统一表示，因此具有更强的鲁棒性。式（5）中的第四项和第六项是状态空间中的预测误差，第五项则旨在缩小状态估计。该正则化项促使正常样本的状态接近原点，从而避免隐状态分布出现意外情况，稳定训练过程；它还有助于异常检测过程，因为状态远离原点的异常样本会导致较大的重构误差。

在验证阶段，对于验证数据 D_{val} 中的每一对信号（$\boldsymbol{x_{t-1}}$， $\boldsymbol{x_t}$）和控制量 $\boldsymbol{u_{t-1}}$，我们按如下方式计算重构误差：

$$\begin{aligned} \boldsymbol{s_{t-1}} &= E(\boldsymbol{x_{t-1}}) \\ \boldsymbol{e_t} &= \boldsymbol{x_t} - D(F(\boldsymbol{s_{t-1}}, \boldsymbol{u_{t-1}})). \end{aligned} \tag{7}$$

此类重构误差的协方差通过经验计算得出，记为 $\boldsymbol{\Sigma}$。

在测试阶段，时间为 t' 时，给定信号对（$\boldsymbol{x_{t'-1}}$， $\boldsymbol{x_{t'}}$）和控制量 $\boldsymbol{u_{t'-1}}$，异常分数定义为马氏距离 [19], 如下所示：

$$\begin{aligned} \boldsymbol{s_{t'-1}} &= E(\boldsymbol{x_{t'-1}}) \\ \boldsymbol{\mu_{t'}} &= D(F(\boldsymbol{s_{t'-1}}, \boldsymbol{u_{t'-1}})) \\ \text{anomaly score} &= \sqrt{(\boldsymbol{x_{t'}} - \boldsymbol{\mu_{t'}})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x_{t'}} - \boldsymbol{\mu_{t'}})}, \end{aligned} \tag{8}$$

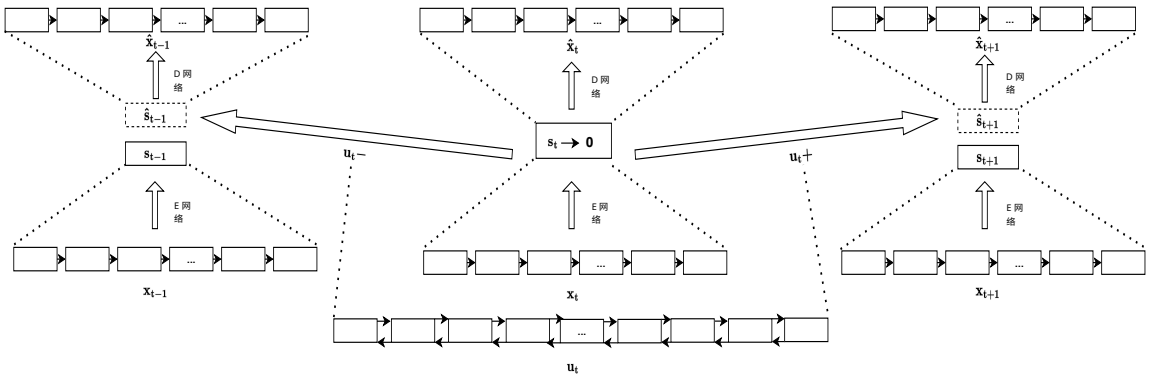

图1：所提出网络的架构

较高的异常分数表示可能存在异常。请注意，与普通重构误差相比，马氏距离会考虑变量的尺度；也就是说，误差大小是相对于其基线协方差进行评估的，而不是依据误差本身。

4 结果

在本节中，我们将所提出的方法与若干先进的异常检测方法在合成数据集和真实世界数据集上进行比较。全文中，所提出的方法均采用相同的网络结构： E(。)包含一个维度为 4 的 LSTM 层，其隐状态维度也为 4； D(。)包含一个维度为 4 的 LSTM 层，后接一个维度为 4 的全连接层； $BiLSTM$(。)包含两个维度为 4 的双向LSTM层； f(。)包含两个维度为 4 的全连接层。全文中，我们对连续变量进行最小-最大归一化，对离散变量进行独热编码。我们使用 3/4 的训练数据来训练所提出的模型，并将其余 1/4 的训练数据用于验证。对于所有实验，式 (5) 中的 α_1, α_2, α_3, β_1, β_2, β_3 均固定为 1、1、1、0.1、0.1、0.1，因为我们发现这样通常能取得理想的结果。

