# 基于 VAE-WGAN 的多维时间序列异常检测方法

段雪源[1,2]，付钰[1]，王坤[1,3]

（1. 海军工程大学信息安全系，湖北 武汉 430033；
2. 信阳师范学院计算机与信息技术学院，河南 信阳 464000；
3. 信阳职业技术学院数学与信息工程学院，河南 信阳 464000）

**摘 要**：针对传统半监督深度异常检测模型对非平衡多维数据分布学习能力不足及模型训练困难等问题，提出一种基于 VAE-WGAN 架构的多维时间序列异常检测方法，利用 VAE 作为 WGAN 的生成器，使用 Wasserstein 距离作为模型拟合分布与待测数据真实分布之间的度量，学习复杂的高维数据分布。利用滑动窗口划分时间序列，使用正常序列数据训练模型；根据待测序列在训练好的模型中的异常得分，结合自适应阈值技术进行异常判定。实验表明，该方法具有模型容易训练且稳定性强的特点，并且在精确率、召回率、F1 值等异常检测性能指标上，比现有的生成式异常检测模型有明显提升。

**关键词**：时间序列数据；变分自编码器；Wasserstein 生成对抗网络；异常检测

**中图分类号**：TP391

**文献标志码**：A

**DOI:** 10.11959/j.issn.1000−436x.2022050

# Multi-dimensional time series anomaly detection method based on VAE-WGAN

DUAN Xueyuan[1,2], FU Yu[1], WANG Kun[1,3]

1. Department of Information Security, Naval University of Engineering, Wuhan 430033, China
2. College of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China
3. School of Mathematics and Information Engineering, Xinyang Vocational and Technical College, Xinyang 464000, China

**Abstract:** As the deficiency of learning ability of traditional semi-supervised depth anomaly detection model to unbalanced multidimensional data distribution and the difficulty of model training, a multi-dimensional time series anomaly detection method based on VAE-WGAN architecture was proposed. VAE was used as a generator of WGAN. The Wasserstein distance was used as a measure between the model fitting distribution and the real distribution of the data to be measured, complex and high-dimensional data distributions could be learned. A sliding window was applied to divide the time series, the normal sequence data were used to train the model. According to the abnormal score of the waiting test sequence in the trained model, the anomaly was judged with adaptive threshold technology. The experimental results show that the model is easy to train and stable, and has obvious improvement over the existing generative anomaly detection model in accuracy, recall rate, F1 score and other anomaly detection performance indicators.

**Keywords:** time series data, variational auto-encoder, Wasserstein generative adversarial network, anomaly detection

**收稿日期**：2021−12−01；**修回日期**：2022−02−21

**通信作者**：付钰，fuyu0219@163.com

**基金项目**：国家重点研发计划基金资助项目（No.2018YFB0804104）

**Foundation Item:** The National Key Research and Development Program of China (No.2018YFB0804104)

## 0 引言

时间序列异常检测就是在时间序列中寻找不符合规则的事件或行为[1]，常被应用于网络安全检测、金融数据分析、医学病理诊断等众多领域。然而随着科技发展和各类传感器在电子信息系统中的广泛应用，记录系统状态信息的时间序列数据呈现出海量高维的特点，传统的以特征工程为基础的异常检测方法很难从海量高维数据中自动完成特征提取和异常识别。而且现实环境产生的时间序列中异常样本稀少并且标注困难限制了以分类、预测为基础的有监督异常检测方法的应用。因此，利用无监督或半监督的深度学习方法对时间序列进行异常检测逐渐成为电子信息设备异常检测领域的研究热点。

以变分自编码器（VAE, variational auto- encoder）[2]和生成对抗网络（GAN, generative adversarial network）[3]结构为代表的无监督或半监督深度生成异常检测方法，利用无标签的正常数据即可完成训练，突破了对数据标注和正负样本平衡性要求的限制。VAE 以分布参数的重构概率作为异常度量，能够对不同结构的输入数据进行检测，但由于 VAE 训练时缺少限制指导，导致其稳健性较差且容易出现过拟合。GAN 通过学习正常样本数据的分布生成近似数据，当异常样本输入训练好的 GAN 时，生成样本与输入样本之间存在较大差异，这种差异可以作为异常检测的依据，但由于 GAN 缺少自编码器，导致其在训练初期较困难。传统的 GAN 使用 $f$ 散度作为目标函数进行训练，存在着因“梯度失稳”（“梯度消失”或”“梯度爆炸”）而无法完成训练和因“模式崩溃”生成样本单一的问题。

进行异常判定时都需要根据一定门限（阈值）作为划分正常和异常的界限，阈值的设定通常由人工完成。但对于复杂多维时间序列，尤其是对于有多个关键绩效指标（KPI, key performance indicator）的多维时间序列，很难由人工进行统一的阈值设定。

为解决上述问题，本文提出一种基于 VAE-WGAN 架构的半监督多维时间序列异常检测方法。利用 VAE 作为生成器，并利用 WGAN 的判别结果调整 VAE 的分布参数，使用 Wasserstein 距离作为模型损失函数，期望在模型训练时更加稳健，避免出现“梯度失稳”和“模式崩溃”等问题；利用时间窗口划分出时序子序列，力求提升模型训练与检测的准确率和时效性；探索自适应的异常判定及裁剪方法，以提升模型对时间序列异常检测的效果。

## 1 相关研究现状

时间序列数据中的异常是由不符合规则的非正常行为引发的超预期数据模式[4]。按异常数据的表现形式通常可分为以下三类：1) 点异常，即与正常模式数据表现出较大差异的单点数据；2) 上下文异常，即在特定上下文环境里与正常模式差异较大的某项数据；3) 集合异常，即与其他数据有显著不同的一组数据。

VAE 融合了贝叶斯算法和深度学习的优势，在传统自编码器（AE, auto encoder）的基础上引入变分思想，然而因隐变量分布未知无法直接进行梯度的反向求导，使用蒙特卡洛采样法求解又需要较大的计算量。因此，Kingma 提出重参数化，使隐变量分布尽量拟合独立的已知分布，VAE 结构如图 1 所示。

