USAD：多变量时间序列的无监督异常检测

朱利安·奥迪贝尔
julien.audibert@orange.com
欧洲通信委员会
比奥，法国橙
法国索菲亚安提波利斯

皮埃特罗·米基亚尔迪
pietro.michiardi@eurecom.fr
欧洲通信委员会
法国比奥

弗雷德里克·盖亚尔
frederic.guyard@orange.com
橙色实验室
法国索菲亚安提波利斯

塞巴斯蒂安·马蒂
sebastien.marti@orange.com
橙子
法国索菲亚安提波利斯

玛丽亚·A·祖卢阿加
zuluaga@eurecom.fr 欧洲通信委员会
法国比奥

摘要

IT 系统的自动监控是 Orange 目前面临的挑战。鉴于其 IT 运营的规模和复杂性，用于推断正常和异常行为的随时间推移的测量数据所需的传感器数量急剧增加，使得传统的基于专家的监控方法变得缓慢或容易出错。在本文中，我们提出了一种基于逆向训练自动编码器的快速稳定的方法，称为多变量时间序列无监督异常检测 (USAD)。其自动编码器架构使其能够以无监督的方式学习。对抗性训练及其架构的使用使其能够在快速训练的同时隔离异常。我们通过在五个公共数据集上的实验研究了我们的方法的特性，从而证明了其鲁棒性、训练速度和高异常检测性能。通过使用 Orange 的专有数据进行可行性研究，我们能够验证 Orange 对可扩展性、稳定性、鲁棒性、训练速度和高性能的要求。

CCS概念

• 计算方法→神经网络；异常检测；无监督学习；•应用计算；

• 关键词

异常检测；多元时间序列；神经网络；自动编码器；对抗网络；无监督学习；监督

ACM 参考格式：
Julien Audibert、Pietro Michiardi、Frédéric Guyard、Sébastien Marti 和 Maria A. Zuluaga。 2020. USAD：多元时间序列的无监督异常检测。第 26 届 ACM SIGKDD 知识发现和数据挖掘会议 (KDD '20) 会议记录，2020 年 8 月 23-27 日，虚拟活动，美国加利福尼亚州。 ACM，美国纽约州纽约市，10 页。https://doi.org/10.1145/3394486.3403392

KDD '20，2020 年 8 月 23-27 日，虚拟活动，美国加利福尼亚州

ISBN 978-1-4503-7998-4/20/08。
https://doi.org/10.1145/3394486.3403392

1 介绍

IT 系统监控是对系统可测量事件和输出的监督过程，这些事件和输出可作为系统正常运行的参考。通过分析与参考值的偏差来确定是否存在故障。以往，这种分析是由系统监控专家完成的，他们会为每个测量事件/输出设定正常行为阈值。如果测量值超出了专家定义的相关阈值，则认为系统运行不符合预期。由于 Orange 当前 IT 运营的规模和复杂性，获取测量值所需的传感器数量急剧增加，使得传统的基于专家定义的阈值的方法因不可扩展而不再适用。在这种情况下，IT 系统监控的自动化已成为必然。自动化 IT 系统监控需要开发能够观察传感器获取的不同测量值并由此推断正常和异常行为的方法。

检测一组随时间推移而相互关联的测量值的意外行为是一门名为多变量时间序列异常检测的活跃研究学科 [2]。在过去的几年中，已经开发出许多方法来解决这个问题。最常用的技术包括基于距离的技术，例如 K 最近邻 [3]、聚类（例如 K 均值 [9]）和使用一类 SVM 进行分类 [11]。然而，当今的 IT 系统已经达到了一定的复杂程度，不再允许使用这些方法。事实上，随着维数的增加，由于维数灾难，这些技术通常性能不佳。最近，基于深度学习的无监督异常检测方法能够推断时间序列之间的相关性，从而识别异常行为，这种能力受到了广泛关注 [12][20][17][18]。

在用于检测时序数据异常的深度学习方法中，基于循环神经网络[7] (RNN) 的方法非常流行。然而，RNN 的结果以计算量大和训练耗时长而闻名。因此，RNN 的时间成本很高，

能耗和二氧化碳₂排放。对于Orange而言，使用高度可扩展且低能耗的方法至关重要。事实上，Orange正通过其“绿色IT与网络”计划不断努力提高能源效率。这些高可扩展性和迈向绿色人工智能（GreenAI）[16]的制约因素，迫使我们重新思考待实施的深度学习方法的重要特性。因此，开发具有高算法效率的实施方法势在必行。

近期，其他备受关注的深度学习方法还包括基于生成对抗网络的方法 [5]。然而，由于模式崩溃和不收敛等问题，GAN 的训练并非易事 [1]。在 Orange 考虑将这些方法实施并部署到生产环境中时，缺乏稳定性是一个主要障碍。生产环境需要开发能够定期重新训练的稳健方法。

在本文中，我们提出了一种名为“多变量时间序列无监督异常检测”（USAD）的新方法，该方法基于一种自动编码器架构 [15]，其学习方式受到 GAN 的启发。USAD 背后的原理是，其编码器-解码器架构的对抗性训练使其能够学习如何放大包含异常输入的重构误差，同时相比基于 GAN 架构的方法而言，其稳定性更高。该架构使其能够快速训练，在可扩展性和算法效率方面满足 Orange 的期望。本文的主要贡献如下：

- 我们在对抗训练框架内提出了一种编码器-解码器架构，可以结合自动编码器和对抗训练的优势，同时弥补每种技术的局限性。
- 我们对公开可用的数据集进行了实证研究，以分析所提出方法的鲁棒性、训练速度和性能。
- 我们使用 Orange 的专有数据进行可行性研究，以分析所提出的方法是否满足公司对可扩展性、稳定性、稳健性、训练速度和高性能的要求。

本文的其余部分组织如下。第 2 节讨论了检测多变量时间序列中无监督异常的方法。第 3 节讨论了我们方法的细节。第 4 节和第 5 节描述了实验并展示了我们方法的当前最佳性能。

