# 上下文信息融合与噪声自适应的异常检测方法

衡红军, 周文华

引用本文

衡红军, 周文华. 上下文信息融合与噪声自适应的异常检测方法[J]. 计算机科学, 2023, 50(7): 237-245.

HENG Hongjun, ZHOU Wenhua. Anomaly Detection Method Based on Context Information Fusion and Noise Adaptation [J]. Computer Science, 2023, 50(7): 237-245.

---

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220700078

# 上下文信息融合与噪声自适应的异常检测方法

**衡红军　周文华**
中国民航大学计算机科学与技术学院　天津 300300
(henghjcauc@163.com)

**摘　要**　信息物理系统(CPSs)中传感器和执行器等现场设备收集的数据中隐含复杂的上下文信息和未知分布噪声。为了提取并融合数据中的上下文信息以及减轻噪声带来的干扰,提出了上下文信息融合与噪声自适应的异常检测方法。该方法中设计了一种由自适应降噪编码器、上下文信息编码器和解码器构成的编解码网络建模CPSs状态空间模型。自适应降噪编码器在训练过程中通过拟合数据中噪声的分布模式生成自适应噪声,并利用该噪声对训练数据中的传感器数据加噪,以提升编解码网络的鲁棒性,减轻噪声带来的干扰,同时可迫使降噪自编码器学习到泛化性更强的系统的隐藏状态;上下文信息编码器利用LSTM和CNN提取数据窗口内的时序和空间上下文信息,并使用自注意力机制融合这两类上下文信息和系统隐藏状态,融合结果用于推断当前时刻系统隐藏状态,以提升此隐藏状态中的信息量;解码器利用以上系统隐藏状态可以更准确地解码出相应的传感器数据。编解码网络训练完成后,得到系统隐藏状态和传感器解码值,基于无迹卡尔曼滤波算法计算异常评分。在SWaT和PUMP两个实际CPSs数据集上的实验结果表明,所提方法的F1值均优于其他对比方法,验证了其有效性。

**关键词**:异常检测;自适应噪声;上下文信息;状态空间模型;信息物理系统
**中图法分类号**　TP183

## Anomaly Detection Method Based on Context Information Fusion and Noise Adaptation

HENG Hongjun and ZHOU Wenhua
College of Computer Science and Technology,Civil Aviation University of China,Tianjin 300300,China

**Abstract**　The data collected by field devices such as sensors and actuators in cyber-physical systems(CPSs) contains complex context information. To extract and fuse context information in data and reduce the interference caused by noise,an anomaly detection method based on context information fusion and noise adaptation is proposed. In this method,an encoder-decoder network composed of adaptive denoising encoder,context information encoder and decoder is designed to model the state-space model of CPSs. The adaptive denoising encoder generates adaptive noise by fitting the distribution of noise in the data during the training process,and adds the adaptive noise to the sensor data of the training data,so as to improve the robustness of the encoder-decoder network,reduce the interference caused by noise,and force the adaptive denoising decoder to learn the system hidden state with stronger generalization. Context information encoder uses long-short term memory(LSTM) and convolutional neural networks (CNN) to extract temporal context information and spatial context information in the data window,and uses self-attention mechanism to fuse these two types of context information with system hidden state,so as to obtain the fusion result,which is used to infer the system hidden state at the current moment,so as to increase the amount of information of system hidden state at the current moment. The decoder can decode the corresponding sensor data more accurately by using the above system hidden states. After the encoder-decoder network training is completed,the system hidden state and the decoded sensor data are obtained,and the anomaly score is calculated based on the unscented Kalman filter algorithm. Experimental results on two actual CPSs datasets of SWaT and PUMP show that the F1 value of the proposed method is better than other comparison methods,which verifies its effectiveness.

**Keywords**　Anomaly detection,Adaptive noise,Context information,State space model,Cyber-physical systems

---

到稿日期:2022-07-08　返修日期:2022-12-08
基金项目:国家自然科学基金(U1333109)
This work was supported by the National Natural Science Foundation of China(U1333109).
通信作者:周文华(zwhhouwhua@163.com)

## 1 引言

信息物理系统(Cyber-Physical Systems,CPSs)作为一种集计算机技术、控制技术和通信技术于一体的新型网络系统,已被广泛部署于化工生产、能源和航空航天等物理自动化领域,对社会经济产生了重大影响[1-2]。传感器和执行器等现场设备能实时监控 CPSs 中的物理过程,对这些设备产生的多维时序数据实施异常检测可以在早期发现异常的系统行为,便于相关人员及时采取预防措施,从而避免系统故障造成的社会经济损失。

将深度学习应用于 CPSs 场景下的异常检测已成为一种明显的趋势,但目前基于深度学习的异常检测方法未充分考虑数据中隐含的上下文信息和未知分布噪声对检测效果的影响。

(1)异常依赖于上下文环境,需提供局部上下文信息[3-4]。CPSs 产生的多维时序数据同时含有时序上下文信息和空间上下文信息。如异常邻域内的时间点可能会包含相似的异常模式,本文中称其为时序上下文信息;某些异常可能需要多个传感器和执行器数据共同标识,本文中称其为空间上下文信息。有效地提取并融合这两类上下文信息,对异常判定而言至关重要。

(2)CPSs 数据中未知分布噪声引起的异常误判。实际的 CPSs 场景中存在未知分布噪声,传感器收集到的数据可能与实际测量存在差异。如传感器制造过程中由硬件问题而引起的传感器噪声,传感器使用过程中由外部因素干扰或自身部件老化消耗而引起的过程噪声,且这些噪声可能不规则地分布在数据空间中[5-6]。

为解决上述问题,本文提出了上下文信息融合与噪声自适应的异常检测方法,设计了一种由自适应降噪编码器、上下文信息编码器和解码器构成的编解码网络建模 CPSs 状态空间模型,主要工作如下。

(1)设计了自适应降噪编码器。此编码器在训练过程中自适应地拟合 CPSs 数据中噪声的分布模式,生成了自适应噪声,并利用该噪声对训练数据中的传感器数据加噪,以提升编解码网络的鲁棒性,从而减轻噪声带来的干扰;同时可迫使降噪自编码器学习到泛化性更强的系统隐藏状态。

(2)设计了上下文信息编码器。此编码器利用 LSTM(Long-Short Term Memory)和 CNN(Convolutional Neural Networks)分别提取数据窗口内的时序上下文信息和空间上下文信息;经自注意力机制融合这两类上下文信息和自适应降噪编码器输出的系统隐藏状态,得到融合信息;由多层隐藏层根据融合信息推断当前时刻系统的隐藏状态,以提升此隐藏状态中的信息量,从而更准确地检测异常。

(3)设计了解码器。此解码器利用上述两类编码器输出的系统隐藏状态,可以更准确地解码出相应的传感器数据。

编解码网络训练完成后,得到系统隐藏状态和传感器解码值,应用无迹卡尔曼滤波算法递归预测系统状态随时间的联合概率分布,由待测样本中传感器数据与其预测分布的马氏距离作为异常评分。SWaT 和 PUMP 两个实际 CPSs 数据集上的实验结果表明,本文方法的 F1 值较基准算法有所提升。

## 2 相关工作

时间序列异常检测指在时序数据的各时间步中识别系统的异常状态[7],在各种应用领域中,不一致的点被称为异常、入侵、故障或污染物。近年来,多维时序异常检测研究广泛采用深度学习方法,根据其使用的训练数据,可被分为有监督方法[8-9]和无监督方法[10-18]。由于缺乏标记的异常数据实例,多维时序异常检测方法多采用无监督方法。