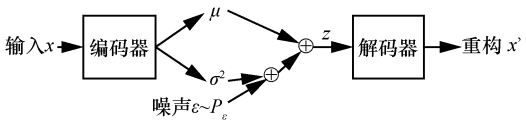



图 1 VAE 结构

利用已知分布样本代替隐变量样本进行求解，解决原梯度参数不可微的问题，推动 VAE 及其变体在现实领域的研究与应用。

利用 VAE 进行时间序列异常检测，以时间序列中潜在变量分布参数的重建概率作为异常检测的度量，不受数据结构限制，比计算数据本身的重构误差更加客观合理[5]。清华大学的裴丹团队提出基于 VAE 的无监督周期性 KPI 异常检测算法 Dount[6]，这是第一个具有坚实理论解释的基于 VAE 架构的异常检测算法。该团队还针对非高斯分布的复杂多关键绩效指标数据，提出另一种基于 VAE 的高性能无监督异常检测算法 Buzz[7]。与 Donut 只能处理平滑的 KPI 数据相比，Buzz 对于复杂 KPI 数据中的异常也具有很好的检测能力；在 Donut 中，输入检测模型的数据单位是窗口，模型直接把 KPI 数据切分成若干个窗口，而 Buzz 则是先把大型复杂数据分割成多个区域，再对每个区域进行窗口切分。这种“分区”的做法在缓解计算压力的同时，能够更好地提取数据特征。Pol 等[8]提出基于条件变分自编码器（CVAE, conditional variational auto-encoder）异常检测方法，在 VAE 中添加

了一些关联性的条件限制，通过在特定数据集上进行实验证明了其方法的有效性。基于 VAE 架构的异常检测技术还被广泛应用于网络安全威胁态势评估[9]、飞行数据异常检测[10]和对抗攻击防御技术研究[11]等应用领域。

GAN 的生成器通过学习真实数据的分布，将随机噪声转换为与真实数据相似的生成数据，具有对复杂高维数据分布的建模能力，然而原始的 GAN 对生成数据与真实数据的语义上有意义的特征并不十分清晰。因此，Beula 等[12]提出了双向生成对抗网络（BiGAN, bidirectional GAN）模型。从图 2 可知，BiGAN 就是在 GAN 的基础上加入了一个将数据映射到隐特征空间的编码器，目的是能够利用编码器无监督地提取原始数据特征；判别器也进行了相应改进，从对 $x$ 或 $G(z)$一组数据的判断转变为对 $(G(z),z)$、$(x,E(x))$ 两组数据的判断，博弈的最终目标也演变为判别器无法分辨输入数据是来自生成器还是编码器。这种加入自编码器的改进使 GAN 具有学习有意义特征表示的能力。

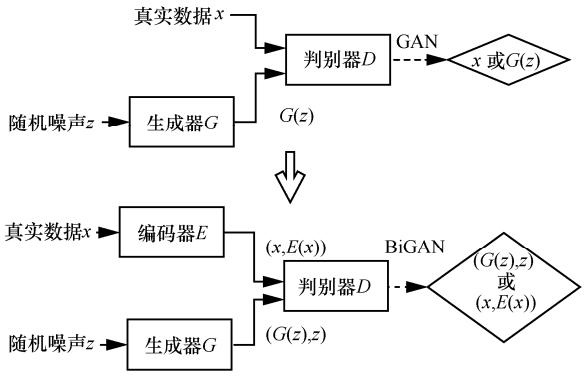



图 2　GAN 与 BiGAN 结构

利用 GAN 进行异常检测的一般步骤是先使用正常样本数据训练 GAN，使生成器完全学习到正常数据的真实分布，此时 GAN 已经具备重建正常数据样本的能力。当输入的数据是从未见过的异常样本时，GAN 重建的样本和原始输入样本将表现出明显差异，当这种差异超过一定范围时即判定输入数据存在异常[13]。Bashar 等[14]提出了利用 GAN 进行时间序列异常检测的方法 TAnoGAN，利用训练生成器在潜在空间生成逼真样本，然后将真实样本映射到潜在空间，并在潜在空间进行重构，通过输入样本分布与生成样本分布的拟合度来判定输入样本是否存在异常。Li 等[15]提出利用 GAN 对时间序列数据进行多元异常检测的方法 MAD-GAN，将带有记忆效应的循环神经网络作为 GAN 的生成器来捕获数据序列中的时间相关性，利用判别器产生的判别误差、生成器生成的重构数据与输入数据的重构误差共同作为异常判别的依据。Chen 等[16]提出基于自编码器和 GAN 结构的 DAEMON 算法，自编码器用于重构输入的时间序列数据，GAN 结构分别用于约束编码器的中间输出以及解码器的重构输出，使自编码器结构的训练过程更加稳健。DAEMON 被应用于在线检测，其将在线数据输入检测器得到重构数据，通过计算在线数据与重构数据的异常得分进行异常判断，该算法在公开数据集上表现出色。还有学者将 GAN 拓展到时间序列预测[17]、恶意流量验证[18]、入侵检测[19]、加密流量识别[20]、僵尸网络检测[21]等研究中。

然而，以上算法都是基于传统 GAN 及其变体，使用 $f$ 散度来衡量假设分布与真实分布之间的“差异”，训练过程中依然可能出现“梯度失稳”和“模式崩溃”问题，因此本文提出利用 Wasserstein 距离代替 $f$ 散度作为分布差异的度量。

Wasserstein 距离又称为推土机距离（EMD, earth mover’s distance）、最优传输距离（OT, optimal transport）等[22]，将一个分布转变为另一个分布所需要的代价定义为 $W(P_r,P_g)$，即

$$W(P_r,P_g)=\inf_{\gamma\sim\prod(P_r,P_g)}\mathbb{E}_{(x,y)\sim\gamma}\left[\|x-y\|\right] \tag{1}$$

其中，$P_r$ 和 $P_g$ 为 2 个样本分布，$\prod(P_r,P_g)$ 为 $P_r$ 和 $P_g$ 分布组成的所有可能的联合分布的集合。对每一个可能的联合分布 $\gamma$，从中采样一个样本对 $x$ 和 $y$ 都能计算出这个样本对的距离 $\|x-y\|$，所有可能的联合分布中期望值的下确界 $\inf_{\gamma\sim\prod(P_r,P_g)}\mathbb{E}_{(x,y)\sim\gamma}\left[\|x-y\|\right]$ 即 Wasserstein 距离。直观上，把 $\mathbb{E}_{(x,y)\sim\gamma}\left[\|x-y\|\right]$ 理解为在 $\gamma$ 这个“路径规划”下，将一堆“沙土”从 $P_r$ 挪到 $P_g$ 所需的“消耗”，而 $W(P_r,P_g)$ 就是“最优路径规划”下的“最小消耗”，因此又叫推土机距离。

Wasserstein 距离相对于 $f$ 散度的优越性在于即便 2 个分布的重叠测度为零，Wasserstein 距离仍然能够反映它们的“远近”。以 Wasserstein 距离作为分布度量的 GAN 称为 WGAN[23]，Wasserstein 距离不仅可以提供有效的梯度信息，解决“梯度失稳”和“模式崩溃”的问题，还能为训练进程提供可靠指标。Geiger 等[24]提出 TadGAN 时间序列异常检测模型，就是利用 2 个以 Wasserstein loss 作为训练