2 相关工作

时间序列异常检测是一项复杂的任务，已被广泛研究[6]。目前已提出的不同分类方法包括聚类[9]、基于密度的[11]、基于距离的[3]和基于隔离的[10]。

除传统方法外，基于深度学习的无监督异常检测方法推断时间序列间相关性的能力近来备受关注 [12, 18, 20]。深度自编码高斯混合模型 (DAGMM) [21] 联合考虑了深度自编码器和高斯混合模型，以对多维数据的密度分布进行建模。多尺度卷积递归编解码器 (MSCRED) [20] 联合考虑了时间依赖性、噪声鲁棒性和对异常严重程度的解释。LSTM-VAE [14] 将 LSTM 与变分自编码器 (VAE) 结合起来，用 LSTM 替换 VAE 中的前馈网络。对抗学习异常检测 (ALAD) [19] 基于双向 GAN，它为异常检测任务获取对抗学习的特征。 LSTM-VAE 通过 LSTM 网络对时间序列的时间依赖性进行建模，并获得比传统方法更好的泛化能力。最近，Su 等人提出了一种用于多变量时间序列异常检测的随机循环神经网络 OmniAnomaly，它通过随机变量连接和平面正则化流学习稳健的多变量时间序列表示，并使用重构概率来确定异常 [17]。然而，这些方法以牺牲训练速度为代价获得了良好的结果。事实上，这些方法都没有将训练时间（即能耗）纳入其性能标准中。这就是为什么今天 Orange 有必要开发出在异常检测方面性能与最先进水平相当的方法，同时倾向于采用能够快速且节能地进行训练的架构。

3 方法

我们首先在第 3.1 节中对要解决的问题进行形式化描述。在第 3.2 节中，我们给出了方法的公式。最后，在第 3.3 节中，我们描述了该方法的实现。

3.1 问题表述

单变量时间序列是一系列数据点

$$\mathcal{T} = \{x_1, \ldots, x_T\},$$

每个变量都是在特定时间 t 测量的一个过程的观测值。单变量时间序列在每个时间点只包含一个变量，而多变量时间序列则同时记录多个变量；我们将多变量时间序列表示为 T = {x₁,., xT }, x ∈ R^m。在本文中，我们关注更一般的多变量时间序列，因为单变量设置是 m = 1 的多变量设置的一个特例。

现在考虑一个无监督学习问题，其中 T 作为训练输入。异常检测是指识别一个未见观测值 xˆt (t > T) 的任务，该任务基于 xˆt 与 T 存在显著差异这一事实，因此假设 T 仅包含正常点。未见样本 xˆt 与正常集 T 的差异量由异常分数来衡量，然后将其与阈值进行比较以获得异常标签。

为了对当前时间点与先前时间点之间的依赖关系进行建模，我们现在定义 Wt，即给定时间 t 的长度为 K 的时间窗口：

$$W_t = \{\mathbf{x}_{t-K+1}, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t\}. \quad (1)$$

可以将原始时间序列 T 转换为窗口序列 W = {W₁,., WT } 以用作训练输入。给定一个二元变量 y ∈ {0, 1}，我们的异常检测问题的目标是为未见窗口 Wt (t > T) 分配一个标签 yt，以指示在时间 t 检测到异常（即 yt = 1）或未检测到异常（yt = 0），这基于窗口的异常得分。为了简单起见并且不失一般性，我们将使用 W 表示训练输入窗口，使用 W- 表示未见输入窗口。

3.2 无监督异常检测

自动编码器 (AE) [15] 是一种无监督的人工神经网络，由编码器 E 和解码器 D 组成。编码器接收输入 X，并将其映射到一组隐变量 Z；解码器则将隐变量 Z 映射回输入空间，形成重构向量 R。原始输入向量 X 与重构向量 R 之间的差值称为重构误差。因此，训练目标是最小化该误差。其定义为：

$$\mathscr{L}_{AE} = \|X - AE(X)\|_2, \tag{2}$$

$$AE(X) = D(Z), \quad Z = E(X)$$

在哪里

并且$/\!/\cdot/\!/_2$表示L2范数。

基于自动编码器的异常检测利用重建异常分数作为误差。分数高的点被视为异常。训练时仅使用来自正常数据的样本。在推理阶段，AE 能够很好地重建正常数据，而无法重建 AE 未遇到的异常数据。但是，如果异常太小，即相对接近正常数据，则重建误差会很小，因此无法检测到异常。这是因为 AE 的目标是尽可能好地重建输入数据（尽可能接近正常值）。为了解决这个问题，AE 应该能够在进行良好的重建之前识别输入数据是否不包含异常。能够判断输入样本是否正常是生成对抗网络 (GAN) 的特征 [5]。GAN 是一种无监督人工神经网络，基于两个同时训练的网络之间的双人极小极大对抗博弈。一个网络，即生成器（G），旨在生成真实数据，而第二个网络充当鉴别器（D），试图区分真实数据和 G 生成的数据。G 的训练目标是最大化 D 犯错的概率，而训练目标 D 是最小化其分类错误。

与基于 AE 的方法类似，基于 GAN 的异常检测使用正常数据进行训练。训练完成后，判别器将用作异常检测器。如果输入数据与学习到的数据分布不同，判别器会将其视为来自生成器，并将其归类为伪数据，即异常。然而，由于模式崩溃和不收敛 [1] 等问题，GAN 的训练并非易事，这些问题通常归因于生成器和判别器之间的不平衡。

我们提出的无监督异常检测 (USAD) 方法，被设计为一个基于两阶段对抗训练框架的 AE 架构。一方面，这种方法能够通过训练一个能够识别输入数据何时不包含异常的模型来克服 AE 的固有局限性，从而实现良好的重构。另一方面，AE 架构能够在对抗训练过程中获得稳定性，从而解决 GAN 中遇到的崩溃和非收敛模式问题。

USAD 由三个元素组成：一个编码器网络 E 和两个解码器网络 D_1和 D_2。如图 1 所示，这三个元素连接成一个由两个自动编码器 AE_1和 AE_2组成的架构，这两个自动编码器共享同一个编码器网络：

$$AE_1(W) = D_1(E(W)), \quad AE_2(W) = D_2(E(W)) \tag{3}$$

公式 3 中的架构分两个阶段进行训练。首先，训练两个 AE 学习重建正常输入窗口 W。其次，以对抗的方式训练两个 AE，其中 AE1 会试图欺骗 AE2，而 AE2 则旨在学习数据何时是真实的（直接来自 W）或重建的（来自 AE_1）。更多细节将在下面提供。

第一阶段：自动编码器训练。在第一阶段，目标是训练每个自动编码器（AE）来复现输入。输入数据 W 由编码器 E 压缩到潜在空间 Z，然后由每个解码器重建。根据公式 2，训练目标是：

$$\begin{aligned}\mathscr{L}_{AE_1} &= \|W - AE_1(W)\|_2 \\ \mathscr{L}_{AE_2} &= \|W - AE_2(W)\|_2\end{aligned} \tag{4}$$