Malhotra 等[10]提出了基于 LSTM 的自编码器(Autoencoder,AE)异常检测方法,该方法建模正样本的分布,利用 LSTM 提取数据的时序上下文信息,由待测样本的重构误差判定异常。由于训练样本与其隐藏表示之间为确定映射,因此使得基于 AE 的异常检测方法对噪声敏感。Li 等[12]提出了基于生成对抗网络(Generative Adversarial Networks,GAN)的异常检测方法,该方法通过生成器和鉴别器之间的对抗训练建模正样本的分布,由待测样本的生成器误差和鉴别器误差判定异常。Hundman 等[15]提出了一种基于预测的异常检测方法,该方法利用 LSTM 预测正样本下一时刻的值,由预测误差判定异常,同时提出了一种非参数阈值选择法,在异常的误检和漏检之间找到一个合适的平衡点。使用误差作为异常评分是一种基于偏差的方法,由于正样本和异常样本可能共享相同的平均值,仅使用误差不能很好地区分正样本和异常样本。

基于密度的异常评分方法以概率的方式计算样本点的可能性,考虑了样本空间中隐含的随机性,在一定程度上提升了异常检测的效果。Su 等[16]提出了一种基于 VAE 的异常检测方法,该方法采用门控循环单元提取数据的时序上下文信息,同时采用标准化流技术来拟合真实的近似后验分布,由待测样本的重构概率判定异常。Deng 等[17]提出了一种基于图注意力的异常检测方法,该方法利用有向图建模正样本时序特征之间的关联模式,由图偏离分数判定异常,提供了更好的异常解释。Feng 等[18]提出了一种基于神经网络的状态估计异常检测方法,该方法结合神经网络与传统状态估计算法,在由神经网络表示的状态空间模型上,基于贝叶斯滤波方法计算异常评分,同时与基于重构误差和预测误差的异常评分进行对比,实验结果验证了基于密度的异常评分方法具有更高的检测指标。Schneider 等[19]提出了一种基于对比表示学习和转换学习的异常检测方法,该方法利用对比预测编码损失提升网络嵌入表示的质量,同时设计动态确定对比损失迫使网络提取嵌入表示中不同语义的上下文信息,由动态确定对比损失判定异常。

基于深度学习的异常检测方法建模正常情况下时序数据的分布模式,由待测样本的重构误差、预测误差或基于密度的方法计算异常评分。这些方法在诸多时间序列异常检测任务上表现出了优异的性能。对于 CPSs 异常检测任务,仍可以进一步提取并融合数据中隐含的上下文信息,以更准确地检测异常,同时提升网络的鲁棒性,以减轻数据中噪声带来的干扰。

## 3 CPSs 状态空间模型

为检测异常的系统行为,避免系统故障造成社会经济

损失，可以通过传感器和执行器实时监控 CPSs 中的物理过程。如图 1 所示，传感器、执行器和控制器 3 种现场设备直接控制 CPSs 中的物理过程[20]。

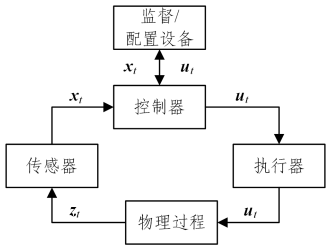



图 1 CPSs 中的物理过程

Fig. 1 Physical processes in CPSs

传感器负责将系统物理状态转换为电子测量值，并向控制器发送报告；控制器根据从传感器接收到的电子测量值向执行器发送控制信号，如开关阀门的控制信号；执行器根据接收到的控制信号转换系统物理状态，例如，系统中阀门由开到关；监督设备或其他配置设备通过与控制器通信来监控系统或更改控制器的配置。

设 $\boldsymbol{x}_t$ 为 $t$ 时刻传感器电子测量值，$\boldsymbol{u}_t$ 为 $t$ 时刻执行器的状态，则上述 CPSs 系统的物理过程可由如下状态空间模型[18]表示：

$$\boldsymbol{z}_t=\boldsymbol{F}\boldsymbol{z}_{t-1}+\boldsymbol{B}\boldsymbol{u}_{t-1}+\boldsymbol{\theta}_t \tag{1}$$

$$\boldsymbol{x}_t=\boldsymbol{H}\boldsymbol{z}_t+\boldsymbol{\varepsilon}_t \tag{2}$$

其中，$\boldsymbol{z}_t$ 为系统隐藏状态，例如真实的液体温度，由于传感器噪声的干扰，真实的液体温度与传感器电子测量值之间可能存在一定差异；$\boldsymbol{F}$ 为状态转换矩阵，表示系统隐藏状态随时间演化的系统物理过程；$\boldsymbol{B}$ 为控制矩阵，表示控制信号引起系统物理状态转换的系统物理过程，$\boldsymbol{\theta}_t$ 为 $\boldsymbol{F}$ 和 $\boldsymbol{B}$ 过程中的噪声；$\boldsymbol{H}$ 为测量矩阵，表示系统隐藏状态转换为传感器电子测量值的系统物理过程，$\boldsymbol{\varepsilon}_t$ 为 $\boldsymbol{H}$ 过程中的噪声。

在 CPSs 状态空间模型的基础上，应用状态估计算法可以捕获复杂的系统动态，检测异常的系统行为。状态估计算法递归预测系统状态随时间的联合概率分布，当测量值与预测值的误差或测量值与其预测分布的距离大于阈值时判定为异常。为此，本文设计了一种编解码网络建模 CPSs 状态空间模型，并基于无迹卡尔曼滤波算法[21]计算异常评分，检测异常的系统行为。

## 4 本文方法

考虑到实际 CPSs 数据中隐含的复杂上下文信息与未知分布噪声，提出了一种上下文信息融合与噪声自适应的异常检测方法，该方法提取并融合数据中隐含的上下文信息，以更准确地检测异常。同时在训练过程中生成自适应噪声，并利用该噪声对训练数据中的传感器数据加噪，以提升网络的鲁棒性，减轻噪声带来的干扰。图 2 描述了所提方法的异常检测流程，主要包含 3 个阶段：数据预处理阶段、编解码网络建模阶段、异常检测阶段。

(1)数据预处理阶段：为便于网络训练，依次对 CPSs 数据进行降采样、传感器数据归一化、执行器数据独热编码、滑动窗口划分、数据集划分处理。

(2)编解码网络建模阶段：设计了一种基于编解码结构的神经网络建模 CPSs 状态空间模型。依据目标函数，使用随机梯度下降算法更新编解码网络中的参数。

(3)异常检测阶段：将待测样本输入训练完成的编解码网络中，基于无迹卡尔曼滤波算法计算异常评分来检测异常。

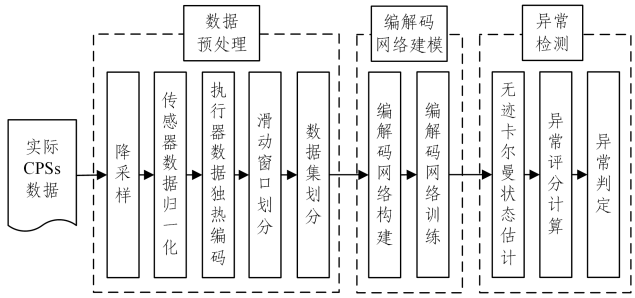



图 2 所提方法的异常检测流程

Fig. 2 Anomaly detection process of the proposed method

### 4.1 数据预处理

时序数据指等时间间隔收集的连续测量数据。本文的研究目标是 CPSs 产生的多维时序数据，由 $N$ 个传感器和执行器在固定时间间隔所收集的长度为 $L$ 的测量值构成，表达式如下：

$$\boldsymbol{X}=\{\{\boldsymbol{x},\boldsymbol{u}\}_1,\{\boldsymbol{x},\boldsymbol{u}\}_2,\cdots,\{\boldsymbol{x},\boldsymbol{u}\}_L\}^{\mathrm{T}}\in R^{L\times N} \tag{3}$$