4.1 基线方法

我们将以下异常检测方法作为基线:

- 孤立森林（IF） [14]是一种基于隔离的方法，通过学习树形架构来实现
- 自动编码器（AE） [13] 由全连接层构成的编码器和解码器组成，并使用重构误差作为异常分数
- 长短期记忆自动编码器（LSTM AE） [16]由通过 LSTM 网络实现的编码器和解码器组成，并使用重构误差作为异常分数
- 深度自编码高斯混合模型（DAGMM） [21] 联合学习深度自动编码器和高斯混合模型，以计算观测值的似然
- 神经系统辨识与贝叶斯滤波（NSIBF） [5]联合学习编码器、解码器和状态转移函数，并使用贝叶斯滤波递归更新状态估计
- 无监督异常检测（USAD） [2] 通过对抗训练自动编码器模型，放大异常样本的重构误差

4.2 合成数据示例

在本节中，我们使用一个模拟时间序列场景，将所提出的方法与上述先进的异常检测方法进行比较。在训练阶段，我们考虑由以下简单动态模型生成的正常样本。对于 $t = 1,\ 2,\ ...T$:

$$
\begin{aligned}
u^{(t)} &= \lceil \frac{t - 1000 \times \lfloor \frac{t-1}{1000} \rfloor}{100} \rceil \\
s^{(t)} &= sin(t-1) + sin(u^{(t)}) + w^{(t)} \\
x^{(t)} &= s^{(t)} + v^{(t)},
\end{aligned}
\tag{9}
$$

其中 $T = 10000$, $w^{(t)} \sim N(0,\ 0.5_2)$，且 $v^{(t)} \sim N(0,\ 1)$。在测试阶段，使用相同的动态模型生成另外 $T' = 10000$ 个样本，但在每 1000 个样本中的最后 100 个样本里注入异常，注入参数为 $w^{(t)} \sim N(0,\ 1)$ 和 $v^{(t)} \sim N(0, 2^2)$。构建信号和控制序列时，xl 和 ul 分别取 8 和 16。

下面给出了候选方法的受试者工作特征（ROC）曲线及相应的曲线下面积（AUC）值 [9]，用于评估其排序性能。如图 2 所示，我们的方法取得了最高的 AUC，达到 0.95，其次是孤立森林、自编码器、USAD 和长短期记忆自编码器；在将假阳性率控制在较低水平时，我们的方法具有更高的真阳性率。值得注意的是，在这个动力学简单的合成示例中，与其他基于深度学习的方法相比，传统机器学习方法孤立森林取得了第二好的排序性能，AUC 为 0.935。NSIBF 和深度自编码高斯混合模型的排序性能相近，且与其他方法相比竞争力较弱。

