DOI:10.11918/202212029

基于上下文生成对抗网络的时间序列异常检测方法

胡智超,余翔湛,刘立坤,张 宇,于海宁

(哈尔滨工业大学 网络空间安全学院,哈尔滨 150001)

摘 要:时间序列的异常检测是网络服务保障、数据安全检测、系统监控分析等应用中所依赖的一项关键技术。为解决在实际场景的时间序列异常检测中由于时间序列上下文的模糊性、数据分布的复杂性以及异常检测模型的不确定性所带来的异常检测结果的有效性、合理性、稳定性等不足的问题,本文提出了一种新的基于上下文生成对抗网络的时间序列异常检测方法 AdcGAN。首先,通过处理历史数据,提取用于生成时序数据的条件上下文;然后,采用条件生成对抗网络的设计策略,使用条件上下文,构建上下文生成对抗网络,实现对任意时刻数据的条件分布预测,同时 AdcGAN 采用 Dropout 近似模型不确定性,使用概率分布代替点估计作为预测结果;接着,从观测的差异(用期望偏差表示)和模型的不确定性(用预测方差表示)两个方面来衡量异常;最后,提出基于数据统计信息的异常阈值自动设置方法,减少手动调节的参数量。实验结果表明,与同类基准算法进行对比,在 NAB 数据集中的 47 个真实时序数据上,本文提出的 AdcGAN 可以有效地检测出时序数据中的异常,在大多数评价指标上都优于其他基准方法,并且具有更好的稳定性。

关键词:时间序列异常检测;生成对抗网络;模型不确定性;生成模型;深度学习

中图分类号:TP309.2 **文献标志码**:A **文章编号**:0367-6234(2024)05-0001-11

A time series anomaly detection method based on contextual generative adversarial network

HU Zhichao, YU Xiangzhan, LIU Likun, ZHANG Yu, YU Haining

(School of Cyberspace Science, Harbin Institute of Technology, Harbin 150001, China)

Abstract: Time series anomaly detection is a key technology relied upon in applications such as network service, data security, and system monitoring. In order to address the limitations of effectiveness, rationality and stability in anomaly detection results caused by the fuzziness of time series context, complexity of data distribution, and the uncertainty of anomaly detection models in practical scenarios, this paper proposes a new anomaly detection method (AdcGAN), based on contextual generative adversarial network. Firstly, AdcGAN extracts conditional context for generating time series data by processing historical data. Secondly, AdcGAN constructs a context generative adversarial network following conditional generative adversarial network strategy to achieve conditional distribution prediction of the data at any moment. Meanwhile, AdcGAN uses Dropout to approximate model uncertainty and replacing point estimates with probability distribution as prediction result. Then, anomalies are measured based on the differences in observations (represented by the expected deviations) and the uncertainty of the model (represented by prediction variances). Finally, an automatic method for setting anomaly thresholds based on statistical information of the time series data is proposed to reduce the number of manually adjusted parameters. Our experimental results on 47 real-time series data of the NAB dataset compared with baselines show that, compared to similar benchmark algorithms, the proposed AdcGAN method can effectively detect anomalies in time series data. It outperforms other benchmark methods in most evaluation metrics and achieves better stability.

Keywords: time series anomaly detection; GAN; model uncertainty; generative model; deep learning

随着互联网技术的快速发展,社会生活的各个领域都逐步进入开放化、智能化、便利化的数字生态阶段。海量的时序数据不断的产生,从时间维度上反映了不同场景的状态变化,在数据监控分析中,该变化与服务稳定性、数据安全性、系统可靠性等密切相关。时间序列异常检测可以及时地检测出时序数据中的异常变化,从而发现异常事件,并通过及时地处置与修复,有效地减少损失与影响。因此是网络

收稿日期:2022-12-09;**录用日期**:2023-03-13;**网络首发日期**:2023-11-06
网络首发地址:http://kns.cnki.net/kcms/detail/23.1235.T.20231103.1141.004.html
基金项目:国家重点研发计划(2018YFB1800702)
作者简介:胡智超(1990—),男,博士研究生;余翔湛(1973—),男,教授,博士生导师
通信作者:余翔湛,yxz@hit.edu.cn

服务保障、数据安全检测、系统监控分析等应用中所依赖的一项关键技术,已经成为当前国内外时间序列研究中的热门课题[1-4]。

由于异常标签的缺乏,时间序列的异常检测通常被视为一个无监督的机器学习任务[5-6]。传统的方法主要有基于统计模型(如 ARIMA、GARCH)[7-8]、基于聚类(如 K-Means、EM、SVM)[9-12]、基于相似性度量[13]、基于降维(如 PCA)[14]等方法。然而随着时序数据复杂性的增加,传统方法逐渐难以满足异常检测的性能要求。所以,越来越多的深度学习方法被应用到时间序列的异常检测研究中,包括:基于预测的方法(如 LSTM、MLP)[15-16]、基于重构的方法(如全连接自动编码器 Dense AE、稀疏自动编码器 Sparse AE)[17-20]等。这些方法着重从不同的角度量化了正常数据与异常数据之间的差异,被广泛使用。由于生成对抗网络(generative adversarial network,GAN)在数据生成类任务中取得了十分不错的效果[21-22],部分研究人员引入 GAN 来提升正常数据预测或重构的有效性,从而提高异常检测效果,比如:TadGAN[23]、MAD-GAN[24]、TAnoGAN[25]等。还有一些研究人员探索了如何通过结合其他学习方式来提升异常检测效果,其中以迁移学习为主。基于迁移学习的异常检测方法[26]使用数据迁移和特征迁移来增强异常检测效果。在提高时序建模效果的同时,研究者还探索了多维时间序列的异常检测方法,通过对多维时间序列间的关系分析增强时序异常检测的效果[27-28]。

然而,时间序列的异常检测是特别具有挑战性的,原因有二:1)在复杂的时间序列上,数据在时间维度上有着模糊的时间上下文以及复杂的数据分布,现有的异常检测方法并不能很好捕捉这种时间上下文以及对时间序列的真实分布进行建模,因此在异常检测的性能上还存在较大的提升空间;2)大多数的深度学习方法通过预测或者重构损失来衡量异常程度,这些方法仅考虑观测数据和预测数据的差异性,并没有考虑到模型本身引入的不确定性,因此异常检测结果的合理性和稳定性还需要提升。

本文提出了一种基于上下文生成对抗网络的异常检测方法(anomaly detection based on context generative adversarial network, AdcGAN),用于复杂时间序列数据的异常检测,见图 1。