第二阶段：对抗训练。在第二阶段，目标是训练 AE_2区分真实数据和来自 AE1 的数据，并训练 AE1 欺骗 AE2。来自 AE1 的数据再次通过 E 压缩为 Z，然后由 AE_2重建。使用对抗训练配置，AE_1的目标是最小化 W 与 AE_2输出之间的差异。AE_2的目标是最大化这一差异。AE_1的训练目标是判断它是否能够成功欺骗 AE_2，而 AE_2则能够区分 AE_1重建的候选集与真实数据。训练目标是：

$$\min_{AE_1}\max_{AE_2} \|W - AE_2(AE_1(W))\|_2 \tag{5}$$

造成以下损失

$$\begin{aligned}\mathscr{L}_{AE_1} &= +\|W - AE_2(AE_1(W))\|_2 \\ \mathscr{L}_{AE_2} &= -\|W - AE_2(AE_1(W))\|_2\end{aligned} \tag{6}$$

两阶段训练。在我们的架构中，自动编码器具有双重目的。AE_1最小化 W 的重构误差（阶段 1），并最小化 W 与 AE2 的重构输出之间的差异（阶段 2）。与 AE1 一样，AE2 最小化 W 的重构误差（阶段 1），但随后最大化由 AE_1重构的输入数据的重构误差（阶段 2）。每个 AE 的双重目的训练目标可以用公式 4 和公式 6 的进化方案组合来表示，其中各部分的比例随时间演变：

$$\mathscr{L}_{AE_1} = \frac{1}{n}\|W - AE_1(W)\|_2 + \left(1 - \frac{1}{n}\right)\|W - AE_2(AE_1(W))\|_2 \tag{7}$$

$$\mathscr{L}_{AE_2} = \frac{1}{n}\|W - AE_2(W)\|_2 - \left(1 - \frac{1}{n}\right)\|W - AE_2(AE_1(W))\|_2 \tag{8}$$

是在算法 1 中进行了总结。

需要注意的是，AE_2并非严格意义上的 GAN 判别器，因为如果其输入是原始数据，则公式 4 中的损失函数会起作用。如果其输入是重构数据，则公式 5-6 中的目标函数会起作用。推理。在检测阶段（算法 2），异常分数定义为：

$$\mathscr{A}(\widehat{W}) = \alpha\|\widehat{W} - AE_1(\widehat{W})\|_2 + \beta\|\widehat{W} - AE_2(AE_1(\widehat{W}))\|_2 \tag{9}$$

其中 $\alpha + \beta = 1$，用于参数化假阳性和真阳性之间的权衡。如果 α 大于 β，则减少真阳性的数量，并减少

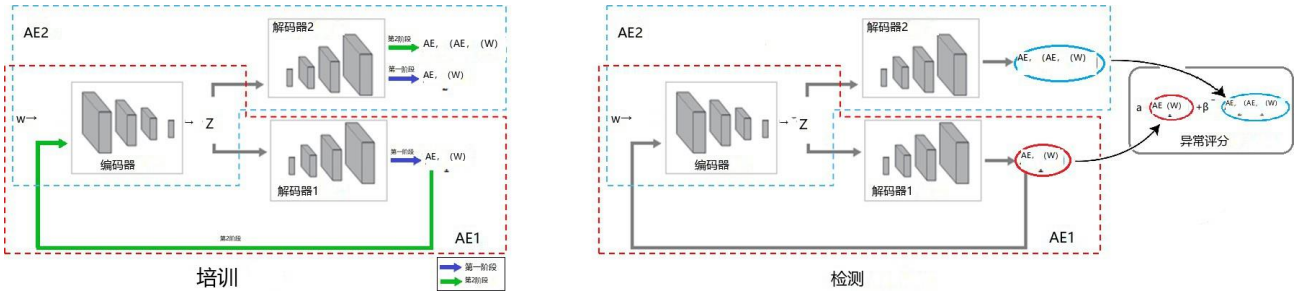

图 1：提出的架构展示了训练阶段（左）和检测阶段（右）的信息流。

算法1 USAD训练算法

输入： 普通窗口数据集 W = {W , ..., W }， 数字历元 N

输出： 训练好的AE1， AE2 E， D1，D2←初始化权重n←1

重复

对于 t = 1 到 T
做

$$Z_t \leftarrow E(W_t)$$
$$W_t^{1'} \leftarrow D_1(Z_t)$$
$$W_t^{2'} \leftarrow D_2(Z_t)$$
$$W_t^{2''} \leftarrow D_2(E(W_t^{1'}))$$
$$\mathcal{L}_{AE_1} \leftarrow \frac{1}{n}\left\|W_t - W_t^{1'}\right\|_2 + \left(1-\frac{1}{n}\right)\left\|W_t - W_t^{2''}\right\|_2$$
$$\mathcal{L}_{AE_2} \leftarrow \frac{1}{n}\left\|W_t - W_t^{2'}\right\|_2 - \left(1-\frac{1}{n}\right)\left\|W_t - W_t^{2''}\right\|_2$$
$$E, D_1, D_2 \leftarrow$$

结束于

$$n \leftarrow n+1$$

直到 n = N

算法2 USAD检测算法

输入：测试窗口数据集 W : (W1, ..., WT ∗)， 阈值 λ， 参数 α 和 β

输出：标签 y : {y1, ..., yT ∗ }

对于 t = 1 到 T*

$$W_t^{1'} \leftarrow D_1(E(W_t))$$
$$W_t^{2''} \leftarrow D_2(E(W_t^{1'}))$$
$$A \leftarrow \alpha \| W_t - W_t^{1'} \|_2 + \beta \| W_t - W_t^{2''} \|_2$$

如果 A ≥ λ 则

$$y_t \leftarrow 1$$

别的

$$y_t \leftarrow 0$$

结束

如果结束为

误报率。相反，如果取 α 小于 β，则会增加真阳性的数量，但代价是也会增加假阳性的数量。我们将 α < β 表示为高检测灵敏度场景，将 α > β 表示为低检测灵敏度场景。这种参数化方案具有很高的工业应用价值。它允许使用单个训练模型在推理过程中获得一组不同的灵敏度异常分数。这将在第 5.2 节中进一步说明。