目标的 WGAN 来克服“梯度失稳”和“模式崩溃”问题，且在异常检测实验中表现出较好的效果，但因为缺少自编码器，模型在训练初始阶段存在一定困难。Wasserstein 距离还被应用到自编码网络中，王星等[25]利用 Wasserstein 自编码器构建的半监督异常检测模型 WAE-AD，能够学习复杂的高维数据分布，利用待测数据在训练好的模型中的异常得分进行异常判定，但对该模型进行训练时需要提供带标签的训练样本。

如上所述，虽然当前单独利用 VAE 和 WGAN 进行异常检测的研究比较多，但是将两者结合起来对时间序列进行异常检测的研究并不多。虽然 2016 年 Larsen 等[26]曾提出 VAE-GAN 结构并利用 GAN 来改善 VAE 生成图像模糊的问题，但在时间序列异常检测领域，直到 2020 年 Niu 等[27]提出基于循环神经网络结构的 VAE-GAN 时间序列异常检测方法，才将编码器、生成器和鉴别器进行联合训练，这种方法不需要在异常检测阶段考虑空间的最佳映射问题，即可降低异常检测的时间消耗。然而该方法仍然使用 $f$ 散度作为目标函数，训练过程中依旧可能会出现“梯度失稳”和“模式崩溃”等问题。

## 2　VAE-WGAN 异常检测模型

本文提出的 VAE-WGAN 异常检测模型总体框架如图 3 所示，由数据设计、VAE-WGAN 和异常检测 3 个模块组成。

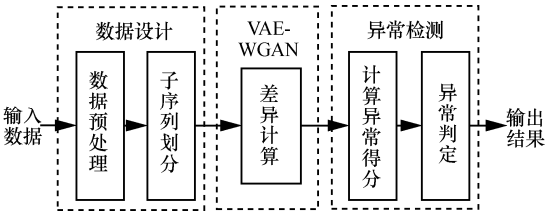



图 3　VAE-WGAN 异常检测模型总体框架

### 2.1　数据设计

现实环境中产生的时间序列数据可能是不完整、不一致的，为确保 VAE-WGAN 异常检测模型的训练及异常检测结果的客观性和准确性，需要对原始输入数据进行必要的“设计”，主要有数据预处理和时间子序列划分 2 个处理过程。

#### 2.1.1　数据预处理

数据预处理是为了保证数据的可读性和统一性而进行的数据降维、文本数值化、数据清洗、数据去势、数据归一化等操作。

1) 数据降维。原始序列中一些属性具有高度相关性或者具有线性关系，去除一些高相关性特征，可以减少运算开支，提高检测效率。

2) 文本数值化。原始序列的属性特征值并不完全是数字，还可能是文本信息，因此需要将这些文本转换成相应的离散数值以便参与运算。

3) 数据清洗。原始序列中可能存在重复或缺失的数据，需要利用数据清洗技术对这些冗余和缺失数据进行清除整理。

4) 数据去势。数据中的线性趋势会影响训练效果，可以采用减去线性最小二乘拟合的方法去除数据中的线性趋势，使时间序列更加平稳。

5) 数据归一化。不同属性数据的量纲不同，特征向量的取值范围也不尽相同，差异较大时会影响检测结果，需要对数据进行归一化处理，使数据分布在设定的区间内。

#### 2.1.2　时间子序列划分

预处理后的数据被分为训练集、验证集和测试集，这些数据序列仍然非常长，直接输入 VAE-WGAN 模块进行运算会导致训练时间增加、参数更新缓慢、计算开支过度消耗等问题。因此，本文对各时间序列再利用滑动窗口技术进行子序列划分，以保证训练与检测的准确率和时效性。

### 2.2　VAE-WGAN 模型构建与训练

#### 2.2.1　VAE-WGAN 模型构建

VAE-WGAN 模型的框架结构如图 4 所示，其包含 3 个组块，即一个 VAE 组块和 2 个 WGAN 组块，其中，WGAN 组块包括 $\mathrm{WGAN}_e$ 和 $\mathrm{WGAN}_d$。

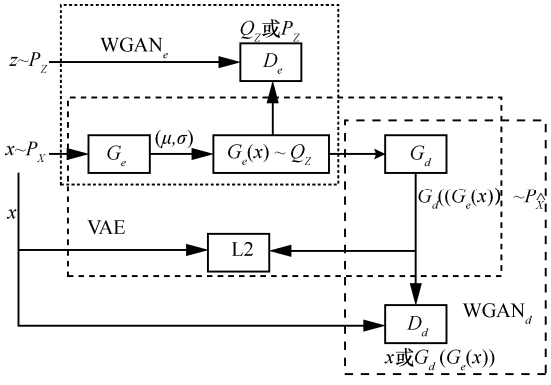



图 4　VAE-WGAN 模型的框架结构

1) VAE 组块。该组块由编码器 $G_e$ 和解码器 $G_d$ 组成（$G_e$ 和 $G_d$ 同时作为 2 个 GAN 结构中的生成

器），学习输入序列 $x$ 的特征，并通过编码和解码的方式得到重构序列 $G_d(G_e(x))$，利用周期一致性损失函数，如式(2)所示，最小化输入序列与重构序列之间的欧氏距离（L2 范数距离）来训练编码器 $G_e$ 和解码器 $G_d$，防止它们作为生成器时产生矛盾。

$$\min_{\{G_e,G_d\}} V_{\mathrm{L2}}(G_e,G_d)=\mathbb{E}_{x\sim P_X}\left[\left\|x-G_d(G_e(x)\right\|_2\right] \quad (2)$$

2) WGAN$_e$ 组块。该组块由 VAE 组块的编码器 $G_e$ 和判别器 $D_e$ 组成，其中 $G_e$ 作为生成器。对于输入 $x\sim P_X$，VAE 会产生一个中间输出 $G_e(x)\sim Q_Z$，引入一个高斯分布 $P_Z$ 作为约束，并找到一个条件分布 $Q(Z|X)$，使 $Z$ 的边界为 $Q_Z(Z)\coloneqq\mathbb{E}_{x\sim P_X}[Q(Z|X)]$，即 $Q_Z=\int Q(Z|X)\mathrm{d}P_X$，通过调整 $G_e$ 的参数使 $G_e(x)$ 服从的 $Q_Z$ 分布与先验分布 $P_Z$ 之间的距离不断减小，以实现训练生成器 $G_e$ 的目标，利用 Wasserstein 距离计算 WGAN$_e$ 的目标函数为

$$\min_{G_e}\max_{\|D_e\|\leqslant 1} V(D_e,G_e)=\mathbb{E}_{z\sim P_Z}[D_e(z)]-\mathbb{E}_{x\sim P_X}[D_e(G_e(x))] \quad (3)$$