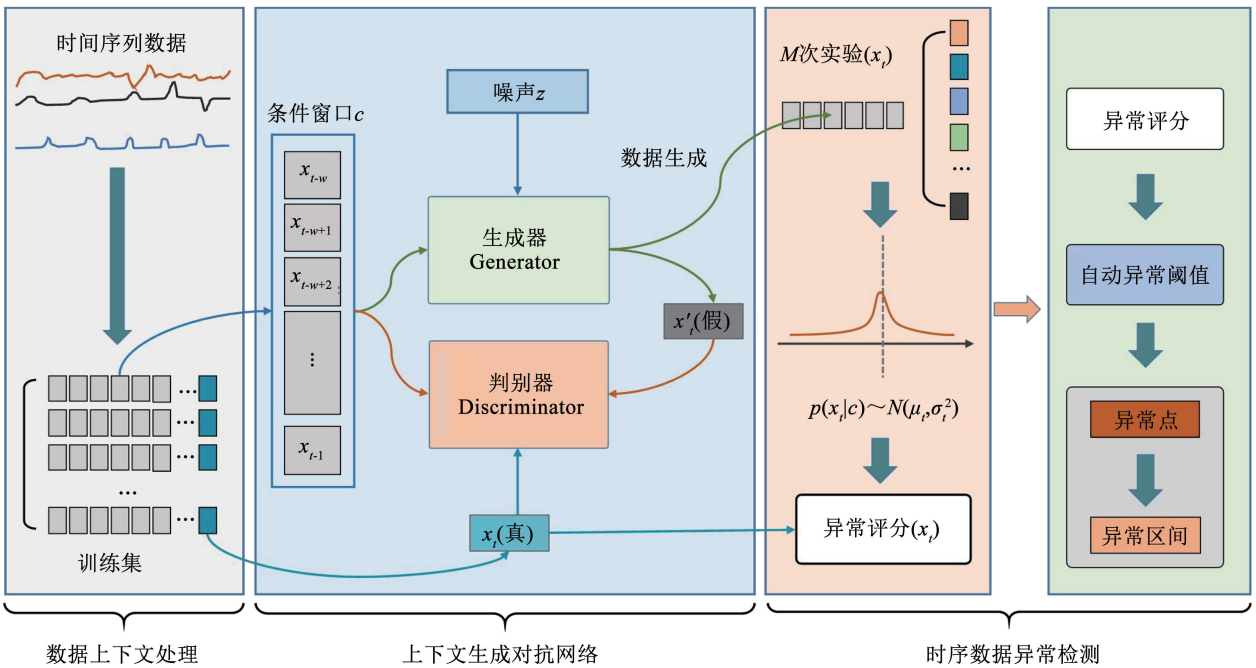


图 1 AdcGAN 框架的概述

Fig. 1 Overview of AdcGAN framework

针对以上问题,本文提出的网络设计如下:首先,通过数据上下文处理,将时序数据转换成条件上下文的形式。然后,提出了上下文生成对抗网络,通过扩展 GAN 学习指定上下文条件下的时间序列数据分布,为了减少深度学习模型不确定性带来的影响,该方法增加 Dropout 网络层对其进行模拟,从而学习时序数据的条件生成分布。以输出的概率分布代替点估计,增加了模型预测的可信度和鲁棒性。最后,提出基于数据统计信息的自动阈值方法为不同时间序列设置异常阈值,避免了过多地手动调参。在真实的监控数据集上测试,实验结果表明该方案对于时间序列具有良好的异常检测效果,并且增强

了异常检测的鲁棒性。

1　数据上下文处理

记 $T=\{t_1,t_2,\cdots,t_N\}$ 是有序的时刻集合，长度为 N，X 是目标随机变量，表示在时刻集合 T 上观测得到的时序数据，记为

$$X_n=\{x_1,x_2,\cdots,x_n,\cdots\}\quad 1\leqslant n\leqslant N \tag{1}$$

该序列的长度为 N。对于任何时刻 t，其真实值与之前的 $t-1$ 时刻有关。时间序列上的异常检测问题，要求学习这样一个函数 $f: X\rightarrow Y$，其中 $Y\in\{0,1\}$。输入是时间序列 X_n，输出是序列中每个项目的异常标签。$Y_t=1$ 表示时刻 t 是异常的，否则 t 是正常的。此外，函数 $g: Y\rightarrow W$，以异常检测结果作为输入，并根据异常窗口大小找到异常区间。如果区间内的所有时刻都被标记为异常，那么该时间区间就是异常的。

通常，在时序的处理中，将对过去的依赖简化为过去一段时间来简化问题，记 w 是简化过去时刻依赖性的固定窗口大小。则目标函数为

$$f(x_t)\Leftrightarrow f(x_t|x_1,x_2,\cdots,x_{t-1})\simeq f(x_t|x_{t-w},x_{t-w+1},\cdots,x_{t-1}) \tag{2}$$

对于一个时间序列中的任何时刻 t，它都依赖于之前的 $t-1$ 时刻。这种无限长的依赖关系给序列建模带来了困难。在本文中，采用序列数据建模中的一般处理方法，使用固定的时间窗口 w 来约束依赖关系。约束 w 表示用过去的 w 时刻代替对以前 $t-1$ 时刻的依赖。称 $x_{t-w}\sim x_{t-1}$ 是 x_t 的预测上下文，即生成条件，表示为 c。在训练和检测阶段，第一步是提取序列数据的上下文条件。本文使用过去的 w 窗口数据作为时刻 t 的生成条件，在时间序列上有唯一的观测值对应。此外，为了捕捉序列的长期趋势并减少平滑序列中的噪声影响，在时序处理时使用步长，根据步长对序列进行间隔处理。步长一般取值大于 1，在 AdcGAN 中设置步长为 3。

通过上述对数据上下文的处理，为每一个时刻提取依赖上下文作为数据的生成条件。上下文由过去 w 个时刻决定。这样，将时间序列数据转化为“条件与观测值”的组合对，用于模型的训练和异常检测。

2　上下文生成对抗网络

本节通过生成对抗网络 GAN 学习时间序列数据的分布，然后结合 Dropout 模拟模型不确定性，从而获得每个时刻数据的条件分布。

2.1　基于 GAN 的数据分布学习

生成模型是学习真实数据分布的一种有效方法。GAN 是训练生成模型的常用方法之一。它由 2 个对抗性模型组成：生成器模型 G 和判别器模型 D。生成器 G 捕捉真实数据分布，判别器 D 估计样本来自训练数据（真实数据）而不是 G（假数据）的概率。

生成器的目标是学习一个分布，从该分布中生成的数据可以迷惑判别器。为了在数据 x 上学习一个生成器分布 p_g，生成器建立了一个从先验噪声分布 $p_z(z)$ 到数据空间 $G(z;\theta_g)$ 的映射函数。判别器的目标是最大限度地区分真实数据和虚假数据。判别器 $D(x;\theta_d)$ 输出一个标量，代表 x 是真的概率。当 $x\sim p_{\text{data}}(x)$ 时，它期望输出 1；当 $x\sim p_g(x)$ 时，它期望输出 0。

G 和 D 是同时训练的，目标是调整 G 的参数以最小化 $\log(1-D(G(z)))$，调整 D 的参数以最大化 $\log(D(x))$

$$\mathscr{L}=\min_G\max_D V(D,G) \tag{3}$$

损失函数 $\mathscr{L}$ 等于

$$\mathscr{L}=\boldsymbol{E}_{x\sim p_{\text{data}}(x)}[\log D(x)]+\boldsymbol{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{4}$$

GAN 对 G 和 D 的实现没有要求，它可以是一个任意的非线性映射函数，如多层感知机。这为扩展 GAN 提供了的灵活性。对于时间序列的预测，需要在给定的条件下产生预测值。因为原始的 GAN 过于自由，训练会很容易失去方向，从而导致不稳定又效果差。而条件 GAN 就是在原来的 GAN 模型中加入一些先验条件，使得 GAN 变得更加的可控制。具体来说，可以在生成模型 G 和判别模型 D 中同时加入条件约束来引导数据的生成过程。条件可以是任何补充的信息，如类标签、其他模态的数据等。然后这样的做法应用也很多，比如图像标注、利用文本生成图片等等。

本文提出了上下文 GAN，它是一种设计用于时间序列的条件 GAN[29]。在上下文 GAN 中，生成器和判别器都以一些额外的上下文信息作为条件。使用预测上下文 c 作为条件，通过输入 c 作为额外的输入条件。AdcGAN 中生成器和判别器的结构见图 2。