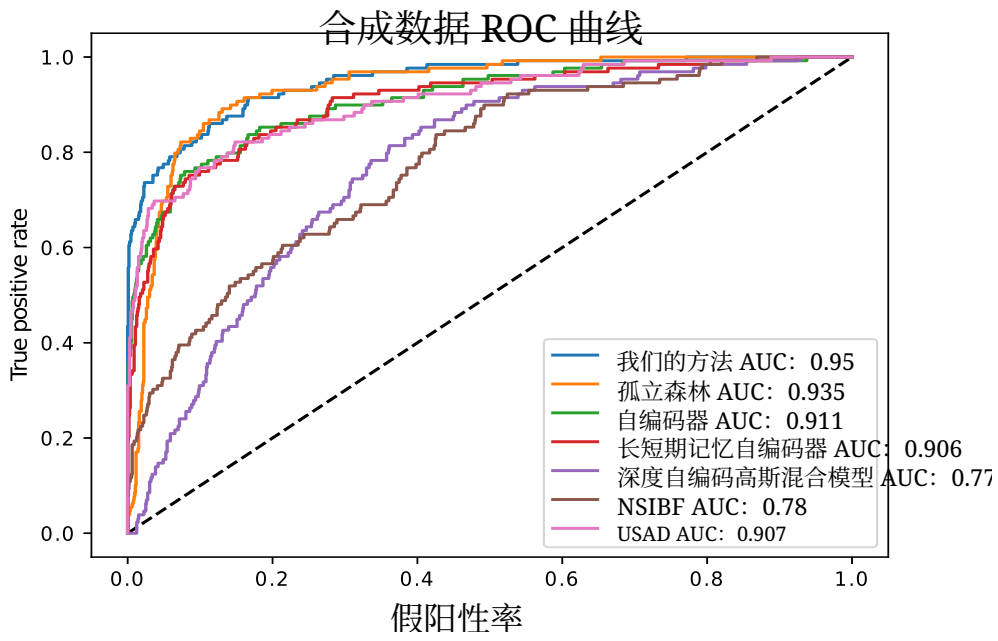

图 2：合成数据集的 ROC 曲线

4.3 真实世界示例

本节中，我们使用真实世界数据集评估所提出的方法。实践中，此类对潜在异常进行恰当标注的数据集十分稀少，因此我们使用了两个来自水处理厂的数据集：$SWaT$ [18] 和 $WADI$ [1]，，其中的异常由领域专家根据模拟攻击场景进行标注。这两个数据集最初由功能齐全的测试平台生成，旨在模拟真实工业设施的动力学。数据集包含传感器测量值（信号变量）以及执行器状态（控制变量），

时间序列。

- $SWaT$ 测试平台是一座按比例缩小的工业水净化厂。该测试平台采集的数据覆盖11天，每秒采集一次；最后四天模拟了攻击场景，并将其反映在数据集中。按照 [5], ，我们对数据进行降采样，每五秒保留一个样本。按照[12], ，根据训练数据与测试数据的概率分布相似性，从数据集中移除以下变量：AIT201、AIT202、AIT203、P201、AIT401、AIT402、AIT501、AIT502、AIT503、AIT504、FIT503、FIT504、PIT501、PIT502和PIT503。处理后，数据中包含11个信号变量和25个控制变量。

- $WADI$ testbed 是一个按比例缩小的工业供水系统。训练数据包含正常运行条件下连续 14 天、每秒采集的数据。在运行的最后两天，模拟了各种攻击场景。随后 [5], 我们对数据进行降采样，使每五秒保留一个样本，并移除训练数据中取值恒定的执行器。由于运行模式变化导致分布偏移，因此忽略最后一天的数据。随后， [12], 根据训练数据与测试数据之间概率分布的相似性，从数据集中移除以下变量：1_AIT_001_PV、 1_AIT_003_PV、 1_AIT_004_PV、 1_AIT_005_PV、 2_LT_001_PV、2_PIT_001_PV、2A_AIT_001_PV、2A_AIT_003_PV、2A_AIT_004_PV、2B_AIT_001_PV、2B_AIT_002_PV、2B_AIT_003_PV、2B_AIT_004_PV，以及 3_AIT_005_PV。处理后，共有 53 个信号变量和 26 个控制变量。