其中，$\boldsymbol{x}$ 表示传感器，$\boldsymbol{u}$ 表示执行器，$\{\boldsymbol{x},\boldsymbol{u}\}_t\in R^N$ 为 $t$ 时刻 $N$ 个传感器和执行器的测量值。

为了加速网络训练，将 CPSs 数据降采样为每 5 秒一次测量；为了消除不同传感器数据之间量纲的影响，对传感器数据进行归一化处理；执行器数据为分类变量，为了便于提取网络特征，对执行器数据进行独热编码；为了提取每个时刻的上下文信息，同时增加数据集样本数量，利用步长为 $K$ 且长度为 $W$ 的滑动窗口将 CPSs 数据划分为若干个数据窗口，如式(4)所示：

$$\boldsymbol{S}=\{\boldsymbol{S}_1,\boldsymbol{S}_2,\cdots,\boldsymbol{S}_M\}^{\mathrm{T}}\in R^{M\times W\times N} \tag{4}$$

其中，$M=\lfloor(L-W)/K\rfloor+1$ 为数据窗口的个数，$\boldsymbol{S}_t=\{\boldsymbol{u},\boldsymbol{x}\}_{t-W+1:t}\in R^{W\times N}$ 为 $t$ 时刻的数据窗口。对于训练数据，按照 3∶1 的比例将其划分为训练集 $\boldsymbol{S}_{\mathrm{train}}$ 和验证集 $\boldsymbol{S}_{\mathrm{val}}$，且训练数据中不包含异常样本。

### 4.2 编解码网络建模

如图 3 所示，本文结合深度学习中的神经网络与 CPSs 状态空间模型，设计了一种由自适应降噪编码器、上下文信息编码器和解码器构成的编解码网络建模 CPSs 状态空间模型，利用上下文信息编码器拟合状态转换矩阵 $\boldsymbol{F}$ 和控制矩阵 $\boldsymbol{B}$，利用解码器拟合测量矩阵 $\boldsymbol{H}$。自适应降噪编码器以训练数据中前一时刻的传感器数据作为输入，输出泛化性更强的前一时刻系统隐藏状态 $\boldsymbol{z}_{t-1}$；上下文信息编码器以前一时刻数据窗口作为输入，输出信息量更丰富的当前时刻系统隐藏状态 $\boldsymbol{z}_t$；解码器以上述两类编码器输出的系统隐藏状态作为输入，解码出相应的传感器数据 $\tilde{\boldsymbol{x}}_{t-1}$ 与 $\tilde{\boldsymbol{x}}_t$。其中，前一时刻表示为 $t-1$ 时刻，当前时刻表示为 $t$ 时刻。

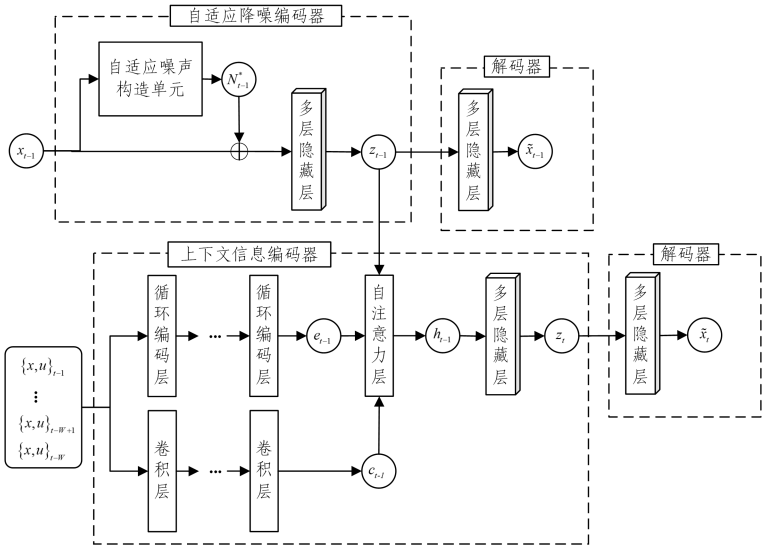



图 3 所提的编解码网络结构

Fig. 3 Structure of the proposed encoder-decoder network

4.2.1 自适应降噪编码器

自编码器作为一种生成网络，以重构原始输入为训练目标。然而训练数据仅代表真实数据分布的局部情况，当训练数据的分布与测试数据的分布存在差异时，自编码器无法较好地拟合测试数据的分布情况，其鲁棒性有待进一步提升。一般地，在自编码器的数据输入层引入随机噪声构建降噪自编码器，可以避免自编码器在训练数据上过度拟合，从而提升自编码器在测试数据上的鲁棒性，并学习到泛化性更强的隐藏表示。例如，在某些自然语言处理任务中，采用 Dropout 对词嵌入产生随机扰动的方式构建降噪自编码器，以提升网络的鲁棒性。与 NLP 任务不同，在 CPSs 异常检测任务中，每个传感器和执行器所收集的数据都是系统状态的标识，对异常判定至关重要，若直接采用 Dropout 对数据随机丢弃，则可能会损失判定异常的关键信息，从而对网络产生不可避免的劣化影响。除 Dropout 方法外，另一种提升网络鲁棒性的方法是对输入数据注入高斯噪声。然而，对于实际的 CPSs 数据，噪声通常不规则地分布在数据空间中，且不同 CPSs 数据中噪声的分布不同。因此，简单的高斯噪声无法充分地表示不同 CPSs 数据中的噪声，直接加入高斯噪声无法提升网络的鲁棒性。

为解决上述问题，提升编解码网络的鲁棒性，本文基于多样本 Dropout[22] 构造自适应降噪编码器 $G_w$。如图 3 所示，自适应降噪编码器主要由自适应噪声构造单元和多层隐藏层组成。首先，自适应噪声构造单元利用训练数据中的传感器数据，在训练过程中拟合数据中噪声的分布模式，生成自适应噪声。然后，利用自适应噪声对训练数据中的传感器数据加噪，以提升编解码网络的鲁棒性，从而减轻噪声带来的干扰，同时可迫使降噪自编码器学习到泛化性更强的系统隐藏状态。

图 4 给出了自适应噪声构造单元的详细结构，主要由多 Dropout 层和自注意力层组成。$m$ 个 Dropout 层扰动传感器数据生成 $m$ 个随机噪声样本；自注意力层利用随机噪声样本拟合数据中噪声的分布模式，生成自适应噪声。

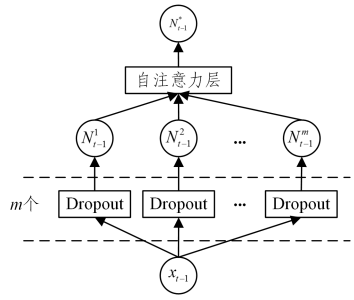



图 4 自适应噪声构造单元结构

Fig. 4 Structure of adaptive noise construction unit

具体地，首先将 $t-1$ 时刻的传感器数据 $\boldsymbol{x}_{t-1}$ 作为自适应噪声构造单元的输入，经多 *Dropout* 层对传感器数据产生随机扰动，生成 $m$ 个随机噪声样本 $\boldsymbol{N}_{t-1}$，如式(5)所示：

$$\boldsymbol{N}_{t-1}=\{\boldsymbol{N}_{t-1}^1,\boldsymbol{N}_{t-1}^2,\cdots,\boldsymbol{N}_{t-1}^m\} \tag{5}$$