其中，图 2 左侧是条件生成器 G，它使用随机噪声样本生成符合给定条件的数据；图 2 右侧是条件判别器 D，它判断数据是否来自真实数据分布并输出概率。使用向量连接操作来混合上下文和噪声、生成条件和观测值。生成条件 c 的大小取决于条件窗口大小的超参数 w。AdcGAN 中的 2 个条件对抗模型：生成器 G 和判别器 D。具体说明如下。

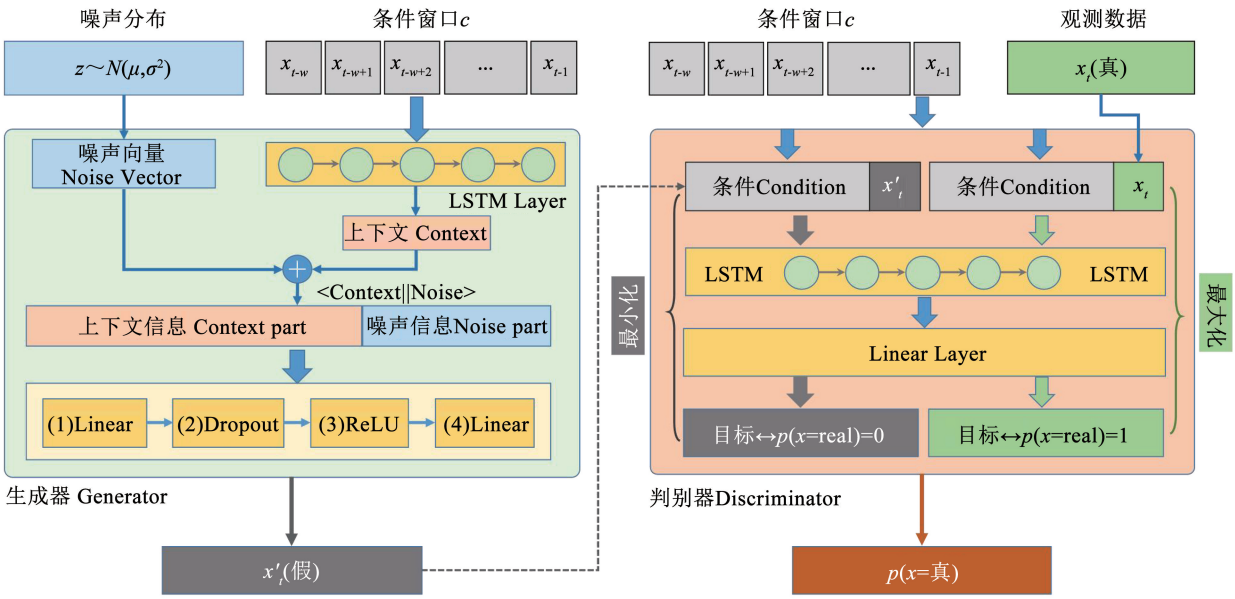


图 2　生成器和判别器的结构

Fig. 2　Architecture of generator and discriminator

1)生成器:由 LSTM 和包含 Dropout 的全连接层构成。它使用一个序列数据作为条件,并使用 LSTM 提取得到上下文向量 $\boldsymbol{c}'$。噪声 z 采用高斯分布,即 $z\sim N(\mu,\sigma^2)$。从噪声分布中随机采样得到噪声向量,通过用 c' 整合噪声向量 z 形成新的输入。生成器的输出是由多个全连接层产生的,被称为假数据。优化时,生成器需要使得生成的数据可以迷惑判别器,最大化将生成数据判别为真实数据的概率。

2)判别器:由 LSTM 和全连接层构成。输入为给定条件下的观测值:(a)如果观测值是生成器产生的,即 $x'_{t+1}=G(z|c)$,则对应的分类标签为 0。判别器的期望输出为 0,即 $p(x=\text{real})=0$,优化时判别器需要最小化这个概率;(b)如果观测值来源于时间序列的观测,对应的分类标签为 1,此时判别器的期望输出为 1,即 $p(x=\text{real})=1$,优化时判别器需要最大化这个概率。

对抗性训练框架允许在如何组成上下文隐向量的设计上有相当大的灵活性。在生成器 G 中,将先验输入噪声 $p_z(z)$ 和 c 合并为联合隐向量表示。而在判别器 G 中,使用 x 和 c 作为输入和判别函数。相比 GAN,AdcGAN 的目标函数将被更新为

$$\mathscr{L}=\boldsymbol{E}_{x\sim p_{\text{data}}(x)}[\log D(x|c)]+\boldsymbol{E}_{z\sim p_z(z)}[\log(1-D(G(z|c)))] \tag{5}$$

在 AdcGAN 的训练阶段,输入是上下文条件和对应生成数据的组合 C 和 X,批次大小 m,每次迭代判别器的训练次数 k。输出是生成器 G 和判别器 D。训练过程中,生成器 G 和判别器 D 以对抗的方式训练,直至收敛。在每个循环迭代中,先训练 D,然后再训练 G。具体算法实现流程如下:

Step1　初始化生成器模型 G 和判别器模型 D;

Step2　执行迭代训练:在迭代次数内,重复执行 Step3 ~ Step9,进行生成器和判别器的训练;

Step3　首先执行判别器的训练:在训练次数 k 次内,重复执行 Step4 ~ Step7,不断更新判别器;

Step4　从噪声先验分布 $p_g(z)$ 中采样 m 个噪声样本 $\{z_1,\cdots,z_m\}$;

Step5　从数据生成分布 $p_{\text{data}}(c,x)$ 中采样 m 个样本 $\{(c_1,x_1),\cdots,(c_m,x_m)\}$;

Step6　使用 m 个噪声样本生成假数据 $\{G(z_1|c_1),\cdots,G(z_m|c_m)\}$;

Step7　更新判别器:$\nabla\theta_d\frac{1}{m}\sum_{i=1}^{m}[\log D(x_i|c_i)+\log(1-D(G(z_i|c_i)))]$;

Step8　然后执行生成器的训练:从噪声先验分布 $p_g(z)$ 中重新采样 m 个噪声样本 $\{z_1,\cdots,z_m\}$;

Step9　更新生成器:$\nabla\theta_d\frac{1}{m}\sum_{i=1}^{m}\log(1-D(G(z_i|c_i)))$;

Step10　完成迭代训练后,返回 G,D。

对于任意时刻 t,过去 w 个时刻组成了生成条件 c,是生成器和判别器的主要输入。通过噪声分布 z 的随机采样,生成器 G 可以生成假数据 x'_t。真实的观测值 x_t 为判别器 D 提供了正例样本,与生成器产生的负样本一起构成了 GAN 的训练集。通过生成器和判别器的不断对抗训练,最终生成器学习到了时间序列数据的近似生成分布。

2.2　考虑不确定性的条件分布

预测模型的不确定性是一个需要解决的重要问题。通常，贝叶斯概率理论为我们提供了有数学基础的工具来推理模型的不确定性，但这些工具通常会有较高的计算成本，并对非贝叶斯模型产生较大的破坏性。研究表明，在不改变模型或优化的情况下，可以将深度学习工具转化为贝叶斯模型[30]。在神经网络中使用 Dropout（及其变体）可以被解释为概率模型的贝叶斯近似：高斯过程（Gaussian process，GP）。因此，可以在不增加计算复杂性或减低测试准确性的情况下，在深度学习模型中表示不确定性。