3.3 执行

我们的异常检测方法分为三个阶段。第一个阶段是数据预处理，它与训练和检测阶段相同，在此阶段，数据被归一化并分割成长度为 K 的时间窗口。第二个阶段用于训练该方法。训练是离线的，旨在捕捉多变量时间序列中预定义部分（几周/几个月）的正常行为，并为每个时间窗口生成异常分数。此离线训练过程可以定期自动执行，注意选择的训练周期不要包含太多被视为异常的周期。最后一个阶段是异常检测。它使用在第二阶段。当新的时间窗口到来时，该模型用于获取异常分数。如果某个时间窗口的异常分数高于定义的异常阈值，则该时间窗口被判定为异常。

表 1：基准数据集。（%）是数据集中异常数据点的百分比。

数据集	火车	测试	方面	异常（%）
扑打	496800	449919	51	11.98
瓦迪	1048571	172801	123	5.99
贴片	708405	708420	28*38	4.16
斯玛特	135183	427617	55*25	13.13
语言学硕士	58317	73729	27*55	10.72
橙子	2781000	5081000	33	33.72

4 实验设置

本节介绍实验和可行性研究中使用的数据集和性能指标。

4.1 公共数据集

我们的实验使用了五个公开的数据集。表1总结了数据集的特征，下面简要描述它们。

安全水处理 (SWaT) 数据集。SWaT 数据集[1]是现实世界中一个生产过滤水的工业水处理厂的缩小版 [4]。收集的数据集 [13] 包含 11 天的连续运行数据：其中 7 天为正常运行，4 天为攻击场景。

供水系统 (WADI) 数据集。该数据集[2]来自 WADI 测试平台，该测试平台是 SWaT 测试平台的扩展 [13]。该数据集包含 16 天的连续运行数据，其中 14 天为正常运行，2 天为攻击场景。

服务器机器数据集。SMD 是一个由大型互联网公司收集并公开的、为期 5 周的新数据集[3][17]。它包含来自 28 台服务器的数据，每台服务器由 m = 33 个指标监控。SMD 被划分为两个大小相等的子集：前半部分为训练集，后半部分为测试集。

土壤湿度主动/被动监测 (SMAP) 卫星和火星科学实验室 (MSL) 探测车数据集。SMAP 和 MSL 是两个真实世界的公共数据集，由 NASA [8] 的专家标注。它们分别包含 55/27 个实体的数据，每个实体由 m = 25/55 个指标监测。

4.2 可行性研究：Orange 的数据集

我们的可行性研究是基于一个专门为此目的收集的内部数据集进行的。收集的数据来自 Orange 网站上广告网络的技术和业务指标。数据总共代表 m = 33 个连续变量，包括 27 个技术测量值和 6 个业务测量值。数据集分为两个子集：一个对应于约 32 天活动的训练集和一个对应于约 60 天活动的测试集。我们选择了 60 天的测试集，这对应于 Orange 的关键时期。为了获得训练集，我们选择了公司之前连续未发生任何重大事故的日子。我们能够获得一个包含 32 个主要正常日子的训练集。测试集中的异常由领域专家根据事故报告标记。其主要特征如表 1 所示。

4.3 评估指标

使用准确率（P）、召回率（R）和F1分数（F1）来评估异常检测性能：

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

其中 TP 表示真正例 (True Positive)，FP 表示假正例 (False Positive)，FN 表示假负例 (False negative)。我们认为，只要窗口包含的某个点被检测到异常，该窗口就会被标记为异常。

在 [17] 中，作者使用平均精确率和平均召回率计算 F1 分数。为了完整起见，我们在将我们的方法与他们的基准进行比较时报告了这一指标。我们将该指标记为 F1* 分数：

$$F1^* = 2 \cdot \frac{\bar{P} \cdot \bar{R}}{\bar{P} + \bar{R}}$$

其中 $\bar{P}$ 和 $\bar{R}$ 分别表示平均准确率和召回率。通过比较每个评估的结果来评估性能。

使用带注释的地面实况进行评估的方法。为了能够与 [17] 提出的基准进行直接比较，我们使用了他们的方法。异常观测通常以连续异常段的形式出现。在这种方法中，如果至少一个异常段的观测被正确检测到，则该段的所有其他观测也被视为正确检测，即使它们实际上并未被正确检测到。地面实况异常段之外的观测按常规处理。我们将这种方法称为点调整。我们还在不属于基准 [17] 的两个数据集（SWaT 和 WADI）上评估了没有点调整的性能。

5 实验与结果

我们通过评估 USAD 的性能并将其与其他最先进的方法 (5.1) 进行比较、分析不同参数如何影响该方法的性能 (5.2)、估计其计算性能 (5.3) 以及通过消融研究 (在每次研究中，我们抑制其中一个训练阶段 (5.4)) 来研究 USAD 的关键特性。最后，在第 5.5 节中，我们报告了使用 Orange 内部数据的可行性研究，以证明 USAD 满足部署到生产中所需的要求。

5.1 总体性能

为了展示 USAD 的整体性能，我们将其与五种用于检测多变量时间序列异常的无监督方法进行了比较。这五种方法是：孤立森林 (IF) [10]、自动编码器 (AE)、LSTM-VAE [14]、DAGMM [21] 和 OmniAnomaly [17]。由于并非所有用于比较的异常检测方法都提供选择异常阈值的机制，我们测试了每个模型的可能异常阈值，并报告与最高 F1 分数相关的结果。表 2 详细列出了所有方法在公共数据集上获得的性能结果。上面展示了使用 SWaT 和 WADI 数据集获得的结果，而表格的下面报告了使用其余三个数据集从 [17] 提出的基准测试中获得的结果。在没有点调整数据集的情况下，USAD 在 SWaT、MSL、SMAP 和 WADI 上的表现优于所有方法，其 F1 在 SMD 数据集上排名第二。在所有数据集上（表 3）的平均表现最佳，比当前最先进的方法 [17] 高出 0.096。

总体而言，IF 和 DAGMM 的性能最差。这两种无监督异常检测方法没有利用观测值之间的时间信息。对于时间序列而言，时间信息至关重要且必要，因为观测值具有依赖性，而历史数据对于重建当前观测值非常有用。在 USAD 中，无论是训练还是检测，输入都是包含时间关系的观测值序列，以保留这些信息。

尽管在大多数数据集中结果相对较差，但 IF 在 WADI 数据集上通过点调整获得了最高的 F1 分数。这可以通过点调整方法和 WADI 数据集的性质来解释。IF 独立考虑每个观测值/时间点，并将标签分配给单个时间点，而不是窗口。WADI 的异常持续存在，而点调整则验证了整体性。