其中，单个 Dropout 层在训练迭代过程中利用原始输入数据生成一个随机选择的子集，称为随机噪声样本，用于模拟数据中的未知噪声。$\boldsymbol{N}_{t-1}^i=Dropout(\boldsymbol{x}_{t-1},r^i)$ 为第 $i$ 个随机噪声样本，$r^i$ 为丢弃概率。

然后，自注意力层利用 $m$ 个随机噪声样本拟合数据中噪声的分布模式，生成自适应噪声 $N_{t-1}^*$，其具体计算过程如式(6)－式(8)所示：

$$\boldsymbol{e}_{t-1}=\boldsymbol{v}^{\mathrm{T}}\tanh(\boldsymbol{W}_w\boldsymbol{N}_{t-1}+\boldsymbol{b}_w) \tag{6}$$

$$\boldsymbol{a}_{t-1}^i=\frac{\exp(\boldsymbol{e}_{t-1}^i)}{\sum_{i=1}^m\exp(\boldsymbol{e}_{t-1}^i)} \tag{7}$$

$$\boldsymbol{N}_{t-1}^*=\sum_{i=1}^m\boldsymbol{a}_{t-1}^i\boldsymbol{N}_{t-1}^i \tag{8}$$

其中，$\boldsymbol{e}_{t-1}$ 为打分函数，$\boldsymbol{a}_{t-1}^i$ 为第 $i$ 个随机噪声样本的注意力权重，$\boldsymbol{W}_w$，$\boldsymbol{v}^{\mathrm{T}}$ 和 $\boldsymbol{b}_w$ 为相应需训练的权重矩阵和偏置向量。

最后，利用自适应噪声 $\boldsymbol{N}_{t-1}^*$ 对 $t-1$ 时刻的传感器数据 $\boldsymbol{x}_{t-1}$ 加噪，并将加噪后的传感器数据作为多层隐藏层的输入，从而得到泛化性更强的 $t-1$ 时刻系统的隐藏状态 $\boldsymbol{z}_{t-1}$，具体计算过程如式(9)所示：

$$\boldsymbol{z}_{t-1}=MLP(LayerNorm(\boldsymbol{x}_{t-1}+\boldsymbol{N}^{*}_{t-1})) \tag{9}$$

其中，MLP 表示多层隐藏层。

综上所述，自适应降噪编码器可表示为：

$$\boldsymbol{z}_{t-1}=G_{w}(\boldsymbol{x}_{t-1})$$

其中，$\omega$ 为网络中需训练的参数。

#### 4.2.2 上下文信息编码器

CPSs 数据由系统内若干传感器和执行器以固定的时间间隔收集而来，同时含有复杂的时序上下文信息和空间上下文信息。与异常相邻的时间点可能会包含相似的异常模式，这种局部的时序模式可作为异常判定的显著特征，文中称为时序上下文信息。在系统物理过程中传感器和执行器协同体现系统状态，某些异常可能需要多个传感器数据和执行器数据共同标识，本文中称其为空间上下文信息。上下文信息是异常检测的关键，有效地提取并融合这两类上下文信息，对异常判定至关重要。

为此，本文基于 LSTM 与 CNN 构造上下文信息编码器 $F_{\theta}$，以分别提取并融合各数据窗口内的时序上下文信息和空间上下文信息。如图 3 所示，上下文信息编码器主要由循环编码层、卷积层、自注意力层和多层隐藏层组成。循环编码层和卷积层分别提取数据窗口内的时序上下文信息和空间上下文信息；自注意力层融合这两类上下文信息和自适应降噪编码器输出的系统隐藏状态，得到融合信息；多层隐藏层以融合信息作为输入，推断出当前时刻系统隐藏状态，以提升此隐藏状态中的信息量，从而更准确地检测异常。

具体地，首先将 $t-1$ 时刻的数据窗口 $\boldsymbol{S}_{t-1}$ 作为循环编码层和卷积层的输入，循环编码层利用 LSTM 的时序依赖建模能力，提取 $t-1$ 时刻数据窗口内的时序上下文信息 $\boldsymbol{e}_{t-1}$；卷积层利用 CNN 的空间特征提取能力，提取 $t-1$ 时刻数据窗口内的空间上下文信息 $\boldsymbol{c}_{t-1}$，具体计算过程如式(10)、式(11)所示：

$$\boldsymbol{e}_{t-1}=\Psi_{\mathrm{LSTM}}(\boldsymbol{S}_{t-1}) \tag{10}$$

$$\boldsymbol{c}_{t-1}=MLP(Flatten(\Phi_{\mathrm{CNN}}(\boldsymbol{S}_{t-1}))) \tag{11}$$

其中，$\Psi_{\mathrm{LSTM}}$ 表示多层 LSTM 的内部计算过程，$\Phi_{\mathrm{CNN}}$ 表示多层卷积的内部计算过程，$Flatten(\cdot)$ 表示拉平操作，$MLP$ 表示多层隐藏层。

然后，自注意力层融合t−1时刻的时序上下文信息 $\boldsymbol{e}_{t-1}$、空间上下文信息 $\boldsymbol{c}_{t-1}$ 和系统隐藏状态 $\boldsymbol{z}_{t-1}$，得到$t-1$时刻的融合信息 $\boldsymbol{h}_{t-1}$。其中，$\boldsymbol{h}_{t-1}$ 包含了$t-1$时刻的时序上下文信息和空间上下文信息。

最后多层隐藏层以 $\boldsymbol{h}_{t-1}$ 作为输入，可以更准确地推断出 $t$ 时刻的系统隐藏状态 $\boldsymbol{z}_{t}$，具体计算过程如式(12)、式(13)所示：

$$\boldsymbol{h}_{t-1}=Atten(Stack(Cat(\boldsymbol{z}_{t-1},\boldsymbol{e}_{t-1}),\boldsymbol{c}_{t-1})) \tag{12}$$

$$\boldsymbol{z}_{t}=MLP(\boldsymbol{h}_{t-1}) \tag{13}$$

其中，Cat(·)表示拼接操作，Stack(·)表示堆叠操作，Atten(·)表示式(6)−式(8)中的自注意力计算过程。

综上所述，上下文信息编码器可表示为：

$$\boldsymbol{z}_{t}=F_{\theta}(\boldsymbol{z}_{t-1},\boldsymbol{S}_{t-1})$$

其中，$\theta$ 为网络中需训练的参数。

#### 4.2.3 解码器

如图 3 所示，解码器 $H_{\varphi}$ 由多个隐藏层组成，其利用上述两类编码器产生的系统隐藏状态，可以更准确地解码出相应的传感器数据。具体地，依次以自适应降噪编码器产生的 $t-1$时刻的系统隐藏状态 $\boldsymbol{z}_{t-1}$ 与上下文信息编码器产生的 $t$ 时刻的系统隐藏状态 $\boldsymbol{z}_{t}$ 作为解码器的输入，经多层隐藏层将相应时刻的系统隐藏状态解码为 $\tilde{\boldsymbol{x}}_{t-1}$ 与 $\tilde{\boldsymbol{x}}_{t}$，具体计算过程如式(14)、式(15)所示：

$$\tilde{\boldsymbol{x}}_{t}=\boldsymbol{H}_{\varphi}(\boldsymbol{z}_{t}) \tag{14}$$

$$\tilde{\boldsymbol{x}}_{t-1}=\boldsymbol{H}_{\varphi}(\boldsymbol{z}_{t-1}) \tag{15}$$