在我们的实验中， xl 在 SWaT 数据集上设为 16，在 WADI 数据集上设为 8； ul 在 SWaT 数据集上设为 32，在 WADI 数据集上设为 16。获得各方法的异常分数后，我们枚举所有可能的异常阈值，并以最佳 F1 分数作为评估指标。我们还报告相应的精确率和召回率。结果汇总于表 1。可以看到，我们的方法在两个数据集上的 F1 分数均为最佳；与排名第二的方法相比，在 SWaT 和 WADI 数据集上分别提升了 2.4% 和 18.2%。传统机器学习方法 IF 在两个数据集上的相对表现都较差，表明它难以在高维场景中捕捉复杂的时间模式。AE 和长短期记忆自编码器的表现相近，这可能是因为它们的重构误差忽略了不同变量的尺度。深度自编码高斯混合模型在 SWaT 数据集上的表现具有竞争力，但在 WADI 数据集上的表现最差，这可能是因为它难以在高维场景中推断似然。NSIBF 在 WADI 数据集上的 F1 分数与 USAD 相近，在 SWaT 数据集上的表现则更好；由于采用了滤波方案，NSIBF 是唯一一种在测试阶段不支持批处理的方法，因此分析历史数据时可能耗时更长。

4.4 进一步研究

我们分析了所提出的异常度量定义的性质。NSIBF 和我们的方法都使用马氏距离来衡量异常程度。在测试阶段，NSIBF 假设隐状态服从高斯分布，并递归地应用无迹卡尔曼滤波器 [10] ，根据由神经网络实现的测量函数和转移函数更新状态分布。由于神经网络具有高度非线性，输入的微小变化就可能显著改变输出。因此，当将它们应用于来自

表 1：SWaT 和 WADI 数据集上的异常检测性能比较 ts

方法	SWaT 精确率	SWaT 召回率	SWaT 最佳F1值	WADI 精确率	WADI 召回率	WADI 最佳F1值
IF	1.0	0.567	0.723	0.180	0.455	0.258
AE	0.995	0.613	0.759	0.220	0.537	0.312
长短期记忆自编码器	0.997	0.598	0.748	0.492	0.220	0.304
深度自编码高斯混合模型	0.957	0.643	0.769	0.904	0.131	0.228
NSIBF	0.892	0.712	0.792	0.234	0.496	0.318
USAD	0.995	0.629	0.771	0.243	0.462	0.319
我们的方法	0.991	0.685	0.811	0.276	0.593	0.377

状态分布的样本（例如 [5] 中的西格玛点）时，那些远离实际状态的样本可能导致马氏距离出现意外行为。下面我们重新考察第 4.2 节中的合成数据示例，图 3 对比了 NSIBF 和所提方法生成的异常分数。可以看到，两种方法总体上都会为异常时段赋予较高的异常分数，但 NSIBF 在正常时段的表现有所波动，而我们的方法生成的异常分数则更加稳定。因此，它能更好地区分正常样本和异常样本。