[1] https://github.com/JulienAu/Anomaly_Detection_Tuto

[2] https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/#wadi

[3] https://github.com/smallcowbaby/OmniAnomaly

表 2：性能比较。上图：SWaT 和 WADI 数据集上，使用和不使用点调整（Without）的准确率 (P)、召回率 (R) 和 F1 得分。下图：使用 [17] 提出的带点调整的基准。报告了 P、R F1 和 F1*。

方法	扑打						瓦迪					
	没有			和			没有			和		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
AE	0.9903	0.6295	0.7697	0.9913	0.7040	0.8233	0.9947	0.1310	0.2315	0.3970	0.3220	0.3556
如果	0.9512	0.5884	0.7271	0.9620	0.7315	0.8311	0.2992	0.1583	0.2071	0.6241	0.6155	**0.6198**
长短期记忆-变分自动编码器	0.9897	0.6377	0.7756	0.7123	0.9258	0.8051	0.9947	0.1282	0.2271	0.4632	0.3220	0.3799
差分自适应多目标模型	0.4695	0.6659	0.5507	0.8292	0.7674	0.7971	0.0651	0.9131	0.1216	0.2228	0.1976	0.2094
OmniAnomaly	0.9825	0.6497	0.7822	0.7223	0.9832	0.8328	0.9947	0.1298	0.2296	0.2652	0.9799	0.4174
万能												
美国田径协会	0.9851	0.6618	**0.7917**	0.9870	0.7402	**0.8460**	0.9947	0.1318	**0.2328**	0.6451	0.3220	0.4296

方法	贴片				斯玛特				语言学硕士			
	P	R	F1	F1*	P	R	F1	F1*	P	R	F1	F1*
AE	0.8825	0.8037	0.8280	0.8413	0.7216	0.9795	0.7776	0.8310	0.8535	0.9748	0.8792	0.9101
如果	0.5938	0.8532	0.5866	0.7003	0.4423	0.5105	0.4671	0.4739	0.5681	0.6740	0.5984	0.6166
长短期记忆-变分自动编码器	0.8698	0.7879	0.8083	0.8268	0.7164	0.9875	0.7555	0.8304	0.8599	0.9756	0.8537	0.9141
差分自适应多目标模型	0.6730	0.8450	0.7231	0.7493	0.6334	0.9984	0.7124	0.7751	0.7562	0.9803	0.8112	0.8537
OmniAnomaly	0.9809	0.9438	**0.9441**	**0.9620**	0.7585	0.9756	0.8054	0.8535	0.9140	0.8891	0.8952	0.9014
万能												
美国田径协会	0.9314	0.9617	0.9382	0.9463	0.7697	0.9831	**0.8186**	**0.8634**	0.8810	0.9786	**0.9109**	**0.9272**

表 3：使用点调整的所有数据集的平均性能（±标准差）。

	P	R	F1	F1*
AE	0.77(0.21)	0.76(0.24)	0.73(0.19)	0.86 (0.04)
如果	0.64(0.17)	0.68(0.11)	0.62(0.12)	0.60 (0.09)
长短期记忆-变分自动编码器	0.72(0.15)	0.80(0.25)	0.75 (0.18)	0.86 (0.04)
差分自适应多目标模型	0.62(0.21)	0.76(0.29)	0.65(0.22)	0.79 (0.04)
办公自动化	0.73(0.25)	**0.95(0.04)**	0.78(0.19)	0.91(0.04)
美国田径协会	**0.84(0.12)**	0.80(0.25)	**0.79(0.18)**	**0.91(0.04)**

异常被很好地检测到。因此，IF 几乎不受其不良预测 (FP) 的影响，因为不良预测一次仅影响一个观测值，而点调整则具有优势，尽管可能错过了几个异常，但它可以验证整个良好预测部分。

不同的是，AE 和 LSTM-VAE 使用连续观测值作为输入，这使得这两种方法能够保留时间信息。无论输入窗口中是否存在异常，这些方法都能执行最佳重建。这使得它们无法检测到接近正常数据的异常。USAD 通过对抗训练弥补了基于 AE 方法的这一缺陷。OmniAnomaly 也存在类似的情况，因为它没有允许放大“轻微”异常的机制。

5.2 参数的影响

在本节中，我们研究不同参数和因素对 USAD 性能的影响。所有实验均使用 SWaT 数据集完成。

我们研究的第一个因素是 USAD 如何响应不同的训练数据下采样率。下采样通过减小数据量来加速学习，同时还具有去噪效果。但是，如果丢失过多信息，则可能产生负面影响。图 2(A) 总结了使用 5 种不同采样率 [1, 5, 10, 20, 50] 获得的结果。结果表明，USAD 的性能对下采样相对不敏感，在各种采样率下性能相对稳定。这表明下采样率的选择对该方法并不重要。在我们的实验中，我们选择了 5 的采样率。这是在训练数据去噪和限制信息丢失之间的最佳平衡。此外，它还可以将 USAD 所需的训练时间缩短 5 倍。

我们研究的第二个因素是 USAD 如何响应数据中不同的窗口大小。窗口大小会影响可检测到的异常行为类型，直接影响异常检测的速度，因为检测速度由窗口的持续时间决定。图 2(B) 展示了五种不同窗口大小 K ∈ [5, 10, 20, 50, 100] 的结果。窗口大小 K = 10 时效果最佳。窗口较小时，USAD 可以更快地检测到行为变化，因为每次观察对异常分数的影响更大。窗口过大需要等待更多观察才能检测到异常。然而，较大的窗口可以检测到持续时间较长的异常。然而，如果异常时间过短，它可能隐藏在过大窗口的点数中。对于 Orange 来说，小窗口更好，因为它可以加快训练速度和检测速度。

隐变量 Z 位于 m 维空间，假设该空间小于原始数据之一。我们研究了 m 对 USAD 性能的影响。图 2(C) 展示了 m ∈ [5, 10, 20, 40, 100] 的结果。结果表明，Z 的维数过小会导致编码时信息大量丢失。

解码器无法恢复，从而导致性能不佳。另一个极端是，使用较大的 m 值
导致记忆训练数据并导致
性能。相反，m 的中间值似乎对性能没有很大的影响，显示出相对较高且稳定的 F1 分数。

USAD 的训练假设是训练集仅由正常样本组成。但实际上，训练集并非完全由正常数据组成。因此，我们研究当在训练数据集中注入噪声时，如果打破这一假设，方法的性能会受到何种程度的影响。我们在随机选择的时间点（占训练数据集大小的一定百分比）注入高斯噪声（$\mu = 0$，$\sigma = 0.3$）。我们将该百分比从 1% 到 30% 不等。噪声在下采样（采样率 = 5）后注入，以避免下采样导致的噪声衰减。