其中，$\tilde{\boldsymbol{x}}_{t-1}$ 为$t-1$时刻传感器解码值，$\tilde{\boldsymbol{x}}_{t}$ 为 $t$ 时刻传感器解码值，$\varphi$ 为网络中需训练的参数。两次解码过程中解码器权重共享。

#### 4.2.4 目标函数

共同训练自适应降噪编码器、上下文信息编码器和解码器，利用随机梯度下降算法更新编解码网络参数。设训练集样本个数为 $T$，则目标函数的定义如下：

$$\boldsymbol{L}(\boldsymbol{\omega},\boldsymbol{\theta},\boldsymbol{\varphi})=\sum_{t=1}^{T}\boldsymbol{w}_1\parallel\boldsymbol{x}_{t-1}-\tilde{\boldsymbol{x}}_{t-1}\parallel_2^2+\boldsymbol{w}_2\parallel\boldsymbol{x}_{t}-\tilde{\boldsymbol{x}}_{t}\parallel+\boldsymbol{w}_3\parallel\boldsymbol{z}_{t}-\boldsymbol{z}_{t-1}\parallel_2^2 \tag{16}$$

其中，前两项分别为传感器解码误差；第三项为系统隐藏状态推断误差；超参数 $\boldsymbol{w}_1,\boldsymbol{w}_2,\boldsymbol{w}_3$ 分别为对应项的调节因子。

### 4.3 异常检测

在训练完成的编解码网络上，得到系统隐藏状态和传感器解码值，应用无迹卡尔曼滤波算法递归预测系统状态随时间的联合概率分布，以待测样本中传感器数据与其预测分布的马氏距离作为异常评分来判定异常。检测过程主要分为 3 个步骤：系统隐藏状态预测、异常判定和系统隐藏状态更新。

(1)系统隐藏状态预测。该步骤利用$t-1$时刻系统隐藏状态后验分布的均值 $\bar{\boldsymbol{z}}_{t-1}$ 和协方差 $\bar{\boldsymbol{P}}_{t-1}$ 预测其 $t$ 时刻先验分布的均值 $\hat{\boldsymbol{z}}_{t}$ 和协方差 $\hat{\boldsymbol{P}}_{t}$，及传感器数据的均值 $\boldsymbol{\mu}$ 和协方差 $\boldsymbol{\Sigma}$。初始化系统隐藏状态后验分布的均值 $\bar{\boldsymbol{z}}_{0}=G_{\omega}(\boldsymbol{x}_{0})$ 和协方差 $\bar{\boldsymbol{P}}_{0}=\varepsilon\boldsymbol{I}$，其中 $G_{\omega}$ 为自适应降噪编码器，$\boldsymbol{I}$ 为单位阵，$\varepsilon$ 为接近零的常数，$\boldsymbol{x}_{0}$ 为初始时刻的传感器数据。首先，通过 Julier sigma 函数[23]产生 $n$ 个 sigma 点：$\boldsymbol{Z},\boldsymbol{w}^{m},\boldsymbol{w}^{c}=SigmaFunction(\bar{\boldsymbol{z}}_{t-1},\bar{\boldsymbol{P}}_{t-1})$，其中 $Z,\boldsymbol{w}^{m}\in R^{n},\boldsymbol{w}^{c}\in R^{n}$ 分别为$t-1$时刻的 Sigma 点集合及相应的 Sigma 点权重。然后，将 $\boldsymbol{Z}$ 作为上下文信息编码器 $F_{\theta}$ 的输入，得到 $t$ 时刻系统隐藏状态集合：$\boldsymbol{Y}=F_{\theta}(\boldsymbol{Z},(\boldsymbol{x},\boldsymbol{u})_{t-l:t-1})$。无迹转换函数利用 $\boldsymbol{Y}$ 预测 $t$ 时刻系统隐藏状态先验分布的均值 $\hat{\boldsymbol{z}}_{t}$ 方差 $\hat{\boldsymbol{P}}_{t}$，具体计算过程如式(17)、式(18)所示：

$$\hat{\boldsymbol{z}}_{t}=\sum_{i=1}^{n}\boldsymbol{w}_i^{m}\boldsymbol{Y}_i \tag{17}$$

$$\hat{\boldsymbol{P}}_{t}=\sum_{i=1}^{n}\boldsymbol{w}_i^{c}(\boldsymbol{Y}_i-\hat{\boldsymbol{z}}_{t})(\boldsymbol{Y}_i-\hat{\boldsymbol{z}}_{t})^{\mathrm{T}}+\boldsymbol{Q} \tag{18}$$

其中，$\boldsymbol{Q}$ 为过程噪声，由系统隐藏状态推断误差在验证集上计算协方差矩阵得到。最后，解码器将系统隐藏状态集合 $\boldsymbol{Y}$ 解码为传感器解码值集合：$\boldsymbol{X}=H_{\varphi}(\boldsymbol{Y})$。无迹转换函数利用 $\boldsymbol{X}$ 预测 $t$ 时刻传感器数据分布的均值 $\boldsymbol{\mu}$ 和协方差 $\boldsymbol{\Sigma}$，具体计算过程如式(19)、式(20)所示：

$$\boldsymbol{\mu}=\sum_{i=1}^{n}\boldsymbol{w}_i^m\boldsymbol{X}_i \tag{19}$$

$$\boldsymbol{\Sigma}=\sum_{i=1}^{n}\boldsymbol{w}_i^c(\boldsymbol{X}_i-\mu)(\boldsymbol{X}_i-\mu)^{\mathrm{T}}+\boldsymbol{R} \tag{20}$$

其中，$\boldsymbol{R}$ 为过程噪声，由传感器解码误差在验证集上计算协方差矩阵得到。

(2)异常判定。由传感器数据 $x_t$ 与其预测分布的马氏距离作为异常评分，如式(21)所示：

$$AS(\boldsymbol{x}_t)=\sqrt{(\boldsymbol{x}_t-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_t-\boldsymbol{\mu})} \tag{21}$$

$AS(\boldsymbol{x}_t)$ 取值大于某个阈值时，$\boldsymbol{x}_t$ 为异常。

(3)系统隐藏状态更新。该步骤更新系统隐藏状态后验分布的均值和协方差，具体计算过程如式(22)－式(24)所示：

$$\boldsymbol{K}=[\sum_{i=1}^{n}\boldsymbol{w}_i^c(\boldsymbol{Y}_i-\hat{\boldsymbol{z}}_t)(\boldsymbol{X}_i-\boldsymbol{\mu})^{\mathrm{T}}]\boldsymbol{\Sigma}^{-1} \tag{22}$$

$$\bar{\boldsymbol{z}}_t=\hat{\boldsymbol{z}}_t+\boldsymbol{K}(\boldsymbol{x}_t-\boldsymbol{\mu}) \tag{23}$$

$$\bar{\boldsymbol{P}}_t=\hat{\boldsymbol{P}}_t-\boldsymbol{K}\boldsymbol{\Sigma}\boldsymbol{K}^{\mathrm{T}} \tag{24}$$

其中，$\boldsymbol{K}$ 为卡尔曼增益计算结果，$\bar{\boldsymbol{z}}_t$ 和 $\bar{\boldsymbol{P}}_t$ 分别为 $t$ 时刻系统隐藏状态后验分布的均值和协方差。