在预测分布时，生成器网络 G 的 Dropout 将保持开启，通过重复 M 次实验，对给定条件的预测结果进行采样。通过计算样本均值和方差对整体均值和方差进行估计，其中时间 t 的均值为

$$\mu_t \simeq \frac{1}{M}\sum_{i=1}^{M} x_i = \frac{1}{M}\sum_{i=1}^{M} G(z_i \mid c_i) \tag{6}$$

基于平均值，时间 t 的方差可以通过以下方式估计：

$$\sigma_t^2 \simeq \frac{1}{M}\sum_{i=1}^{M}\left(G(z_i \mid c_i) - \frac{1}{M}\sum_{j=1}^{M} G(z_j \mid c_j)\right)^2 \tag{7}$$

时间 t 的预测结果是一个以 c_t 为已知条件的条件概率分布，可以认为它近似地服从高斯分布，描述为 $p(x_t \mid c_t) \sim N(\mu_t, \sigma_t^2)$。

3　时序异常检测

一个数据集的真实数据分布是可学习的基于这样的前提假设：用于训练的时间序列中无异常点，都是正常数据，或者异常的数量在一定的比例之下，并没有对正常的分布产生过大的改变。这种情况下，模型可以学习到有效的数据分布，并可以通过阈值来区分异常点和非异常点。因此，在基于预测的异常检测方案中，异常大小取决于 2 个因素：一是观测值与估计值的偏离程度，二是预测模型对真实值估计的不确定性。因此在进行时序异常检测时，首先进行时序数据的条件数据分布预测，然后进行异常评分，再输出异常结果。

3.1　时序数据的分布预测

一个关于时间序列的概率预测例子如图 3 所示。对于每个时刻 t，下限和上限被设定为 $\mu_t \pm 3^* \sigma_t$，中间的下限和中间的上限被设置为：$\mu_t \pm 2^* \sigma_t$。其中：

1）图 3（a）中，时刻 1950 ~ 2150 之间的正常区域 A，最大的 σ 是 0.119 9；

2）图 3（b）中，时刻 3600 ~ 3800 之间的警告区域 B，最大的 σ 是 0.410 5；

3）图 3（c）中，时刻 4950 ~ 5150 之间是警告区域 C，最大的 σ 是 0.296 3。

4）图 3（d）中，时间序列的平均 σ 为 0.108 1。

从图中可以明显看出，对于区域 B 和 C 这 2 个警告区域，很有可能是异常点所在的区间，因为其中的数据具有明显的上升和下降变化。而且，警告区域的方差和波动边界都明显高于正常区 A。方差和边界说明了模型对该区域进行预测时的不确定性，同时也反映了这一段时间区域的分布与整个时间序列的真实分布之间的差异。

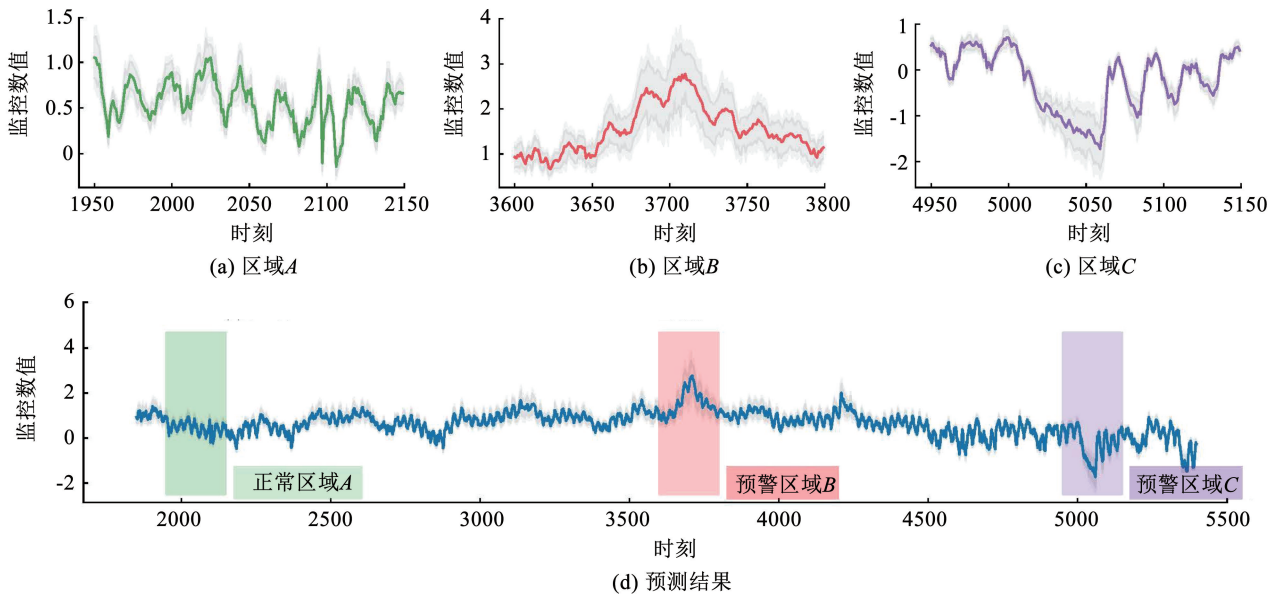


图 3　AdcGAN 对时间序列进行概率预测

Fig. 3　Probabilistic prediction of AdcGAN on time series

在给定的生成条件下,使用生成器估计任意时刻的真实值分布。在异常评分阶段,生成器 G 将运行 M 次,从数据生成分布中采样。因此,可以根据这些样本计算出 $p(x_t|c)$。

3.2 基于预测的异常检测

通过评价观测值与真实值的估计之间的差异程度以及模型对预测结果的不确定程度对时刻 t 的异常进行打分。对于任意的时刻 t,根据公式和可以计算得到它的预测分布,因此它的观测值与真实值的偏离程度可以表示为期望 $E(x_t - X_t)$。t 时刻的异常得分定义如下:

$$A_t = \sigma_t^* \frac{|E(x_t - X_t)|}{\boldsymbol{D}} = \sigma_t^* \frac{|x_t - \mu_t|}{\boldsymbol{D}} \tag{8}$$

其中,A 是所有时间序列的异常得分,σ 是预测方差,$\boldsymbol{D}$ 是从训练数据中得到的时间序列的值边界。对时序进行异常检测时,输入是上下文条件与数据组合 $\boldsymbol{C}$,$\boldsymbol{X}$,以及数据生成器 G。输出是异常得分列表 $\boldsymbol{A}$。具体算法实现流程如下:

Step1 记待检测时间序列的长度 n,使用 $\boldsymbol{C}$ 的长度进行初始化;

Step2 初始化异常得分列表 $\boldsymbol{A}$,所有时刻异常初始得分为0;

Step3 对于时间序列中的每个时刻 i,重复执行 Step4 ~ Step10 更新其异常得分;

Step4 获取第 i 项的数据及其生成上下文:$(c,x) \leftarrow (\boldsymbol{C}[i],\boldsymbol{X}[i])$;

Step5 从噪声分布 $p_z(z)$ 中采样 z;

Step6 创建空的生成结果集合:$\boldsymbol{x}^g \leftarrow \varnothing$;

Step7 重复执行 M 次生成器得到给定条件下的生成分布:$\boldsymbol{x}^g \leftarrow \boldsymbol{x}^g \cup \boldsymbol{G}(z|x)$

Step8 针对生成结果集合,根据公式(6)计算 $\boldsymbol{x}^g$ 的预测均值 μ;

Step9 针对生成结果集合,根据公式(7)计算 $\boldsymbol{x}^g$ 的预测方差 σ;

Step10 计算第 i 项的异常得分:$\boldsymbol{A}_i \leftarrow \sigma^* |x - \mu| / \boldsymbol{D}$;