3) WGAN$_d$ 组块。该组块由 VAE 组块的解码器 $G_d$ 和判别器 $D_d$ 组成，其中 $G_d$ 作为生成器。使用范数距离并不能很准确地描述输入序列与输出序列的相似性，为了让变分自编码器的重构输出更接近于输入数据的真实分布，提出利用判别器 $D_d$ 进一步约束重构数据与输入数据的差异，WGAN$_d$ 同样使用 Wasserstein 距离计算目标函数

$$\min_{G_d}\max_{\|D_d\|\leqslant 1} V(D_d,G_d)=\mathbb{E}_{x\sim P_X}[D_d(x)]-\mathbb{E}_{x\sim P_X}[D_d(G_d(G_e(x)))] \quad (4)$$

### 2.2.2　VAE-WGAN 模型训练

由于 GAN 结构的网络需要异步训练，因此 VAE-WGAN 模型的 3 个组块分别对应了 3 个异步的训练过程，它们都有各自的损失函数及优化器。由于 GAN 的判别器在训练阶段，只涉及自身，因此可以直接用 $z\sim P_Z$ 和 $x\sim P_X$ 中的样本作为输入训练 $D_e$ 和 $D_d$。而 2 个 GAN 的生成器 $G_e$ 和 $G_d$ 又是 VAE 的编码器和生成器，因此在训练 $G_e$、$G_d$ 和 VAE 时要同时考虑 3 个目标函数，利用这 3 个目标函数的加权和作为最终的训练目标

$$\min_{\{G_e,G_d\}}\max_{\{\|D_e\|\leqslant 1,\|D_d\|\leqslant 1\}} \lambda V_{\mathrm{L2}}(G_e,G_d)+\gamma V(D_e,G_e)+\mu V(D_d,G_d) \quad (5)$$

其中，$\lambda$、$\gamma$、$\mu$ 为各损失函数的权值，$\lambda+\gamma+\mu=1$。采用控制变量法固定其他参数，对比$\lambda$、$\gamma$、$\mu$ 不同取值时的实验结果发现，当$\lambda$=0.4、$\gamma$=0.3、$\mu$=0.3 时，检测模型在测试集和验证集上的检测效果较其他取值更加出色。

为满足利普希茨连续条件，WGAN 可加上梯度惩罚项（GP，gradient penalty）来限制梯度变化范围，同时引入权重裁剪方法将参数权重限制在一定区间内。训练时，采取循环嵌套的方法，生成器每训练一次，判别器先训练几次。判别器对生成数据和真实数据交替采样来计算梯度惩罚项，这种训练方法可以使判别器更加“温和”地训练生成器，避免出现权重参数“扎堆”地聚集在权重裁剪参数区间两端的位置，实现过程如算法 1 所示。

**算法 1**　VAE-WGAN 模型的训练算法

**初始化**　判别器 $D_e$ 和 $D_d$ 的参数 $w_{D_e}$ 和 $w_{D_d}$，生成器 $G_e$ 和 $G_d$ 的参数 $\theta_{G_e}$ 和 $\theta_{G_d}$

1) while 生成器未收敛或未达到设定的迭代次数时 do
2) for $m$=1,⋯, $M_{\mathrm{gen}}$
3)　for $n$=1,⋯, $N_{\mathrm{disc}}$
4)　从真实数据中采样 $x\sim P_X$，从噪声数据中采样 $z\sim P_Z$，生成一个随机数 $\varepsilon\sim$uniform(0,1)
5)　$\hat{x}=\varepsilon z+(1-\varepsilon)G_\theta(x)$，生成混合样本
6)　利用 WGAN-gp 方法更新判别器 $D_e$ 参数 $w_{D_e}$
7)　利用 RMSProp 算法优化参数 $w_{D_e}$
8)　每次更新参数 $w_{D_e}$ 后，把它的绝对值截取到固定区间[−$c$, $c$]
9)　利用步骤 5)～步骤 8)相似的步骤更新判别器 $D_d$ 的参数 $w_{D_d}$
10)　end for
11) 从真实数据中采样 $x\sim P_X$，从噪声数据中采样 $z\sim P_Z$
12) 更新生成器 $G_e$ 和 $G_d$ 的参数 $\theta_{G_e}$ 和 $\theta_{G_d}$
13) 利用 RMSProp 算法优化 2 个生成器参数 $\theta_{G_e}$ 和 $\theta_{G_d}$
14) end for
15) end while

## 2.3　异常检测

利用正常数据训练 VAE-WGAN，使 VAE-WGAN 完全学习到正常数据的真实分布。将待测数据输入训练好的 VAE-WGAN 中，利用生成数据与输入数据之间的差异计算异常得分，当异常得分超过阈值时，可判定输入数据中存在异常。

**2.3.1** 计算异常得分

VAE-WGAN 生成的时间序列与输入时间序列的误差包括重构误差和差别误差，综合 2 种误差的异常得分作为异常判定的依据。

1) 利用重构误差评估异常得分

对于第 $i$ 个子序列 $x_i=[x_i^1,x_i^2,\cdots,x_i^M]$，$M$ 为子序列长度，利用 VAE-WGAN 生成的对应重构序列为 $\hat{x}_i=[\hat{x}_i^1,\hat{x}_i^2,\cdots,\hat{x}_i^M]$，可以根据 $x_i$ 与 $\hat{x}_i$ 中各样本之间的差异计算子序列的重构异常得分。由于相邻样本之间在采样时存在着时间间隔，因此，对于等间隔样本的子序列只需要考虑各样本特征值的差异，而对于非等间隔的子序列还需要考虑持续时间因素，为此设计出点差异和面差异。

点差异是等间隔情况下的输入子序列与重构子序列中各样本在对应维度特征值差异的总和，计算方法为

$$\mathrm{Re}'_{x_i}=\left\|x_i-\hat{x}_i\right\|_l=\sum_{j=1}^{M}\left|x_i^j-\hat{x}_i^j\right| \tag{6}$$

面差异是非等间隔条件下的输入子序列与重构子序列中各样本维度特征在时间邻域 $[-l,l]$ 上面积差值的总和，计算方法为

$$\mathrm{Re}''_{x_i}=\sum_{j=1}^{M}\left[\frac{1}{2l}\int_{t-l}^{t+l}\left|x_i^j-\hat{x}_i^j\right|\mathrm{d}t\right] \tag{7}$$