### 4.4 方法复杂度分析

本文方法的时间开销主要来源于异常检测阶段中基于无迹卡尔曼滤波算法计算异常评分的 3 个步骤。

假设测试集样本个数为 $l$，$d$ 为网络中各层维度的最大值，$s$ 为数据窗口长度，$k$ 为卷积核大小，$n$ 为 sigma 点个数，$m$ 随机噪声样本个数。系统隐藏状态预测步骤预测系统隐藏状态先验分布的均值和协方差及传感器数据的均值和协方差，该部分包括无迹转换过程、两类编码器编码过程和解码器解码过程，该部分的时间复杂度约为 $O(l(sk+m+n)d^2)$；异常判定步骤利用马氏距离作为异常评分来判定异常，该部分的时间复杂度约为 $O(ld^2)$；系统隐藏状态更新步骤更新系统隐藏状态后验分布的均值和协方差，该部分的时间复杂度约为 $O(l(nd^2+d^3))$。综上分析，本文方法的时间复杂度约为 $O(l(sk+m+2n+1)d^2+ld^3)$，NSIBF 方法的时间复杂度约为 $O(l(s+2n+2)d^2+ld^3)$。

本文方法的时间复杂度略高于 NSIBF 方法，但取得了更高的异常检测指标。综合考虑方法的时间开销和异常检测效果，本文方法的时间开销可以接受。

## 5 实验

### 5.1 数据集及评价指标

本文基于水处理系统的两个数据集 SWaT 和 PUMP 的进行实验，数据中的异常样本点由技术人员模拟真实世界水处理系统的攻击场景获得。两个数据集上的统计信息如表 1 所列。

本文使用准确率(Precision)、召回率(Recall)和 F1 分数评估本文方法的异常检测效果。准确率衡量方法正确识别异常样本的能力，对于误检代价高的场景，需要提高方法的准确率；召回率衡量方法检出异常样本的能力，对于漏检代价高的场景，需要提高方法的召回率；F1 分数为准确率和召回率的调和平均，综合衡量方法的异常检测效果。

表 1 SWaT 和 PUMP 数据集的统计信息

Table 1 Statistics of SWaT and PUMP datasets

| Datasets | Train | Test | Sensor | Actuator | Anomaly ratio/% |
|---|---|---|---|---|---|
| SWaT | 99360 | 89984 | 25 | 26 | 11.99 |
| PUMP | 76901 | 143401 | 44 | 0 | 10.5 |

### 5.2 实验设置

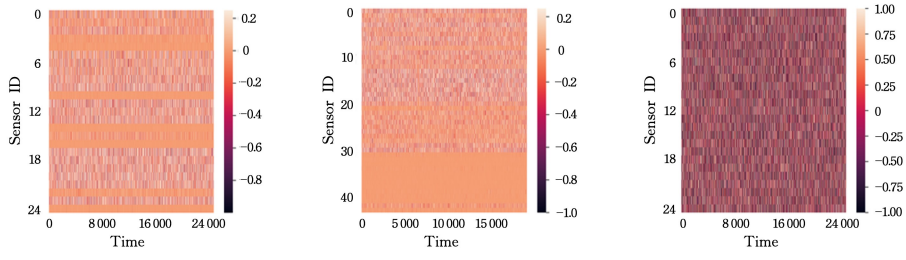
本文实验的主要软件环境为 Ubuntu18.04 系统，Tensorflow2.3 深度学习框架；硬件配置为 Intel i9-9900k@3.6G 8 核心 16 线程 CPU，64GB 内存，单卡 RTX2080Ti 11GB 显存 GPU。

在两个实验数据集上利用随机超参数搜索算法确定网络中各层数量、维度、批量大小、学习率等超参数。确定超参数后，同一数据集上的所有实验均使用相同的超参数。使用均匀负采样算法[24]从验证集生成异常数据，将其作为网络超参数调整过程中计算观测分数所需的数据，最终使用分数最高的网络。此外，网络训练使用 Adam 优化器进行参数优化。

### 5.3 对比实验分析

#### 5.3.1 定性分析

在 CPSs 异常检测任务中，数据中存在未知分布噪声，采用 Dropout 和高斯噪声的方式构建降噪自编码器，对编解码网络的鲁棒性提升有限。图 5(a)－图 5(c)分别展示了 SWAT 数据集中自适应噪声、PUMP 数据集中自适应噪声以及高斯噪声的可视化结果。从图 5(a)和图 5(b)中可以直观地看出，对于不同的 CPSs 数据集，自适应噪声的分布差异显著。对 SWaT 和 PUMP 数据集中的自适应噪声进行正态性检验，结果表明两个数据集中的自适应噪声均不服从高斯分布。因此，采用对原始输入数据注入高斯噪声的方式，无法提升编解码网络的鲁棒性。



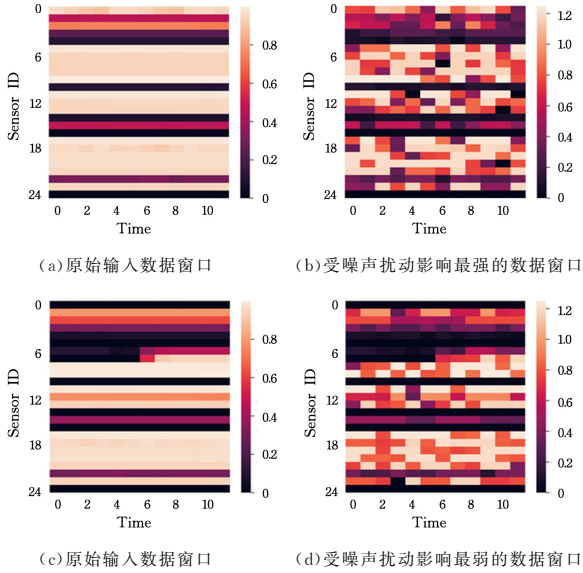

(a)SWaT 数据集中自适应噪声 (b)PUMP 数据集中自适应噪声 (c)高斯噪声

图 5 SWaT 和 PUMP 数据集中的自适应噪声对比

Fig. 5 Comparison of adaptive noise in SWaT and PUMP datasets

以上分析表明，本文提出的自适应降噪编码器可以拟合不同 CPSs 数据中的噪声，提升编解码网络的鲁棒性。

图 6 给出了 SWaT 数据集中自适应噪声对数据窗口扰动强度对比的可视化结果，其中图 6(b)和图 6(d)分别为受噪声扰动影响最强与最弱的数据窗口，图 6(a)和图 6(c)分别为相应的原始输入数据窗口。从图中可以直观地看出，数据窗口中噪声密集时，自适应噪声扰动大，原始输入数据与加噪声后的数据差异更大；数据窗口中噪声稀疏时，自适应噪声扰动小，原始输入数据与加噪声后的数据差异更小，即自适应降噪编码器可以自适应地对不同数据窗口加入相应强度的噪声。




(a)原始输入数据窗口 (b)受噪声扰动影响最强的数据窗口

(c)原始输入数据窗口 (d)受噪声扰动影响最弱的数据窗口

图 6 SWaT 数据集中自适应噪声对数据窗口扰动强度的对比

Fig. 6 Comparison of perturbation strength of adaptive noise to data window in SWaT dataset

以上分析表明，单个 CPSs 数据中噪声不规则地分布在各数据窗口中。本文提出的自适应降噪自编码器，可以有效地拟合各数据窗口中的噪声，提升编解码网络的鲁棒性。

5.3.2 定量分析

为验证本文方法在 CPSs 异常检测任务中的有效性，选取以下 9 种异常检测方法进行对比，分别为：EncDec-AD[10]，USAD[14]，OminiAnomaly[16]，NSIBF[18]，Isolation Forest[25]，Sparse-AE[26]，LSTM-PRED[27]，DAGMM[28]，THOC[29]。