Step11 完成所有时刻的异常评分,返回 $\boldsymbol{A}$。

记 $\boldsymbol{A}^p$ 作为时间序列 X_n 的异常标签。考虑到异常阈值 τ,如果 $\boldsymbol{A}_t \geqslant \tau$,那么 $\boldsymbol{A}_t^p = 1$ 表示时间 t 是一个异常值。通常情况下,τ 取决于专家的经验或模型的性能。本文设计了一个自动阈值方法:记 μ 是 $\boldsymbol{A}$ 的平均值,σ 是 $\boldsymbol{A}$ 的标准差,那么阈值可以是 $\tau_{\text{std}} = \mu + 2^*\sigma$ 或者 $\tau_p = \text{percentile}(\boldsymbol{A}, 99.9)$,见式(9):

$$\tau = \begin{cases} \tau_p & \text{if} \quad \tau_{\text{std}} > \max(\boldsymbol{A}) \\ \max(\tau_p, \tau_{\text{std}}) & \text{else} \end{cases} \tag{9}$$

给定异常窗口大小 W,如果一个时间点在异常点扩展的时间跨度内,则该时间点被视为异常点[31]。这种处理鼓励异常的早期发现,以及惩罚异常的延时发现。通过异常点的评分和异常窗口策略,AdcGAN 最终输出所有的异常点和异常区间检测结果。

4 实验结果

使用本文提出的 AdcGAN 方法,针对多样化的时序数据进行异常检测,通过与基准方法的对比分析,验证了所提方法的有效性。

4.1 数据集说明

本文在 NAB 数据集[31]上进行实验评估。NAB 由一系列真实采集和人工合成的时间序列组成,带有异常点和异常区间的标注。本文选择了所有的真实类别数据用于实验,一共5种类别,共计47个数据序列,321 206 个数据点和 31 077 个异常点。真实数据来源于多种实际场景:

1)云监控(AWSCloudWatch):AWS 服务器的 CPU 占用率、网络流量以及硬盘读写等指标;

2)广告(AdExchange):点击指标包括每千次展示费用和每千次点击费用;

3)多源(KnownCause):多种已知异常原因的时序数据,包括云主机监控、工业传感器和出租车搭乘数据等;

4)交通量(Traffic):由特定传感器上报的道路交通信息的实时监控,包括道路占用率、车辆速度和旅行时间等;

5)推特(Tweets):每隔 5 min 统计的大型上市公司话题次数。

每个数据序列文件包含有 2 个字段:时间戳和监控数值,监控数值均是一维数值。一组数据样例见表1,数据来源于亚马逊云主机的监控,监控数值为主机的 CPU 占用率。

表1 NAB 时序数据样例(ec2_cpu_utilization_5f5533)

Tab.1 Example of NAB time series data(ec2_cpu_utilization_5f5533)

基本参数信息	数值
2014-02-14 14:27:00	51.846
2014-02-14 14:32:00	44.508
2014-02-14 14:37:00	41.244
2014-02-14 14:42:00	48.568
2014-02-14 14:47:00	46.714
2014-02-14 14:52:00	44.986
2014-02-14 14:57:00	49.108

每个序列的数据文件在异常标签文件中对应有异常的记录,该字段记录了该数据序列中所有异常

区间的起始和终止时间戳。本文以无监督的方式训练所有模型，仅在评估检测性能时使用异常标签信息。

数据集中不同类型数据的基本信息汇总见表 2。对于每个数据集，为了了解异常识别的难易程度以及数据集统计特征对异常识别结果的影响，分别统计了超过 2 个标准差及 3 个标准差的数据所占比例，分别记为 2σ 和 3σ。

表 2　数据集 NAB 的统计信息

Tab. 2　Summary of each dataset in NAB

数据来源	数据流	2σ 总数	2σ 占比/%	3σ 总数	3σ 占比/%	数据点	异常区间	异常点	异常占比/%
云监控	6	244	2.53	71	0.73	9 610	14	960	9.98
广告	17	2 772	3.05	785	1.15	67 740	30	6 312	9.31
多源	7	1 396	2.00	765	1.09	69 561	19	6 594	9.47
交通量	7	351	2.24	141	0.90	15 664	14	1 560	9.95
推特	10	3 846	2.42	1 748	1.10	158 631	33	15 651	9.86
总计	47	7 909	2.46	3 510	1.09	321 206	110	31 077	9.67

4.2　实验设置

本文选择了 9 种方法作为基准进行比较：LOF[32]、OCSVM[33]、IF[34]、RRCF[35]、MLP[15]、LSTM[36]、DenseAE[19]、TAnoGAN[25]和 GMM[37]。本节的实验环境是基于 Python 3.9.7、pytorch 1.8.0 和 cuda 11.1 构建。对于每种检测方法，有以下训练设置：

1）使用时间序列数据的 80% 的作为训练数据集，另外 20% 作为验证数据集。

2）时间序列的依赖窗口大小设定为 $w=19$，表示使用过去 19 个时刻预测未来时刻。对于使用小窗口作为输入的检测方法，窗口大小为 $w+1=20$。

3）为了将输入序列转换成适当的格式：向量或条件观测对，在训练阶段将步长设置为 3，在检测阶段设置为 1。

4.3　AdcGAN 的异常检测结果

为了说明 AdcGAN 的检测结果，对一个有 3 个异常区间的时间序列进行了异常检测，其结果见图 4。待检测时间序列源自亚马逊数据中心的一台服务器的监控数据，该数据持续记录 CPU 使用情况，直至完全系统故障而结束，故障由 AWS API 服务器的记录故障导致的。共有 3 个已知的异常区间。

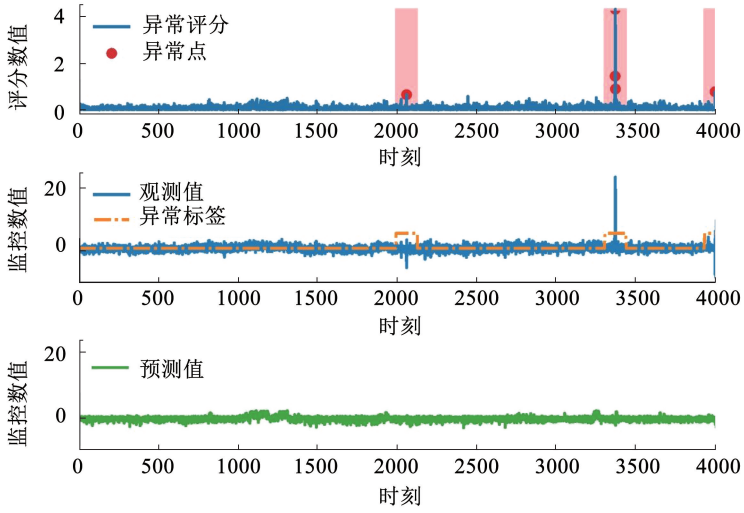


图 4　AdcGAN 异常检测样例

Fig. 4　An example of anomaly detection using AdcGAN

通过将预测的均值与实际时间序列数据进行比较，共有 5 个检测到的异常点和 3 个异常区间，与 3 个真实的异常区间相对应。警告点的异常得分接近于阈值 τ，也值得关注。

4.4　性能比较

使用 4 个标准评价指标 F1-Score（F_1）、召回率 Recall（R）、精确度 Precision（P）、准确度 Accuracy（A_{cc}）作为异常检测的评价指标。与基准方法的比较结果见表 3 和图 5，其中平均值 MEAN 代表了每种方法在整个 NAB 数据集上的平均表现。