相对于点差异，面差异更适合发现较长时间段中存在微小差别的区域。点差异也可以认为是特殊的面差异，因此本文使用面差异作为计算重构误差异常得分的方法。

2) 利用判别误差评估异常得分

普通 GAN 的判别器完成“是”与“否”的二分类任务，而 WGAN 输出的是 Wasserstein 距离，表示真实数据或生成数据的“距离”，属于回归任务，因此训练好的 VAE-WGAN 输出的判别误差可以直接作为生成数据与输入数据的异常性度量，即

$$\mathrm{Dd}_{x_i}=D_d(x_i) \tag{8}$$

3) 利用 2 个异常得分计算综合得分

对重构误差异常得分 $\mathrm{Re}_{x_i}$ 和判别误差异常得分 $\mathrm{Dd}_{x_i}$，利用 Z-Score 标准化方法做标准化处理，将处理结果的凸组合作为综合异常得分（简称异常得分）是异常判定的依据，表示为

$$\mathrm{Score}(x_i)=\alpha Z_{\mathrm{Re}}(x_i)+(1-\alpha)Z_{\mathrm{Dd}}(x_i) \tag{9}$$

其中，$\alpha$ 为控制 $Z_{\mathrm{Re}}(x_i)$ 和 $Z_{\mathrm{Dd}}(x_i)$ 的相对重要性的参数，$\alpha\in(0,1)$，$\alpha$ 的取值可根据测试集在训练好的模型上的检测结果达到最优时来确定。实验发现，当 $\alpha=0.5$ 时，模型取得最佳的检测效果。

**2.3.2** 异常判定

根据子序列的异常得分 $\mathrm{Score}(x_i)$，使用阈值法即可判断该子序列是否存在异常。传统的阈值法需要人工设定阈值，对于多 KIP 的多维时间序列无法统一设置的情况，人工逐一设定的效率极低，因此本文采用滑动窗口自适应技术来确定阈值，根据滑动窗口内的异常得分计算本窗口的阈值，当子序列的异常得分大于所在窗口的阈值时，即可判定该子序列为异常子序列。连续异常子序列组成异常序列的判定过程如图 5 所示。

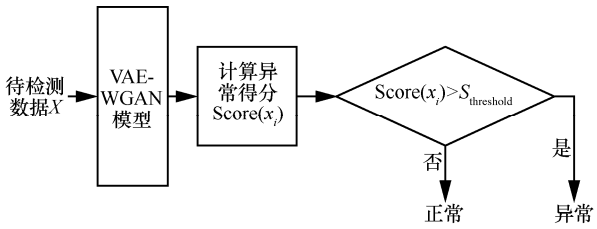



图 5 连续异常子序列组成异常序列的判定过程

1) 设定阈值

滑动窗口自适应阈值技术是把每个子序列的异常得分排列成一维序列 $S_a=\{s_a^1,s_a^2,\cdots,s_a^n\}$，$s_a^n\in R$，计算每个滑动窗口内异常得分的均值 $\overline{s}_a^N$ 和均方差 $\sigma_N$，按照“拉依达准则”将该窗口的阈值设置为均值与 3 个均方差之和[28]。因此每个滑动窗口都有自己的阈值，第 $N$ 个滑动窗口的阈值可表示为 $S_{\mathrm{threshold}}=\overline{s}_a^N+3\sigma_N$，如图 6 所示。

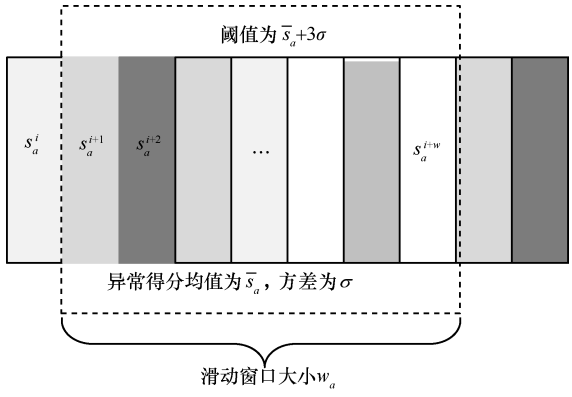



图 6 利用滑动窗口计算阈值示意

滑动窗口大小 $w_a$ 决定了计算设定阈值所需异常得分的个数，步长 $l_a$ 的大小决定了异常检测的精

细度，结合定位需求及运算开支，一般将步长设置为窗口的$\frac{1}{10}$，即$l_a=\frac{w_a}{10}$。

2) 异常判定

采用步进式的异常筛查方法，把每个滑动窗口中那些大于阈值的异常得分对应的子序列判别为异常子序列，连续的异常子序列组成异常序列$X_a=\{x_a^1,x_a^2,\cdots,x_a^k\}$，其中$x_a^i=(x_{\text{start}(i)},\cdots,x_{\text{end}(i)})$。多维时间序列中的每个子序列的异常得分是通过将该时刻每一个维度的误差相加得出的，因此可直接根据特征值误差所对应的属性和时间标签定位异常发生的位置和时刻。

## 3　实验与分析

### 3.1　数据设计

本文选用 4 个公开的时序异常检测数据集，包括 KDD99-sub 数据集、SMAP 数据集、MSL 数据集以及 SWaT 数据集，各数据集主要指标如表 1 所示。

**表 1　　各数据集主要指标**

| 数据集 | 特征维度 | 样本数 | 异常比 |
|---|---|---|---|
| KDD99-sub | 41 | 494 021 | 32.02% |
| SMAP | 25 | 562 800 | 13.13% |
| MSL | 55 | 132 046 | 10.72% |
| SWaT | 51 | 946 719 | 11.98% |

#### 3.1.1　数据降维

使用 Python 数据分析工具包 pandas 读取检测数据集，根据数据的相关性特征，利用决策树技术对原始时间序列数据进行降维处理。以 KDD99-sub 数据集为例，降维前原始数据中包含 DoS、Probe、R2L 和 U2R 四类攻击的 41 种特征，降维后为 16 种，如表 2 所示。表 2 中，各攻击类型的特征在降维前后均有重合。

**表 2 KDD99-sub 数据集降维前后特征数对比**

| 攻击类型 | 原特征数/种 | 降维后特征数/种 |
|---|---|---|
| DoS | 11 | 5 |
| Probe | 14 | 6 |
| R2L | 18 | 6 |
| U2R | 8 | 5 |

#### 3.1.2　数据处理

实验前对数据进行去势、清洗、文本数字化、归一化处理，前两步根据需要选择使用，归一化处理则是利用公式$x'=\frac{x-x_{\min}}{x_{\max}-x_{\min}}$对实验数据做最值运算，将其映射到[0,1]内。