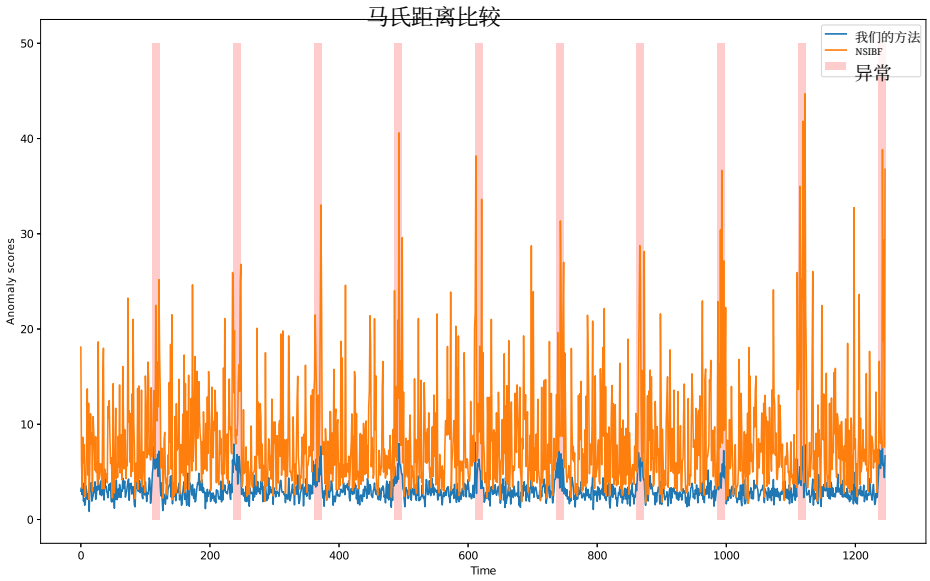

图 3：第 4.2 节合成示例中 NSIBF 与我们的方法的马氏距离比较。红色阴影标记异常时段

与 [5], 类似，所提出的模型可以结合贝叶斯滤波方案进行状态识别。此外，由于前向和后向转移函数是联合学习的，

因此可以基于两个方向的动力学进行推断。

我们采用第 4.2 节中的合成数据示例，但在训练和测试阶段，噪声水平均设为较小值，分别为 $w^{(t)} \sim N(0,\ 0.1_2)$ 和 $v^{(t)} \sim N(0,\ 0.1_2)$。真实值定义为 $w^{(t)} = v^{(t)} = 0$ 条件下无噪声过程的观测值，目标是根据含噪数据进行重构，以恢复真实值。我们将所提出的模型与 [5]，提出的无迹卡尔曼滤波器方案相结合，不同之处在于，我们在两个方向上都有转移函数及相应的误差协方差估计。我们递归地进行滤波（前向传播），根据状态估计 $\boldsymbol{s_t^f}$，通过$\hat{}\boldsymbol{x_t}$重构信号序列 $\boldsymbol{x_t}$。即，

$$\begin{aligned} &\boldsymbol{s_{t-1}^f} \sim N(\hat{\boldsymbol{s}}_{\boldsymbol{t-1}}^{\boldsymbol{f}}, \hat{\boldsymbol{P}}_{\boldsymbol{t-1}}^{\boldsymbol{f}}) \xrightarrow[D(.)]{F(.,.)} \boldsymbol{s_t^f} \sim N(\hat{\boldsymbol{s}}_{\boldsymbol{t}}^{\boldsymbol{f}}, \hat{\boldsymbol{P}}_{\boldsymbol{t}}^{\boldsymbol{f}}) \\ &\hat{\boldsymbol{x}}_{\boldsymbol{t}} = D(\hat{\boldsymbol{s}}_{\boldsymbol{t}}^{\boldsymbol{f}}), \end{aligned} \tag{10}$$

其中，$\hat{}\boldsymbol{s_t^f}$ 和 $\hat{}\boldsymbol{P_t^f}$ 分别是时间 t 前向传播中更新后的状态分布的均值和协方差估计。相比之下，我们根据反向传播的状态估计 $\boldsymbol{s_t^b}$，通过$\hat{}\boldsymbol{x_t'}$重构信号序列 $\boldsymbol{x_t}$；具体而言，对前向传播得到的状态估计应用反向转移函数。即，

$$\begin{aligned} &\boldsymbol{s_{t+1}^f} \sim N(\hat{\boldsymbol{s}}_{\boldsymbol{t+1}}^{\boldsymbol{f}}, \hat{\boldsymbol{P}}_{\boldsymbol{t+1}}^{\boldsymbol{f}}) \xrightarrow[D(.)]{B(.,.)} \boldsymbol{s_t^b} \sim N(\hat{\boldsymbol{s}}_{\boldsymbol{t}}^{\boldsymbol{b}}, \hat{\boldsymbol{P}}_{\boldsymbol{t}}^{\boldsymbol{b}}) \\ &\hat{\boldsymbol{x}}_{\boldsymbol{t}}' = D(\hat{\boldsymbol{s}}_{\boldsymbol{t}}^{\boldsymbol{b}}), \end{aligned} \tag{11}$$