由于异常检测的特殊性，数据中正常类别的比例要远远高于异常类别。然而，由于异常的重要性，

异常数据的识别比正常数据具有更高的价值,所以 F_1、精确度和召回率通常比准确率更重要。从异常检测的结果中,可以看出:

1)整体性能方面:在整个数据集上,AdcGAN 在大多数评价指标上优于其他方法。它取得了最高的 F_1(0.662)、召回率(0.814)和精确度(0.620),与第二优的方法相比,分别提高了 12.4% 的 F_1、8.4% 的召回率和 4.3% 的精确度。所有方法的准确率都很接近,GMM 是最优的(0.889),AdcGAN 也取得了接近的准确率(0.878)。

2)稳定性方面:对于每个类别,AdcGAN 都是性能最优的或者接近最优。在 5 个类别的数据集上,以 F_1 为例,取得了 4 项第一,1 项第二。同时,其 F_1 波动是最小的,说明了 AdcGAN 取得较好效果的同时,也保持了很好的稳定性。

3)时间上下文处理方面:同为基于预测的方法,由于检测对象是时间序列,LSTM 取得了比 MLP 更好的结果,说明对时间上下文的处理可以有效提升时间序列的建模效果。而在 AdcGAN 的上下文生成对抗网络中,同时使用了 LSTM 和 MLP 作为基础组件处理序列数据和模型输出,分别提升了 12.4% 和 14.1% 的 F_1,说明了上下文生成对抗网络的设计对于时间序列数据分布的学习是有效的。

4)与重构模型对比:AdcGAN 与 DenseAE 都是生成式模型,前者利用数据生成进行预测,后者利用数据生成来评估重构损失。从结果上看,使用条件 GAN 的预测数据分布并捕捉模型不确定性的 AdcGAN,相较于 DenseAE 在各项指标上表现更优,其中 F_1 提升 0.207,召回率提升 0.356,精准度提升 0.07,准确率提升 0.01,说明了在本文的场景下 GAN 的数据生成方法是更优的。

表 3　AdcGAN 与基准方法的比较

Tab. 3　Comparison of AdcGAN and baselines

数据集	指标	AdcGAN	DenseAE	GMM	IF	RRCF	LOF	LSTM	MLP	OCSVM	TAnoGAN
广告	F_1	**0.763 6**	0.329 2	0.408 9	0.123 7	0.420 9	0.399 8	0.644 0	0.653 8	0.384 1	0.193 5
	R	0.808 1	0.279 2	0.339 8	0.091 4	0.344 6	0.330 6	0.720 3	**0.837 9**	0.314 1	0.194 4
	P	**0.801 0**	0.463 3	0.584 7	0.203 7	0.636 8	0.619 1	0.605 7	0.599 5	0.571 1	0.195 2
	A_{cc}	**0.943 2**	0.894 8	0.903 2	0.853 4	0.914 5	0.905 4	0.908 2	0.836 9	0.904 3	0.746 7
云监控	F_1	**0.581 6**	0.533 3	0.390 6	0.512 2	0.350 6	0.357 7	0.514 4	0.501 0	0.456 6	0.495 9
	R	**0.732 9**	0.523 0	0.392 8	0.491 7	0.355 9	0.354 8	0.722 9	0.657 3	0.414 8	0.649 0
	P	0.544 8	0.622 2	0.475 8	**0.640 6**	0.415 7	0.437 4	0.424 5	0.457 9	0.622 1	0.404 9
	A_{cc}	0.834 1	0.861 6	0.857 2	**0.867 3**	0.834 1	0.818 8	0.790 4	0.816 1	0.859 0	0.814 3
多源	F_1	**0.622 4**	0.195 6	0.262 7	0.462 0	0.418 5	0.266 7	0.351 2	0.321 4	0.358 3	0.409 4
	R	**0.904 1**	0.260 3	0.289 3	0.401 9	0.343 9	0.246 4	0.484 2	0.424 7	0.299 4	0.522 6
	P	0.520 2	0.316 3	0.411 2	**0.678 5**	0.580 9	0.415 9	0.356 2	0.365 7	0.568 7	0.338 7
	A_{cc}	0.827 0	0.807 8	0.873 7	**0.914 8**	0.906 0	0.887 4	0.778 1	0.761 7	0.910 2	0.844 2
交通量	F_1	**0.779 9**	0.475 8	0.552 0	0.411 9	0.433 0	0.285 9	0.737 4	0.687 1	0.362 1	0.319 7
	R	**0.943 2**	0.425 7	0.533 9	0.333 6	0.386 2	0.251 8	0.896 4	0.818 1	0.297 1	0.339 3
	P	0.699 0	0.567 6	0.650 7	0.605 7	0.538 2	0.341 4	0.682 6	**0.700 3**	0.517 9	0.335 0
	A_{cc}	**0.930 5**	0.899 3	0.922 4	0.903 0	0.893 8	0.846 1	0.921 5	0.892 4	0.896 3	0.819 3
推特	F_1	0.682 3	0.566 5	0.491 0	0.346 2	0.510 4	0.354 0	0.507 5	0.502 5	0.124 3	**0.697 6**
	R	0.802 9	0.616 4	0.379 2	0.249 8	0.511 4	0.290 1	0.804 3	0.778 6	0.090 9	**0.979 3**
	P	0.655 4	0.630 6	**0.778 0**	0.601 0	0.594 9	0.552 1	0.396 6	0.395 5	0.209 0	0.542 1
	A_{cc}	0.910 7	0.882 2	**0.923 6**	0.897 6	0.903 4	0.868 9	0.826 9	0.828 5	0.900 6	0.906 1
平均	F_1	**0.661 9**	0.455 4	0.419 3	0.404 9	0.415 9	0.338 0	0.538 4	0.521 8	0.347 9	0.461 1
	R	**0.814 2**	0.458 1	0.388 8	0.352 2	0.390 3	0.306 4	0.730 2	0.695 5	0.298 3	0.596 3
	P	**0.620 3**	0.550 0	0.570 4	0.576 8	0.524 9	0.467 5	0.469 9	0.485 1	0.504 2	0.387 1
	A_{cc}	0.877 6	0.867 8	**0.889 4**	0.884 4	0.878 7	0.854 8	0.830 9	0.824 6	0.886 8	0.830 4