图 2(D) 展示了我们的方法在噪声水平增加时，P、R 和 F1 方面的性能。USAD 展现了其稳健性，在高达 5% 的噪声水平下仍能保持相对恒定的高性能。当训练集噪声达到 10% 时，性能开始略有下降。然而，以 F1 分数衡量的整体性能仍然良好。有趣的是，这种性能下降是由较低的精确度造成的。由于召回率保持相对恒定，这意味着当训练集噪声较高时，该方法更容易检测到假阳性。这表明，随着噪声开始增加，USAD 不再能够正确学习训练集中存在的最复杂行为。因此，测试集中的假阳性数量会增加，因为 USAD 将复杂的正常行为检测为异常。最后，在高噪声水平（30%）下可以观察到性能显著下降。然而，在生产环境中训练期间出现如此高的异常率是不现实的。这意味着在给定时间段内，30% 的样本是未被注意到的异常。由于生产中存在如此多的异常，Orange 的事件监督机制漏掉如此大量的事件是不现实的。因此，USAD 在 Orange 的生产环境中进行训练时不太可能遇到如此高的异常率。

最后，我们研究了灵敏度阈值（公式 9）的作用。较大的 α 表示在异常得分中更加重视 AE_1自动编码器的重构，而较大的 β 表示更加重视 AE_2自动编码器的重构（见图 1）。无需重新训练模型即可调整检测灵敏度，这对于 Orange 至关重要。表 4 报告了改变 α 和 β 对检测到的 FP、TP 数量以及 F1 得分的影响。

我们观察到，通过增加α并降低β，可以减少FP的数量（从0.0到0.9时最多减少50%），同时限制TP的数量下降（从0.0到0.9时减少3%）。因此，通过调节α和β可以参数化USAD的灵敏度，以满足生产环境的要求。通过模型，可以实现不同级别的灵敏度，从而使检测能够满足Orange监管团队中不同层级的需求。管理人员倾向于较低的灵敏度级别，以限制误报的数量，但在发生重要事件时发出警告，而技术人员则会更喜欢高水平的敏感度，这样他们就会错过最少的事件。

表 4：SWaT 数据集不同敏感度阈值的异常检测结果

α	β	FP	系列	卫生纸	F1
0.0	1.0	604	35,616		0.7875
0.1	0.9	580	35,529		0.7853
0.2	0.8	571	35,285		0.7833
0.5	0.5	548	34,590		0.7741
0.7	0.3	506	34,548		0.7738
0.9	0.1	299	34,028		0.7684

表 5：每个数据集上每次训练的时间（分钟）

方法	斯瓦特水	WADI	SMD	SMAP	MSL
OmniAnomaly 万能	13	31	87	48	11
美国田径协会	0.06	0.12	0.06	0.08	0.03
加速因子	216	258	1331	581	349

5.3 训练时间

在本节中，我们将研究 USAD 的计算性能，并将其与在异常检测方面性能最接近的方法 OmniAnomaly 进行比较（见表 3）。为此，我们测量了 5 个公开数据集上每个 epoch 的平均时间。SMD、SMAP 和 MSL 的参考时间是所有实体（即 28 台 SMD 机器、55 台 SMAP 机器和 27 台 MSL 机器）每个 epoch 的平均时间。两种方法均使用 NVIDIA GeForce GTX 1080 Ti 进行训练。

表5展示了得到的结果。USAD在多变量时间序列的无监督异常检测中表现良好，同时平均将训练时间缩短了547倍。

5.4 消融研究

使用 SMD、SMAP 和 MSL 数据集，我们研究了 USAD 两阶段训练的效果。图 3 展示了使用 USAD（组合训练）、仅使用第一阶段训练（自编码器训练）和仅使用第二阶段训练（对抗训练）的 USAD 的 F1 分数性能比较。不使用对抗学习的 USAD 训练表明使用了公式 4 中的目标函数，而抑制自编码器训练表明使用了公式 5-6 中的目标函数。

与排名第二的 USAD（未进行对抗训练）相比，受 GAN 启发的对抗训练的性能提升了 5.88%（F1 分数），与仅使用对抗训练相比，性能提升了 24.09%。这可以通过 USAD 引入的放大重建误差效应来解释，无论输入窗口中是否存在异常。因此，未进行对抗训练的 USAD 无法检测到最接近正常数据的异常。仅使用对抗训练的 USAD 性能不佳的原因是，该方法在开始对抗训练的第 2 阶段之前没有进行自动编码器训练来将权重调整到有利位置。在