其中，$\hat{}\boldsymbol{s_t^b}$ 和$\hat{}\boldsymbol{P_t^b}$分别是时间 t反向传播得到的更新后状态分布的均值和协方差估计。图4比较了前向传播和反向传播的重构结果。由于获得了未来观测的额外信息，反向传播通常具有更小的重构误差。在底层动态的转变阶段尤其如此，如此处时间 $t = 180$ 附近所示（所有样本中，前向传播和反向传播的均方重构误差分别为0.0020和0.00056）。

5 结论

本文提出了一种新颖的、基于深度学习的状态空间模型，用于时间序列数据的异常检测。我们使用 LSTM 编码器 - 解码器，在状态空间中学习时间序列的隐藏表示及其到观测空间的映射。该模型利用时间序列上的双向LSTM联合学习正向和反向的转移函数。我们对状态空间应用正则化，使学习过程更加稳定且包含更多信息。我们定义了异常分数，以自适应地考虑变量的尺度。合成数据和真实世界数据实验均表明，与多种最先进方法相比，该模型提升了异常检测指标。该模型还易于实现，并有望与贝叶斯滤波方案相结合。一个值得进一步研究的问题是：在联合建模观测空间和状态空间时，如何系统地平衡二者，以实现整体性能最优。

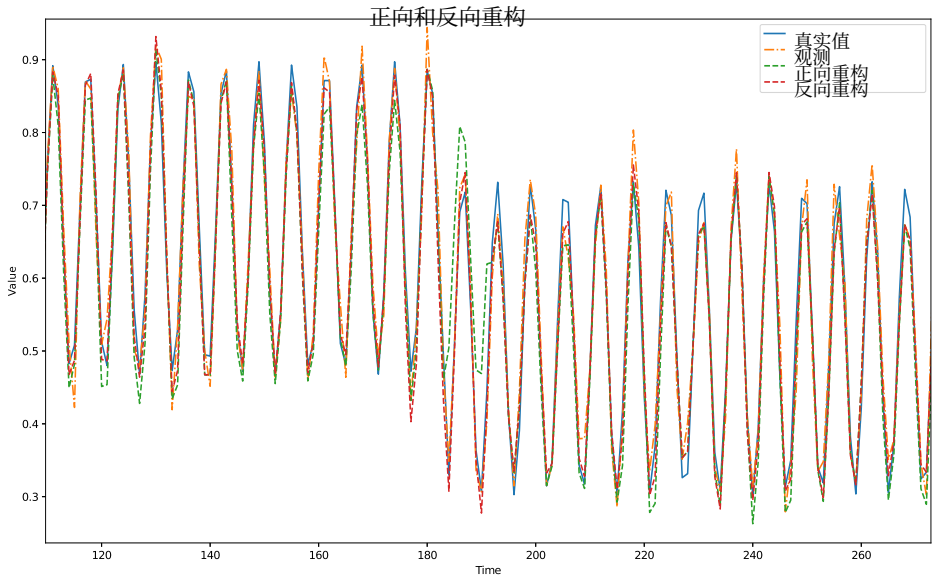

图4：低噪声仿真研究中正向和反向重构的比较（选定时段）。真实值以蓝色实线表示；观测值以橙色虚线表示；正向和反向重构分别以绿色和红色虚线表示

参考文献

[1] Ahmed, C.M., Palleti, V.R., Mathur, A.P.：WADI：用于研究安全信息物理系统设计的供水分配测试平台。载于：第三届智能水网络信息物理系统国际研讨会论文集。第25–28页。美国计算机协会（2017）。https://doi.org/10.1145/3055366.3055375[2] Audibert, J., Michiardi, P., Guyard, F., 等：USAD：多变量时间序列的无监督异常检测。载于：第26届ACM SIGKDD知识发现与数据挖掘国际会议论文集。第3395–3404页。美国计算机协会（2020）。https://doi.org/10.1145/3394486.3403392[3] Chandola, V., Banerjee, A., Kumar, V.：异常检测：综述。《ACM计算调查》41(3), 15:1–15:58（2009）[4] Darban, Z.Z., Webb, G.I., Pan, S., 等：时间序列异常检测的深度学习：综述。ArXiv 预印本（2022）。https://doi.org/10.48550/arxiv.2211.05244[5] Feng, C., Tian, P.：通过神经系统辨识与贝叶斯滤波对信息物理系统进行时间序列异常检测。载于：第27届ACM SIGKDD会议论文集

知识发现与数据挖掘。第2858–2867页。美国计算机协会（2021）。https://doi.org/10.1145/3447548.3467137[6] Garg, A., Zhang, W., Samaran, J., 等：多变量时间序列中异常检测与诊断的评估。《IEEE神经网络与学习系统汇刊》**33**，2508–2517（2021）[7] Graves, A., Schmidhuber, J.：基于双向LSTM及其他神经网络架构的逐帧音素分类。《神经网络》**18**(5)，602–610（2005）。