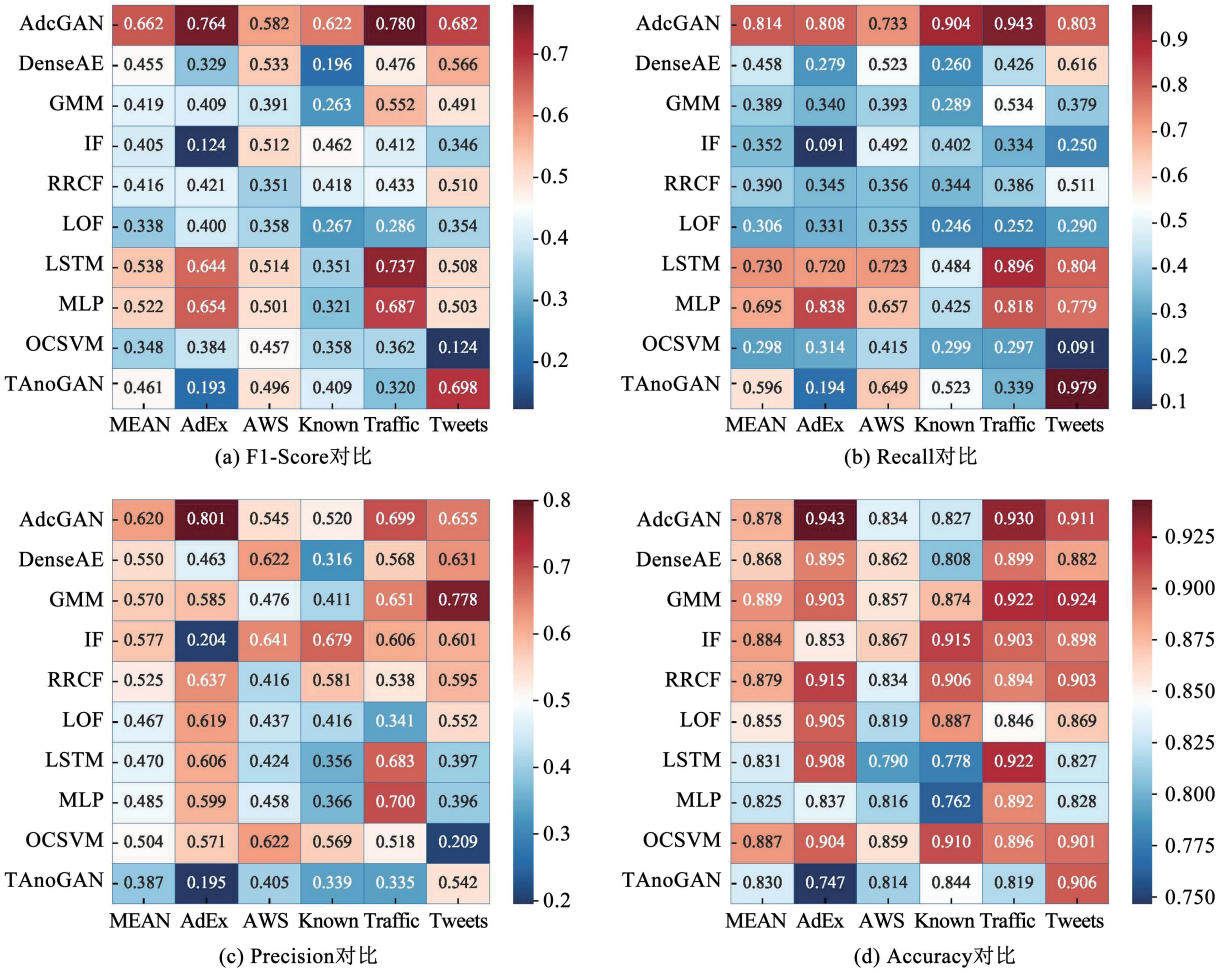


图 5　AdcGAN 与基准方法的异常检测性能

Fig. 5　Performance of AdcGAN and baseline methods

5）与其他 GAN 方法对比：相较于 TAnoGAN，AdcGAN 的优势明显，提升了 20.1% 的 F_1、21.8% 的召回率、23.3% 的精准度以及 4.8% 的准确率。在 TAnoGAN 里学习的是一段窗口数据的生成分布，在其异常评价时还需通过梯度下降的方法求解生成向量，因此在异常检测的效率上比较繁琐，AdcGAN 的上下文生成设计使得其在效率和性能上都有提升。

从实验结果的对比分析中可以知道，基于 GAN 的 AdcGAN 可以有效地进行时间序列上的异常检测。虽然在整体上表现最佳，但是在一些分类数据上表现并非最佳，说明还存在改进的空间。现在对异常检测性能的影响因素做一些讨论：

1）数据分布：不同类型数据的数据分布具有差异性，这对模型的合理架构、类型提出了不同的要求，通用模型是困难的。AdcGAN 由于采用了对抗生成的思想，在不同类型的数据上均取得不错的性能的同时波动最小，说明有着很好的稳定性。

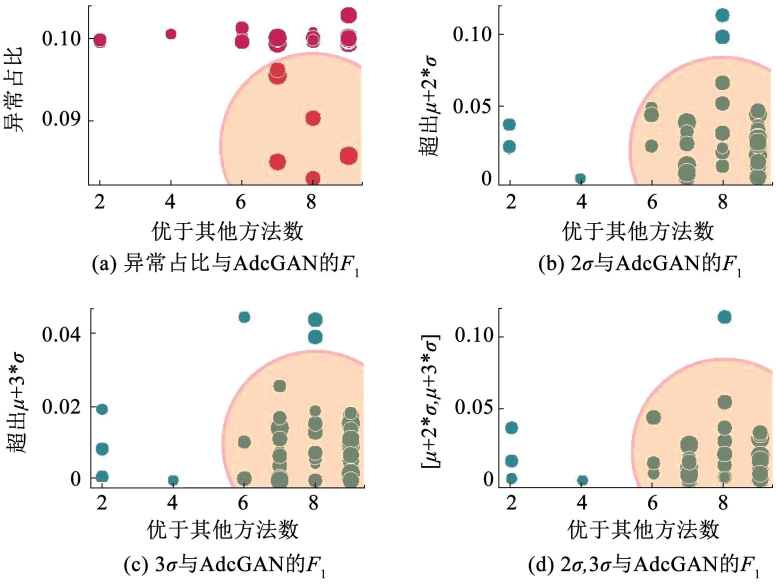
2）数据特征：模型性能与时间序列数据集的值分布也有一定关系，对不同数据的异常占比以及 $N-\sigma$ 进行统计分析，结果见图 6。

从图中可以看出，随着异常占比或超出 $N-\sigma$ 的比例增多，AdcGAN 的性能降低。这是由于大部分的时序异常中的异常值、异常行为、异常模式都包含了异常值或者由异常值引起。因此，异常或 $N-\sigma$ 占比的增多破坏了真实分布的可学习性。

3）异常阈值：阈值并不会影响异常的评分，但是会决定异常点和异常区间的标记。图 4 中的预警点是一个典型的例子，它是否是异常点取决于更多的先验知识。在不同的场景下，什么程度的偏差被认定为异常，这是一个领域的问题，与专家知识更相关。本文提出的自动阈值方法是一种基于数据统计信息的合理方式，在具体应用中可以结合专家知识进行调整。

作为一种基于预测的异常检测方法，AdcGAN 的应用场景不局限于异常检测，还可以用于时间序列的预测。每个时间点的输出是一个概率分布，而不是一个单一的值，这使得结果更适用于需要决策的场景。异常检测结果可以从两个方面来解释：真实数据与预测值之差的期望值表示观察中的数据误

差,而预测分布的方差意味着模型本身对给定条件数据进行预测的不确定性。本文提出的 AdcGAN 将这两者结合起来,对异常的判断更加合理,为后续的异常解释提供了依据。



(a) 异常占比与AdcGAN的F_1 (b) 2σ与AdcGAN的F_1

(c) 3σ与AdcGAN的F_1 (d) $2\sigma,3\sigma$与AdcGAN的F_1

图 6 数据集统计特征和 AdcGAN 的 F_1 关系

Fig. 6 Relation between statistics of datasets and F_1 of AdcGAN

5 结 论

1)在将时序数据转换为条件上下文的基础上,构建了上下文生成对抗网络,利用 GAN 和 Dropout 结合的方法,减少了模型不确定性的影响,并输出了任意时刻时序数据的条件分布。

2)根据时序数据的条件分布,从观测数据偏差和模型预测不确定性 2 个方面评价数据的异常程度,增加了异常检测结果的可信度和鲁棒性,根据数据统计信息为不同的时间序列自动设置异常阈值,最终完成每个时刻的异常检测。