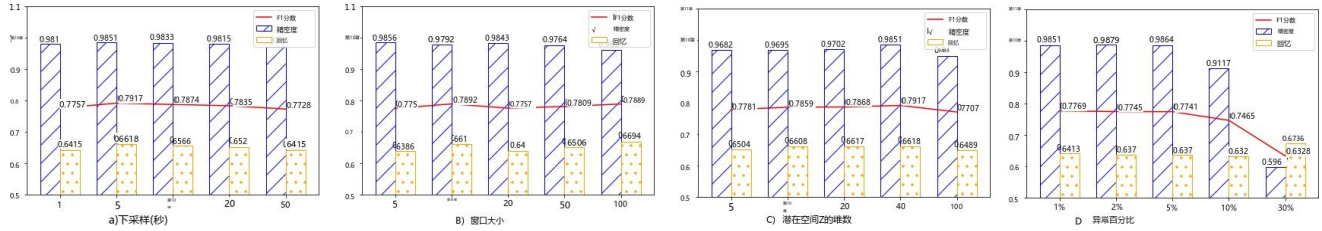

图 2：参数的影响。准确率、召回率和 F1 分数是 A）训练集下采样率、B）窗口大小 K、C）潜在空间维度 Z 和 D）训练集中异常百分比的函数。

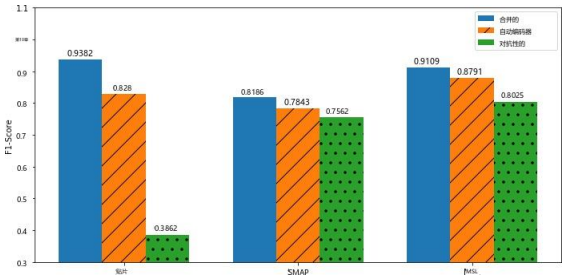

图 3：对抗训练与非对抗训练对 USAD 的影响

表 6：Orange 内部数据集上的异常检测结果（未进行点调整）

方法	精度	召回率	F1 分数
美国田径协会	0.7448	0.6428	0.6901

结论是，任何训练阶段的消融都会导致性能下降。例如，两个消融版本的 USAD 的 F1 得分都低于几个基准方法（表 2 底部）。

5.5 可行性研究

对复杂IT系统进行自动化监管对Orange来说是一项挑战。在研究了USAD的特性并评估其在使用公共数据集时的性能后，该公司必须确保该方法在其数据上同样有效。

表 6 报告了内部数据集中获得的结果。在两个月的测试数据中，USAD 能够在不到 30 分钟的时间内检测到所有重大事件。例如，USAD 能够在不到 30 分钟的时间内检测到 Orange 负责监管的操作员花了 24 小时才检测到的事件（图 4）。此事件是由于配置文件中引入了一个错误，该错误允许将广告展示分配给意外的合作伙伴。这导致广告展示数量（总展示次数）增加，同时平均展示价格（总平均展示价格）降低。结果，收入（总每千次展示费用/每点击费用/收入）等重要业务指标保持稳定，因此，操作员无法快速检测到事件。面对大量需要调查的指标，监管人员将精力集中在对业务影响较大的指标上，因此，需要 24 小时才能检测到此配置事件。

6 结论

在本文中，我们提出了 USAD，这是一种基于自动编码器的多变量时间序列无监督异常检测方法，其对抗训练受到生成对抗网络的启发。它的自动编码器架构使其成为一种无监督方法，并使其在对抗训练期间表现出极高的稳定性。我们使用了五个公共参考数据集来研究 USAD 的预期特性。在标准 F1 分数方面，该方法在公共参考数据集上的表现优于最先进的技术。此外，它表现出的快速训练、对参数选择的鲁棒性和稳定性使该模型在工业环境中具有高度的可扩展性。USAD 还提供了参数化其灵敏度的可能性，并可以从单个模型生成一组检测级别。这种可能性为 Orange 的监督团队提供了必要的功能，使他们能够在大规模基础设施的生产中使用该方法。由于团队需要能够在工作量过大时降低检测灵敏度，以预防重大事故，因此在推理过程中倍增检测灵敏度的能力使得该模型在公司内部具有极强的可扩展性，并带来了巨大的优势。首先，它使我们能够通过将监督模型的数量限制为一个来限制训练所需的时间。其次，投入生产的深度学习模型必须由团队进行监控和监督。限制模型数量可以减少我们在生产过程中监督模型所花费的时间，从而让监督人员有时间专注于其他任务。

使用Orange内部数据进行的可行性研究提供了确凿的结果，证实了USAD为Orange IT系统监管自动化指明了一个有前景的方向。研究也指出了部署和执行过程中可能遇到的一些困难。例如，在数据收集过程中（见第4.2节），我们遇到了意料之外的困难，即收集一个不包含太多异常的连续训练周期。这是一个值得关注的方面，它促使我们思考成功部署USAD需要哪些基础设施。

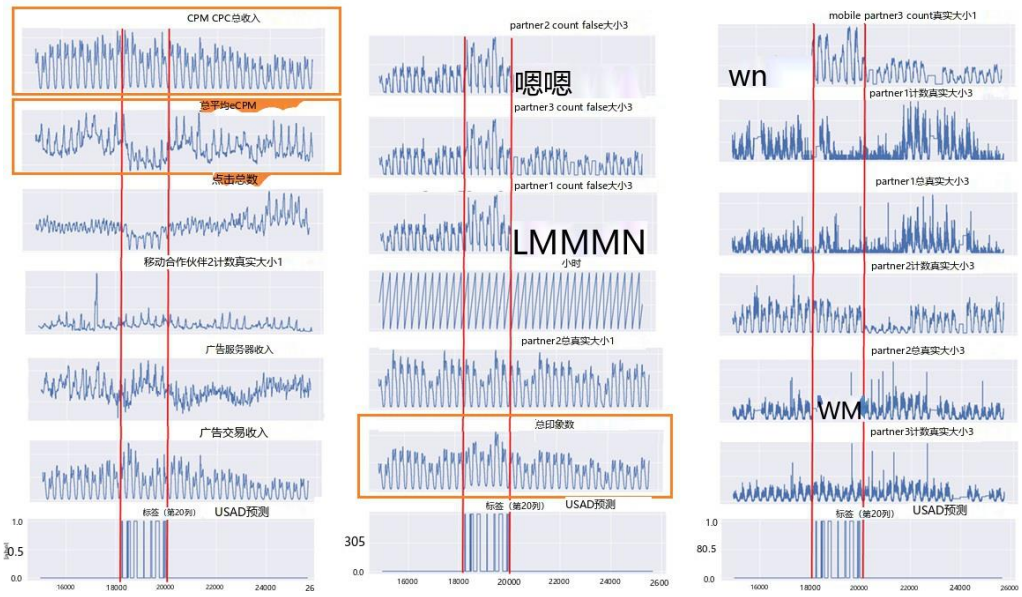

图 4：可行性研究中 USAD 检测到配置事件的时间序列示例。图中显示了 33 个时间变量中的 24 个。橙色框突出显示了所引用的变量。橙色框表示 5.5 节中引用的序列。

参考文献

[1] Martín Arjovsky 和 Léon Bottou。2017 年。《面向训练生成对抗网络的原则性方法》。第五届国际学习表征会议 ICLR 2017。OpenReview.net，法国土伦。

[2] Raghavendra Chalapathy 和 Sanjay Chawla。2019 年。《深度学习在异常检测中的应用：综述》。CoRR abs/1901.03407 (2019)。

[3] Wanpracha Art Chaovalitwongse、Ya-Ju Fan 和 Rajesh C. Sachdeo。2007 年。关于异常脑活动的时间序列 K 最近邻分类。*IEEE 系统、人与控制论汇刊 A 部分 37, 6 (2007), 1005–1016。*

[4] Jonathan Goh、Sridhar Adepu、Khurum Nazir Junejo 和 Aditya Mathur。2017 年。支持安全水处理系统设计研究的数据集。载于《关键信息基础设施安全》。88–99 页。

[5] Ian J. Goodfellow、Jean Pouget-Abadie、Mehdi Mirza、Bing Xu、David WardeFarley、Sherjil Ozair、Aaron C. Courville 和 Yoshua Bengio。2014 年。生成对抗网络。刊于《神经信息处理系统进展》第 27 届：2014 年神经信息处理系统年会。加拿大魁北克省蒙特利尔，邮编 2672–2680。