KDD99-sub 数据集预处理前后信息对比如图 7 所示。从图 7 中可以看出，带有标签的原始数据共有 42 列，而处理后数据只有 17 列。

```
==========================================
原始数据集的形状:  (494021, 42)
==========================================
原始数据集前5条记录
   duration protocol_type  …  dst_host_srv_rerror_rate   label
0         0           tcp  …                       0.0  normal
1         0           tcp  …                       0.0  normal
2         0           tcp  …                       0.0  normal
3         0           tcp  …                       0.0  normal
4         0           tcp  …                       0.0  normal

[5 rows × 42 columns]
**********************************************
降维归一化后数据集形状:  (494021, 17)
==========================================
降维归一化后数据集前5条记录:
   0  1         2  3             4  …        32   33   34   38  41
0  0  0  0.294118  9  2.610418e-07  …  0.035294  1.0  0.0  0.0   0
1  0  0  0.294118  9  3.446905e-07  …  0.074510  1.0  0.0  0.0   0
2  0  0  0.294118  9  3.389216e-07  …  0.113725  1.0  0.0  0.0   0
3  0  0  0.294118  9  3.158461e-07  …  0.152941  1.0  0.0  0.0   0
4  0  0  0.294118  9  3.129617e-07  …  0.192157  1.0  0.0  0.0   0

[5 rows × 17 columns]
==========================================
```

图 7　KDD99-sub 数据集预处理前后信息对比

#### 3.1.3　数据集划分

将数据集划分为训练集、验证集和测试集三部分。以 KDD99-sub 数据集为例，从数据集的正常数据中随机抽取 60%作为训练集，再分别从剩余的正常数据和异常数据各随机抽取一半组成验证集，其余另一半的正常数据和异常数据组成测试集。

#### 3.1.4　子序列划分

利用滑动窗口划分时间子序列如图 8 所示。利用大小为 $w$、步长为 $s$ 的滑动窗口对长度为 $T$ 的 $M$ 维时间序列$X=\{x_1,x_2,\cdots,x_T\}$进行划分，得到$X=\{x_i,i=1,2,\cdots,m\}\subseteq R^{w\times M}$，其中$m=\frac{T-w}{s}+1$为子序列的个数。

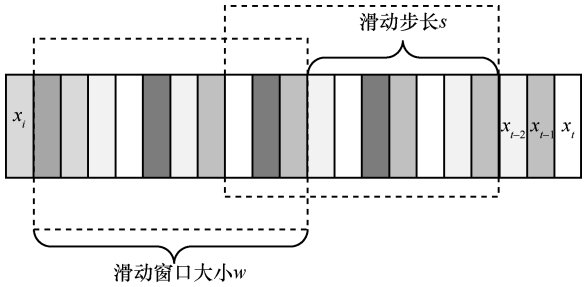



图 8　利用滑动窗口划分时间子序列

KDD99-sub 训练集子序列划分如图 9 所示。从图 9 可知，KDD99-sub 训练集使用窗口宽度为 64、步长为 16 的滑动窗口进行子序列划分后的第 2 个、第 3 个子序列的相关信息。

```
==========================================
训练集第2个子序列形状:  (64, 16)
训练集第2个子序列前5条记录:
    0  1         2  3             4  …   28   32   33   34   38
33  0  0  0.294118  9  2.235441e-07  …  1.0  1.0  1.0  0.0  0.0
34  0  0  0.294118  9  2.913284e-07  …  1.0  1.0  1.0  0.0  0.0
35  0  0  0.294118  9  3.389216e-07  …  1.0  1.0  1.0  0.0  0.0
36  0  0  0.294118  9  3.735349e-07  …  1.0  1.0  1.0  0.0  0.0
37  0  0  0.294118  9  4.341081e-07  …  1.0  1.0  1.0  0.0  0.0

[5 rows × 16 columns]
==========================================
训练集第3个子序列形状:  (64, 16)
训练集第3个子序列前5条记录:
    0  1         2  3             4  …   28   32   33   34   38
49  0  0  0.294118  9  4.571836e-07  …  1.0  1.0  1.0  0.0  0.0
50  0  0  0.294118  9  4.326659e-07  …  1.0  1.0  1.0  0.0  0.0
51  0  0  0.294118  9  4.427614e-07  …  1.0  1.0  1.0  0.0  0.0
52  0  0  0.294118  9  4.456459e-07  …  1.0  1.0  1.0  0.0  0.0
53  0  0  0.294118  9  4.413192e-07  …  1.0  1.0  1.0  0.0  0.0

[5 rows × 16 columns]
==========================================
```

图 9　KDD99-sub 训练集子序列划分

滑动窗口的大小与步长应根据被划分数据集规格及运算设备的性能来综合考虑。过大过密的子序列会对设备内存带来很大压力，而读取过小过密的子序列需要频繁进行内部通信；另外过于稀疏的子序列会影响模型对数据相关性的挖掘，关于滑动窗口及滑动步长的设定应根据实验结果调整优化确定。根据经验通常将滑动窗口的大小设置为 2 的指数幂，滑动步长一般可设为滑动窗口的 $\frac{1}{5}$ 到 $\frac{1}{3}$。

### 3.2　模型设置

VAE-WGAN 模型采用移除了池化层的卷积网络作为变分自编码器的编码器 $G_e$ 和解码器 $G_d$，自编码器输入层和解码器输出层的神经元个数与待检测数据集特征向量维度相同，隐藏层神经元数设置为输入层神经元数的一半。2 个 GAN 的判别器 $D_e$ 和 $D_d$ 为不含 sigmoid 层的卷积结构，超参数 Batch_size 为 256，最大 epoch 为 2 000，学习率 rate 为 0.000 1。

为验证 VAE-WGAN 模型检测的有效性，选取 2 种典型异常检测方法作为对比：MAD-GAN 算法和 TadGAN 算法。MAD-GAN 利用普通的 GAN 结构在潜在空间中以最优的搜索策略来支持多元时间序列重建，利用重构误差和判别误差作为目标函数进行训练和检测。TadGAN 利用 WGAN 组成的双生成对抗网络结构的时间序列异常检测模型，比本文所提的 VAE-WGAN 缺少 VAE 结构。

### 3.3　评估指标

在实际应用场景中，单纯的点异常很少出现，且对系统产生的影响基本可以忽略，因此本文主要针对连续的异常序列进行判断，并提出以下 3 个窗口规则。

1) 如果已知的异常窗口与任何预测窗口重叠，则记为 TP（真阳）。

2) 如果已知的异常窗口不与任何预测窗口重叠，则记为 FN（假阴）。

3) 如果预测窗口不与任何已知的异常区域重叠，则记为 FP（假阳）。

使用 Precision、Recall 和 F1 值等指标来衡量检测方法的性能，其数学表达式如式(10)所示。

$$\begin{cases} \text{Precision}=\dfrac{\text{TP}}{\text{TP+FP}} \\ \text{Recall}=\dfrac{\text{TP}}{\text{TP+FN}} \\ \text{F1值}=\dfrac{2\text{Precision}\times\text{Recall}}{\text{Precision+Recall}} \end{cases} \tag{10}$$