表 2 对比了本文方法与上述 9 种方法在 SWaT 和 PUMP 两个数据集上的准确率、召回率和 F1 值。如表 2 所列，本文方法在两个数据集上的 F1 值和召回率均高于其他对比方法，因为这些方法均未充分利用 CPSs 数据中隐含的上下文信息，且未考虑不同 CPSs 数据中噪声分布模式的差异，不能有效地减轻噪声带来的干扰。

在 SWaT 数据集上，本文方法的 F1 值和召回率均优于其他方法。相比之下，本文方法在准确率大于 0.950 的同时，召回率达到了 0.905。相比 F1 值最优的 NSIBF 方法，本文方法的 F1 值和召回率分别提高了 1.6%和 4.2%，准确率降低了 1.4%。相比 F1 值次优的 THOC 方法，本文方法的 F1 值和召回率分别提高了 5.4%和 10.6%，准确率降低了 1.3%。SWAT 数据集上的实验结果表明，本文方法具有较高的准确率，同时具有最高召回率和 F1 值。

在 PUMP 数据集上，本文方法的 F1 值、召回率和准确率均优于其他方法。相比 F1 值最优的 NSIBF 方法，本文方法的 F1 值、召回率和准确率分别提高了 3.0%，4.9%和 0.5%。PUMP 数据集上的实验结果表明，本文方法能同时提升异常检测的召回率、准确率和 F1 值。THOC 的作者未公开网络训练代码，因此本文无法在 PUMP 测试集上评估其性能。

两个数据集上的定量分析表明，本文方法能充分地利用数据中隐含的上下文信息，且能有效地减轻不同 CPSs 数据中噪声带来的干扰。

表 2 不同方法在 SWaT 和 PUMP 数据集上的准确率、召回率和 F1 对比

Table 2 Comparison of precision, recall and F1 of different methods on SWaT and PUMP datasets

| Methods | SWaT | | | PUMP | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1 | Precision | Recall | F1 |
| Isolation Forest | 0.975 | 0.754 | 0.850 | 0.977 | 0.582 | 0.729 |
| Sparse-AE | **0.999** | 0.666 | 0.799 | 0.798 | 0.737 | 0.767 |
| EncDec-AD | 0.945 | 0.620 | 0.748 | 0.438 | 0.796 | 0.565 |
| LSTM-PRED | 0.996 | 0.686 | 0.812 | 0.925 | 0.581 | 0.714 |
| DAGMM | 0.946 | 0.747 | 0.835 | 0.931 | 0.798 | 0.860 |
| OminiAnomaly | 0.979 | 0.757 | 0.854 | 0.937 | 0.840 | 0.886 |
| USAD | 0.987 | 0.740 | 0.846 | 0.984 | 0.582 | 0.732 |
| THOC | 0.981 | 0.799 | 0.881 | — | — | — |
| NSIBF | 0.982 | 0.863 | 0.919 | 0.988 | 0.840 | 0.908 |
| Ours | 0.968 | **0.905** | **0.935** | **0.993** | **0.889** | **0.938** |

## 5.4 消融实验分析

5.4.1 上下文信息的有效性

本文编解码网络中的上下文信息编码器以固定长度的数据窗口作为输入，分别提取数据窗口内隐含的时序上下文信息和空间上下文信息，并使用自注意力机制融合两类上下文信息。

用仅提取时序上下文信息的方法验证上下文信息编码器中空间上下文信息的有效性；对比两类上下文信息拼接融合与自注意力融合，验证上下文信息编码器是否充分地融合了两类上下文信息。“仅时序上下文”表示上下文信息编码器只提取时序上下文信息，“上下文拼接”表示上下文信息编码器分别提取时序上下文信息和空间上下文信息，并以拼接的方式融合两类上下文信息，“上下文自注意力”表示以自注意力的方式融合两类上下文信息。消融实验的结果为两次实验结果的均值(下文实验也采用此方式)。

如表 3 所列，两个数据集上，“上下文自注意力”的 F1 值最优。在 SWaT 数据集上，“上下文自注意力”的 F1 值和召回率相比“仅时序上下文”分别提升了 1.1%和 3.2%；相比“上下文拼接”F1 值和准确率分别提升了 1.0%和 2.3%。

在 PUMP 数据集上，“上下文自注意力”的 F1 值和召回率相比“仅时序上下文”分别提升了 2.4%和 4.9%；相比“上下文拼接”F1 值和准确率分别提升了 1.2%和 2.5%。

表 3 SWaT 和 PUMP 数据集上上下文信息的有效性分析

Table 3 Context information validity analysis on SWaT and PUMP datasets

| Datasets | Methods | Precision | Recall | F1 |
|---|---|---|---|---|
| SWaT | 仅时序上下文 | **0.982** | 0.863 | 0.919 |
| | 上下文拼接 | 0.946 | 0.895 | 0.920 |
| | 上下文自注意力 | 0.969 | **0.895** | **0.930** |
| PUMP | 仅时序上下文 | **0.988** | 0.840 | 0.908 |
| | 上下文拼接 | 0.954 | 0.889 | 0.920 |
| | 上下文自注意力 | 0.979 | **0.889** | **0.932** |

以上分析表明，本文编解码网络中的上下文信息编码器能充分地提取数据中隐含的时序上下文信息和空间上下文信息，并能有效地融合两类上下文信息。

5.4.2 自适应降噪编码器的有效性

本文编解码网络中的自适应降噪编码器在训练过程中生成自适应噪声，并使用该噪声对训练数据中的传感器数据加噪，以提升编解码网络的鲁棒性，从而减轻噪声带来的干扰。用“高斯噪声”“仅 Dropout”“自适应噪声”和“无噪声”的方法验证自适应降噪编码器的有效性。“高斯噪声”表示采用对原始 CPSs 数据注入高斯噪声的方式实现降噪自编码器，“仅 Dropout”表示采用对原始 CPSs 数据应用 Dropout 的方式实现降噪自编码器，“自适应噪声”即本文提出的自适应降噪自编码器，“无噪声”表示对原始 CPSs 数据不加噪。

如表 4 所列，两个数据集上“自适应噪声”的 F1 值最优。在 SWaT 数据集上，“自适应噪声”的 F1 值和准确率相比“高斯噪声”分别提升了 0.5%和 2.4%；相比“仅 Dropout”F1 值、召回率、准确率分别提升了 3.6%，4.3%和 2.8%；相比“无噪声”F1 值和召回率分别提升了 0.5%和 1.0%。

在 PUMP 数据集上，“自适应噪声”的 F1 值和召回率相比“高斯噪声”分别提升了 5.3%和 9.1%；相比“仅 Dropout”F1 值、召回率、准确率分别提升了 1.9%，1.3%和 2.7%；相比“无噪声”F1 值和准确率分别提升了 0.6%和 1.4%。

表 4 SWaT 和 PUMP 数据集上自适应降噪编码器的有效性分析

Table 4 Validity analysis of adaptive denoising encoder on SWaT and PUMP datasets

| Datasets | Methods | Precision | Recall | F1 |
|---|---|---|---|---|
| SWaT | 无噪声 | 0.969 | 0.895 | **0.930** |
| | 高斯噪声 | 0.944 | **0.916** | 0.930 |
| | 仅 Dropout | 0.940 | 0.862 | 0.899 |
| | 自适应噪声 | **0.968** | 0.905 | **0.935** |
| PUMP | 无噪声 | 0.979 | 0.889 | **0.932** |
| | 高斯噪声 | 0.993 | 0.798 | 0.885 |
| | 仅 Dropout | 0.966 | 0.876 | 0.919 |
| | 自适应噪声 | **0.993** | **0.889** | **0.938** |