3)通过实验验证,本文提出的 AdcGAN 方法有效提升了针对时间序列数据的异常检测效果,在整体性能上要明显优于其它基准算法,与第二好的方法相比,分别提高了 12.4% 的 F_1、8.4% 的召回率和 4.3% 的精确度,并且在不同类别的数据上,具有最小的 F_1 波动,保持了很好的稳定性。

参考文献

[1] SOLDANI J, BROGI A. Anomaly detection and failure root cause analysis in (micro) service-based cloud applications: a survey[J]. ACM Computing Surveys, 2022, 55(3): 1

[2] CUI Lei, QU Youyang, XIE Gang, et al. Security and privacy-enhanced federated learning for anomaly detection in IoT infrastructures[J]. IEEE Transactions on Industrial Informatics, 2021, 18(5): 3492

[3] 丁小欧, 于晟健, 王沐贤, 等. 基于相关性分析的工业时序数据异常检测[J]. 软件学报, 2020, 31(3): 726
DING Xiaoou, YU Shengjian, WANG Muxian, et al. Abnormal detection of industrial time series data based on correlation analysis [J]. Journal of Software Science, 2020, 31(3): 726

[4] FERRAG M A, MAGLARAS L, MOSCHOYIANNIS S, et al. Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study[J]. Journal of Information Security and Applications, 2020, 50: 1

[5] YAN Xudong, ZHANG Huaidong, XU Xuemao, et al. Learning semantic context from normal samples for unsupervised anomaly detection[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2021, 35(4): 3110

[6] ERGEN T, KOZAT S S. Unsupervised anomaly detection with LSTM neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 31(8): 3127

[7] WELLS M. Applied econometric time series[J]. Journal of the American Statistical Association, 1995, 90(431): 1135

[8] ZENG Jia, ZHANG Lei, SHI Gaotao, et al. An ARIMA based real-time monitoring and warning algorithm for the anomaly detection [C]//2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS). Shenzhen: IEEE, 2017: 469

[9] ESTER M, KRIEGEL H P, SANDER J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise [C]//Proceedings of 2nd International Conference Knowledge Discovery and Data Mining. Palo Alto: AAAI Press, 1996, 96 (34): 226

[10] KEOGH E, CHAKRABARTI K, PAZZANI M, et al. Locally adaptive dimensionality reduction for indexing large time series databases [C]//Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2001, 30(2): 151

[11] KISI O, PARMAR K S. Application of least square support vector machine and multivariate adaptive regression spline models in long term prediction of river water pollution[J]. Journal of Hydrology, 2016, 534: 104

[12] SCHÖLKOPF B, WILLIAMSON R, A SMOLA, et al. Support vector method for novelty detection [J]. Advances in Neural Information Processing Systems, 1999, 12: 1

[13] CHANDOLA V, MITHAL V, KUMAR V. Comparative evaluation of anomaly detection techniques for sequence data [C]//2008 Eighth IEEE International Conference on Data Mining. Pisa: IEEE, 2008: 743

[14] SERPEN G, AGHAEI E. Host-based misuse intrusion detection using PCA feature extraction and kNN classification algorithms[J]. Intelligent Data Analysis, 2018, 22(5): 1101

[15] FARZAD A, GULLIVER T A. Log message anomaly detection with fuzzy C-means and MLP[J]. Applied Intelligence, 2022: 1

[16] WU Di, JIANG Zhongkai, XIE Xiaofeng, et al. LSTM learning with Bayesian and Gaussian processing for anomaly detection in industrial IoT[J]. IEEE Transactions on Industrial Informatics, 2019, 16(8): 5244

[17] GAO Honghao, QIU Binyang, DURAN BARROSO R J, et al. TSMAE: A novel anomaly detection approach for internet of things time series data using memory-augmented autoencoder[J]. IEEE Transactions on Network Science and Engineering, 2022: 1

[18] BORGHESI A, BARTOLINI A, LOMBARDI M, et al. Anomaly detection using autoencoders in high performance computing systems [C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2019, 33(1): 9428

[19] THILL M, KONEN W, WANG H, et al. Temporal convolutional autoencoder for unsupervised anomaly detection in time series[J]. Applied Soft Computing, 2021, 112: 1

[20] 陈磊，秦凯，郝矿荣. 基于集成 LSTM-AE 的时间序列异常检测方法[J]. 华中科技大学学报(自然科学版)，2021，49(11)：35
CHEN Lei, QIN Kai, HAO Kuangrong. A time series anomaly detection method based on integrated LSTM-AE[J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2021, 49(11): 35

[21] HAN Xu, CHEN Xiaohui, LIU Lipin. GAN ensemble for anomaly detection[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2021, 35(5): 4090

[22] KLIGER M, FLEISHMAN S. Novelty detection with GAN[Z]. arXiv: 1802. 10560, 2018: 1

[23] GEIGER A, LIU Dongyu, ALNEGHEIMISH S, et al. TadGAN: time series anomaly detection using generative adversarial networks [C]//2020 IEEE International Conference on Big Data (Big Data). Atlanta: IEEE, 2020: 33

[24] LI Dan, CHEN Dacheng, JIN Baihong, et al. MAD-GAN: multivariate anomaly detection for time series data with generative Adversarial Networks[C]//International Conference on Artificial Neural Networks. Munich: Springer International Publishing, 2019: 703

[25] BASHAR M A, NAYAK R. TAnoGAN: time series anomaly detection with generative adversarial networks[C]//2020 IEEE Symposium Series on Computational Intelligence. Canberra: IEEE, 2020: 1778

[26] MICHAU G, FINK O. Unsupervised transfer learning for anomaly detection: application to complementary operating condition transfer [J]. Knowledge-Based Systems, 2021, 216: 1

[27] ZHAO Hang, WANG Yujing, DUAN Juanyong, et al. Multivariate time-series anomaly detection via graph attention network[C]// 2020 IEEE International Conference on Data Mining (ICDM). Sorrento: IEEE, 2020: 841

[28] SU Ya, ZHAO Youjian, NIU Chenhao, 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. New York: ACM, 2019: 2828

[29] MIRZA M, OSINDERO S. Conditional generative adversarial nets [Z]. arXiv: 1411. 1784, 2014: 1

[30] GAL Y, GHAHRAMANI Z. Dropout as a bayesian approximation: representing model uncertainty in deep learning[C]//Proceedings of the 33rd International Conference on Machine Learning. New York: JMLR, 2016: 1050

[31] AHMAD S, LAVIN A, PURDY S, et al. Unsupervised real-time anomaly detection for streaming data[J]. Neurocomputing, 2017, 262: 134

[32] BREUNIG M M, KRIEGEL H P, NG R T, et al. LOF: Identifying density-based local outliers[C]//Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2000: 93

[33] Ma J, PERKINS S. Time-series novelty detection using one-class support vector machines[C]//Proceedings of the International Joint Conference on Neural Networks. Portland: IEEE, 2003, 3: 1741

[34] LIU F T, TING Kaiming, ZHOU Zhihua. Isolation forest[C]// 2008 Eighth IEEE International Conference on Data Mining. Pisa: IEEE, 2008: 413

[35] GUHA S, MISHRA N, ROY G, et al. Robust random cut forest based anomaly detection on streams[C]//Proceedings of the 33rd International Conference on Machine Learning. New York: JMLR, 2016: 2712

[36] HOCHREITER S, SCHMIDHUBER J. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735

[37] BLANCO R, MALAGóN P, BRIONGOS S, et al. Anomaly detection using gaussian mixture probability model to implement intrusion detection system[C]//International Conference on Hybrid Artificial Intelligence Systems. León: Springer International Publishing, 2019: 648

（编辑　苗秀芝）