其中，Precision 为检测的精确率，表示计算检测到异常序列中有多少样本是真正的异常；Recall 为召回率，表示在原始实际序列的异常有多少样本被正确地识别出来；F1 值为精确率和召回率的调和平均数，兼顾分类模型的精确率和召回率。本文采用 F1 值作为衡量异常检测性能的主要指标。

### 3.4　实验结果与分析

#### 3.4.1　模型训练

VAE-WGAN 模型采取循环嵌套的方法进行训练，生成器和判别器交替训练，当生成器通过一次训练更新参数获得进步后，判别器需要利用正常数据重新训练几次以赶上生成器。算法 2 描述了每次迭代时生成器更新一次、判别器更新数次的训练过程。

**算法 2**　循环嵌套训练的伪代码

```
1) gen=inf_train_gen()
2) for i in range(Gen_Iters)//生成器训练次数
3)     if i>0:
4)         _=session.run(gen_train_op)
5)     end if
6)     for j in range(Dis_Iters)//判别器训练次数
7)         _data=gen._ _next_ _()
```

8)　　_disc_cost,_=session.run ([disc_cost, disc_train_op], feed_dict= {real_data: _data})

9)　end for

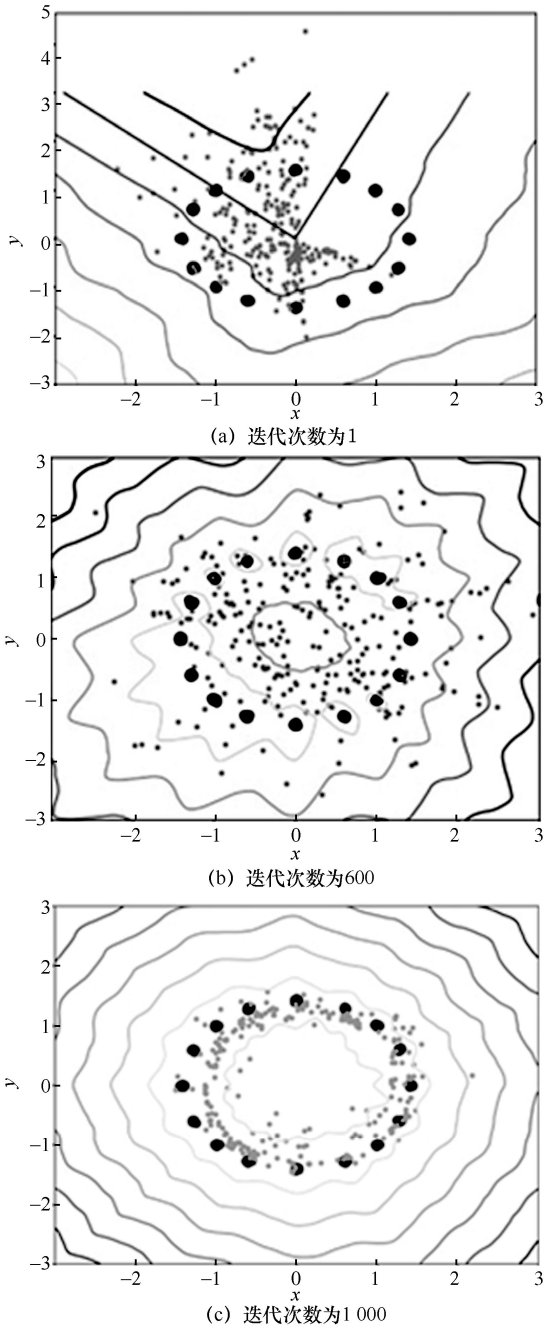
10) end for

图 10 展示了 VAE-WGAN 模型在 KDD99-sub 训练集上分别迭代 1 次、600 次及 1 000 次时生成样本与真实样本的空间状态。图 10 中，大圆点为真实样本 T，小圆点为生成样本 F，它们之间的距离表示生成样本与真实样本的差异，深色线条为高梯度线，浅色线条为低梯度线。生成器使损失函数值向极小值方向移动（浅色线条），而判别器则迫使其向极大值方向移动（深色线条）。经过数次迭代训练后，生成器学习到输入样本的真实分布，生成的样本逐渐向真实样本逼近，直到满足设定的收敛标准或者达到迭代次数，模型结束训练。进行异常检测时，也是通过计算输入真实样本与生成样本的“距离”是否超过阈值来判定输入数据中是否存在异常。

### 3.4.2　模型稳定性对比

在训练一个 epoch 后，固定判别器参数，将 TadGAN、MAD-GAN 和 VAE-WGAN 的生成器分别重新训练，图 11 展示了 3 个模型在 KDD99-sub 训练集上分别迭代 300 次和 1 000 次时损失函数的变化情况，横坐标为迭代次数，纵坐标为损失函数值。从图 11 可以看出，3 个模型在训练初期的损失函数值均较大且变化剧烈，但是经过数次迭代训练后，模型的损失值均明显减少，并保持在一定范围内振荡。总体来看，迭代 150 次后，VAE-WGAN 损失值率先降到最低，并且随着迭代次数的增加而逐渐趋于稳定，最先达到理想状态；TadGAN 和 MAD-GAN 的收敛速度相对较慢。迭代 200 次后，3 个模型的损失值都存在不同程度的增加，MAD-GAN 和 TadGAN 增加得比较明显，损失值均超过 1.5，可能是由于缺少自编器结构导致训练初期梯度更新不稳定。迭代 400 次后，TadGAN 和 MAD-GAN 的损失函数值开始出现振荡减小的趋势，但即使训练到 1 000 次，损失函数值仍然在较大范围内摆动。

值得注意的是，TadGAN 和 MAD-GAN 模型的振荡曲线相似，而 VAE-WGAN 的损失函数图像呈现“V”字形结构，训练初期表现平缓，其次突然近似线性地快速下跌，再次又近似线性地快速上升，再线性下跌再上升，下跌和上升始终围绕着“零点”进行且幅度逐渐减小。其原因可能是引入了权重裁剪和梯度惩罚项这 2 种机制共同约束参数的变化范围。因为训练初期判别器传递回的梯度更新信息变化较大，权重裁剪机制发挥主要作用导致变量参数集中分布在权重裁剪参数区间的两端，从而使运算结果呈现这种近似的线性关系。同样也是因为这种强约束机制使 VAE-WGAN 的损失函数值总在“零点”上下进行幅度递减的振荡。