[6] Manish Gupta, Jing Gao, Charu C Aggarwal 和 Jiawei Han。2013 年。时间数据的异常值检测：一项综述。IEEE 知识与数据工程学报 26, 9 (2013)，2250–2267。

[7] 塞普·霍克赖特 (Sepp Hochreiter) 和于尔根·施米德胡贝尔 (Jürgen Schmidhuber)。 1997.长短期记忆。神经计算 9, 8 (1997), 1735–1780。

[8] Kyle Hundman、Valentino Constantinou、Christopher Laporte、Ian Colwell 和 Tom Söderström。2018 年。使用 LSTM 和非参数动态阈值检测航天器异常。第 24 届 ACM SIGKDD 国际知识发现与数据挖掘会议论文集（KDD 2018，英国伦敦，2018 年 8 月 19-23 日）。第 387–395 页。

[9] Istvan Kiss、Bela Genge、Piroska Haller 和 Gheorghe Sebestyen。2014 年。基于数据聚类的工业控制系统异常检测。载于 2014 年 IEEE 第十届智能计算机通信与处理国际会议论文集（ICCP 2014）。罗马尼亚克卢日-纳波卡，275–281 页。

[10] *Fei Tony Liu、Kai Ming Ting 和 Zhi-Hua Zhou。2008 年。《孤立森林》。载于第八届 IEEE 国际数据挖掘会议 (ICDM 2008) 论文集，2008 年 12 月 15-19 日，意大利比萨，第 413–422 页。*

[11] Junshui Ma 和 Simon Perkins。2003 年。基于单类支持向量机的时间序列新奇点检测。国际神经网络联合会议论文集 3 (2003)，1741–1745。

[12] Pankaj Malhotra、Anusha Ramakrishnan、Gaurangi Anand、Lovekesh Vig、Puneet Agarwal 和 Gautam Shroff。2016 年。基于 LSTM 的多传感器异常检测编码器-解码器。(2016)。arXiv.1607.00148

[13] Aditya P. Mathur 和 Nils Ole Tippenhauer。2016。SWaT：用于工业控制系统安全研究和培训的水处理测试平台。2016 年智能水网信息物理系统国际研讨会（CySWater）。31–36。

[14] Daehyung Park、Yuuna Hoshi 和 Charles C. Kemp。2018 年。基于 LSTM 的变分自编码器的机器人辅助喂食多模态异常检测器。IEEE 机器人与自动化快报 3, 3 (2018)，1544–1551。

[15] David E. Rumelhart、Geoffrey E. Hinton 和 Ronald J. Williams。1986 年。《通过误差传播学习内部表征》。麻省理工学院出版社，美国马萨诸塞州剑桥，第 318–362 页。

[16] Roy Schwartz、Jesse Dodge、Noah A. Smith 和 Oren Etzioni。2019 年。《绿色人工智能》。（2019 年 7 月）。arXiv.1907.10597

[17] 苏娅, 刘蓉, 赵有建, 孙伟, 牛晨昊, 裴丹。2019. 基于随机循环神经网络的多变量时间序列鲁棒异常检测。ACM SIGKDD 知识发现与数据挖掘国际会议论文集 1485 (2019), 2828–2837。

[18] 叶远、光绪勋、马凤龙、王亚庆、杜南、贾克斌、鲁肃、张爱东。2018.MuVAN：用于多元时态数据的多视图注意力网络。会议记录 - IEEE 国际数据挖掘会议，ICDM 2018-11 月 (2018)，717–726。

[19] Houssam Zenati、Manon Romain、Chuan-Sheng Foo、Bruno Lecouat 和 Vijay Chandrasekhar。2018 年。对抗学习异常检测。IEEE 国际数据挖掘会议 ICDM 2018。新加坡，727–736。

[20] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen 和 Nitesh V. Chawla。2019。用于多变量时间序列数据无监督异常检测与诊断的深度神经网络。AAAI 人工智能会议论文集 33 (2019)，1409–1416。

[21] 宗波、宋奇、民仁强、魏成、Cristian Lumezanu、Daeki Cho 和 Haifeng Chen。2018 年。深度自编码高斯混合模型在无监督异常检测中的应用。第六届国际学习表征会议 ICLR 2018。法国土伦，1-19 页。

A 可重复性的补充材料

A.1 实验设置

所有实验均在配备 Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz 和 270 GB RAM 的机器上进行，该机器位于运行 CentOS 7 版本 3.10.0 的 docker 容器中，并可访问 NVIDIA GeForce GTX 1080 Ti 11GBGDDR5X GPU。孤立森林 (IF) 来自 scikit-learn[4]实现。DAGMM 来自 Github 上的 Tensorflow 实现[5]。LSTM-VAE 来自 Github 实现[6]。Om-niAnomaly 来自作者的 Github Tensorflow 实现[7]。最后，我们在 Pytorch 中开发了 USAD 和 AE。

A.2 我们实现中使用的包

我们的算法实现中使用的相关软件包及其版本如下：

- python==3.6.8
- pytorch==1.3.1
- cuda==10.0
- scikit学习==0.20.2
- numpy==1.15.4

A.3 每个数据集的 USAD 超参数

A.4 对于每个数据集，我们有 4 个参数。窗口的大小，对应于输入的时间序列的大小。epoch 的数量，Z 的维度（即 USAD 潜在空间），以及预处理过程中的下采样率。下采样是通过取每个特征的中值来完成的。

表 7：每个数据集的 USAD 超参数。K 表示窗口大小，m 表示潜在空间的维度。

数据集	K 个时期		m	下采样
扑打	12	70	40	5
瓦迪	10	70	100	5
贴片	5	250	38	5
斯玛特	5	250	55	5
语言学硕士	5	250	33	5

A.5 USAD 实施

输入大小对应于窗口大小乘以多元时间序列的维数。

A.5.1 编码器。

- 线性：输入大小 -> 输入大小 / 2
- 热鲁
- 线性：输入大小/2 -> 输入大小/4
- 热鲁
- 线性：输入大小 /4 -> 潜在空间大小
- 热鲁

A.5.2 解码器。两个解码器具有相同的架构。

- 线性：潜在空间大小 -> 输入大小 / 4
- 热鲁
- 线性：输入大小/4 -> 输入大小/2
- 热鲁
- 线性：输入大小/4->输入大小
- S形函数

作为优化器，我们使用 Adam 的 pytorch 实现及其默认学习率。

[4] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html

[5] https://github.com/tnakae/DAGMM

[6] https://github.com/Danyleb/Variational-Lstm-Autoencoder

[7] https://github.com/NetManAIOps/OmniAnomaly