此外，SWaT 和 PUMP 两个数据上，“无噪声”的 F1 值均高于或等于“高斯噪声”和“仅 Dropout”，这表明在 CPSs 异常检测任务中，采用 Dropout 和高斯噪声构建降噪自编码器的方式均不可避免地对网络产生了劣化影响。

以上分析表明，本文编解码网络中的自适应降噪编码器可以有效地提升编解码网络的鲁棒性，从而减轻噪声带来的干扰。

**结束语** 考虑到实际的 CPSs 数据中隐含的复杂上下文信息与未知分布噪声，本文提出了一种上下文信息融合与噪声自适应的异常检测方法。该方法利用编解码网络建模 CPSs 状态空间模型。在编解码网络中，利用上下文信息编码器提取数据中的两类上下文信息，并将该信息融合至系统隐藏状态中，以提升此隐藏状态中的信息量，从而更准确地检测异常；针对数据中不规则分布的噪声，利用自适应降噪编码器拟合数据中的噪声分布，生成自适应噪声，并利用该噪声对编解码网络进行加噪训练，以提升编解码网络的鲁棒性。编解码网络训练完成后，得到神经网络形式表示的 CPSs 状态空间模型。在此基础上，结合待测样本的系统隐藏状态和解码出的传感器数据，基于无迹卡尔曼滤波算法计算异常得分。在 SWaT 和 PUMP 两个实际 CPSs 数据集上，与 9 种经典时序异常检测方法进行对比，实验结果表明本文方法各方面性能都较优。同时消融实验结果验证了上下文信息编码器可以有效地提取并融合上下文信息，降噪自编码器可以自适应地拟合数据中的噪声，利用该噪声进行加噪训练有效地提升了编解码网络的鲁棒性。

尽管本文方法取得了较高的异常检测指标，但基于无迹卡尔曼滤波算法计算异常评分，使得编解码网络在异常检测阶段的时间复杂度较高。在未来的工作中，将进一步研究如何降低异常检测阶段的时间复杂度，并进一步提升异常检测的效果。

## 参 考 文 献


[1] YANG P, STANKEVICIUS D, MAROZAS V, et al. Lifelogging data validation model for internet of things enabled personalized healthcare[J]. IEEE Transactions on Systems, Man, and Cybernetics, Systems, 2016, 48(1): 50-64.

[2] YIN C, XI J, SUN R, et al. Location privacy protection based on differential privacy strategy for big data in industrial internet of things[J]. IEEE Transactions on Industrial Informatics, 2017, 14(8): 3628-3636.

[3] BLÁZQUEZ-GARCÍA A, CONDE A, MORI U, et al. A review on outlier/anomaly detection in time series data[J]. ACM Computing Surveys(CSUR), 2021, 54(3): 1-33.

[4] LIN S, CLARK R, BIRKE R, et al. Anomaly detection for time series using vae-lstm hybrid model[C]// Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). USA, IEEE, 2020: 4322-4326.

[5] LI J, GAO H. Survey on sensor network research[J]. Journal of Computer Research and Development, 2008, 45(1): 1-15.

[6] FEI H, XIAO F, LI G H, et al. An anomaly detection method of wireless sensor network based on multi-modals data stream[J]. Chinese Journal of Computers, 2017, 40(8): 1829-1842.

[7] ZHANG C, SONG D, CHEN Y, et al. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data[C]// Proceedings of the AAAI Conference on Artificial Intelligence. USA, AAAI, 2019, 33(1): 1409-1416.

[8] KIM T Y, CHO S B. Web traffic anomaly detection using C-LSTM neural networks[J]. Expert Systems with Applications, 2018, 106: 66-76.

[9] YIN C, ZHANG S, WANG J, et al. Anomaly detection based on

convolutional recurrent autoencoder for IoT time series[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, 52(1): 112-122.

[10] MALHOTRA P, RAMAKRISHNAN A, ANAND G, et al. LSTM-based encoder-decoder for multi-sensor anomaly detection[J]. arXiv: 1607.00148, 2016.

[11] PARK D, HOSHI Y, KEMP C C. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder[J]. IEEE Robotics and Automation Letters, 2018, 3(3): 1544-1551.

[12] LI D, CHEN D, JIN B, et al. MAD-GAN, Multivariate anomaly detection for time series data with generative adversarial networks[C]// Proceedings of International Conference on Artificial Neural Networks. Cham, Springer, 2019: 703-716.

[13] SCHREYER M, SATTAROV T, SCHULZE C, et al. Detection of accounting anomalies in the latent space using adversarial autoencoder neural networks[J]. arXiv: 1908.00734, 2019.

[14] AUDIBERT J, MICHIARDI P, GUYARD F, et al. Usad, Unsupervised anomaly detection on multivariate time series[C]// Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. USA, ACM, 2020: 3395-3404.

[15] HUNDMAN K, CONSTANTINOU V, LAPORTE C, et al. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding[C]// Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. USA, ACM, 2018: 387-395.

[16] SU Y, ZHAO Y, NIU C, et al. Robust anomaly detection for multivariate time series through stochastic recurrent neural network[C]// Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. USA, ACM, 2019: 2828-2837.

[17] DENG A, HOOI B. Graph neural network-based anomaly detection in multivariate time series[C]// Proceedings of the AAAI Conference on Artificial Intelligence. USA, AAAI, 2021, 35(5): 4027-4035.

[18] FENG C, TIAN P. Time series anomaly detection for cyber-physical systems via neural system identification and bayesian filtering[C]// Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. USA: ACM, 2021: 2858-2867.

[19] SCHNEIDER T, QIU C, KLOFT M, et al. Detecting Anomalies within Time Series using Local Neural Transformations[J]. arXiv: 2202.03944, 2022.

[20] LEE E A, SESHIA S A. Introduction to embedded systems, A cyber-physical systems approach[M]. USA, MIT Press, 2016: 181-207.

[21] JULIER S J, UHLMANN J K. Unscented filtering and nonlinear estimation[J]. Proceedings of the IEEE, 2004, 92(3): 401-422.

[22] INOUE H. Multi-sample dropout for accelerated training and better generalization[J]. arXiv: 1905.09788, 2019.

[23] JULIER S J. The scaled unscented transformation[C]// Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301). USA, IEEE, 2002: 4555-4559.

[24] SIPPLE J. Interpretable, multidimensional, multimodal anomaly detection with negative sampling for detection of device failure [C]// Proceeding of the 37th International Conference on Machine Learning. USA, ACM, 2020: 9016-9025.

[25] LIU F T, TING K M, ZHOU Z H, et al. Isolation forest[C]// Proceedings of 2008 Eighth IEEE International Conference on Data Mining. USA, IEEE Computer Society, 2008: 413-422.

[26] NG A. Sparse autoencoder [J]. CS294A Lecture Notes, 2011, 72(2011): 1-19.

[27] GOH J, ADEPU S, TAN M, et al. Anomaly detection in cyber physical systems using recurrent neural networks[C]// Proceedings of 2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE). USA, IEEE, 2017: 140-145.

[28] ZONG B, SONG Q, MIN M R, et al. Deep autoencoding Gaussian mixture model for unsupervised anomaly detection[C]// Proceedings of International Conference on Learning Representations. USA, IEEE, 2018: 1-19.

[29] SHEN L, LI Z, KWOK J. Timeseries anomaly detection using temporal hierarchical one-class network[J]. Advances in Neural Information Processing Systems, 2020, 33: 13016-13026.

**HENG Hongjun**, born in 1968, Ph.D, associate professor, is a member of China Computer Federation. His main research interests include intelligent information processing, natural language, knowledge graph and anomaly detection.

**ZHOU Wenhua**, born in 1998, postgraduate. Her main research interests include anomaly detection and multivariate time series data.

（责任编辑：何杨）