https://doi.org/10.1109/IJCNN.2005.1556215[8] Hochreiter, S., Schmidhuber, J.：长短期记忆。《神经计算》**9**(8)，1735–1780（1997）[9]Huang, J., Ling, C.X.：使用AUC和准确率评估学习算法。《IEEE知识与数据工程汇刊》**17**(3)，299–310（2005）。https://doi.org/10.1109/TKDE.2005.50[10] Julier, S.J., Uhlmann, J.K.：无迹滤波与非线性估计。《IEEE学报》**92**(3)，401–422（2004）。https://doi.org/10.1109/JPROC.2003.823141[11] Kalman, R.E.：线性滤波与预测问题的新方法。《基础工程杂志》**82**(1)，35–45（1960）[12] Kravchik, M., Shabtai, A.：使用轻量级神经网络和主成分分析高效检测工业控制系统中的网络攻击。《IEEE可信与安全计算汇刊》**19**(04)，2179–2197（2022）。https://doi.org/10.1109/TDSC.2021.3050101[13] LeCun, Y., Bengio, Y., Hinton, G.E.：深度学习。《自然》**521**(7553)，436–444（2015）。https://doi.org/10.1038/nature14539[14] Liu, F.T., Ting, K.M., Zhou, Z.H.：孤立森林。载于：2008年第八届IEEE国际数据挖掘会议。第413–422页。IEEE（2008）[15] Ma, J., Perkins, S.：使用单类支持向量机进行时间序列新颖性检测。载于：国际联合神经网络会议论文集。第3卷，第1741–1745页（2003）。https://doi.org/10.1109/IJCNN.2003.1223670[16] Malhotra, P., Ramakrishnan, A., Anand, G., 等：用于多传感器异常检测的基于LSTM的编码器-解码器。ArXiv预印本（2016）。https://doi.org/10.48550/arxiv.1607.00148[17] Masti, D., Bemporad, A.：使用自编码器学习非线性状态空间模型。《自动化》**129**，109666（2021）。https://doi.org/10.1016/j.automatica.2021.109666[18] Mathur, A.P., Tippenhauer, N.O.：SWaT：用于工业控制系统安全研究与训练的水处理测试平台。2016年智能水网络信息物理系统国际研讨会，第31–36页（2016）。https://doi.org/10.1109/CySWater.2016.7469060[19] McLachlan, G.J.：马氏距离。《共振》**4**，20–26（1999）

[20] Ruff, L., Vandermeulen, R., Goernitz, N., 等：深度单类分类。载于：第35届国际机器学习大会论文集。《机器学习研究论文集》，第80卷，第4393–4402页（2018）[21] Zong, B., Song, Q., Min, M.R., 等：用于无监督异常检测的深度自编码高斯混合模型。载于：国际学习表征会议（2018）