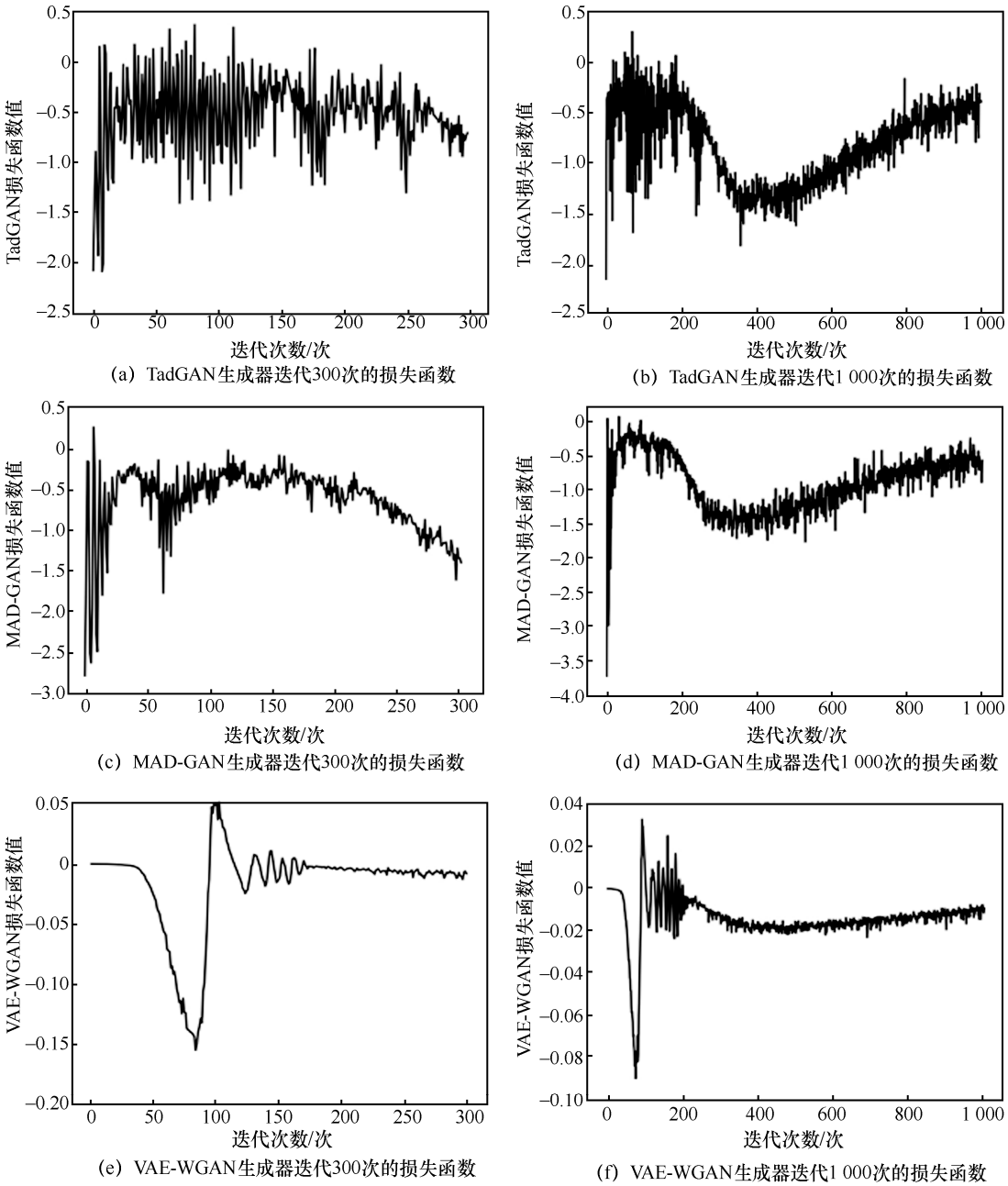



(a) 迭代次数为1

(b) 迭代次数为600

(c) 迭代次数为1 000

图 10　VAE-WGAN 在 KDD99-sub 数据集上的训练过程




(a) TadGAN生成器迭代300次的损失函数　(b) TadGAN生成器迭代1 000次的损失函数

(c) MAD-GAN生成器迭代300次的损失函数　(d) MAD-GAN生成器迭代1 000次的损失函数

(e) VAE-WGAN生成器迭代300次的损失函数　(f) VAE-WGAN生成器迭代1 000次的损失函数

图 11　3 个模型在 KDD99-sub 训练集上训练的损失函数

另外，VAE-WGAN 模型的损失函数曲线看似起伏较大，但相对于 TadGAN 和 MAD-GAN 模型来看，其损失函数值其实是非常小的，并且始终围绕“零点”振荡，最终能够快速达到理想的收敛效果。可以说与 TadGAN 和 MAD-GAN 模型相比，VAE-WGAN 模型具有明显的损失小、收敛快、训练容易的优势。

### 3.4.3　异常检测

异常检测是根据模型计算出的重构误差和判别误差，评估子序列的异常得分，并利用自适应阈值法计算对应的阈值，将大于阈值的子序列判定为异常。如图 12 所示，对前 72 个子序列进行初步的异常筛检，共检测出 6 个异常得分超过阈值的子序列 A、B、C、D、E、F，这 6 个子序列共组成 5 个异常序列，其中，A、B、E、F 子序列各自组成一个异常序列，C、D 这 2 个连续子序列组成一个异常序列。由此可见，每个异常序列的大小不定，可能只有一个子序列构成，也可能由多个连续的子序列组成。

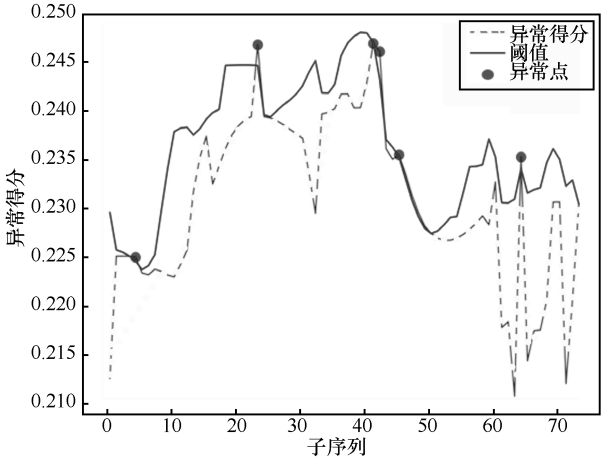



图 12　利用自适应阈值法筛检异常

这种利用滑动窗口划分时序子序列进行异常检测的方法可使每个样本出现在多个子序列中并被多次检测，有助于发现细微异常。但对于一些由噪声引起的偏差经过多次叠加放大，也可能被判别为异常，因此滑动窗口虽然可以提高异常的召回率，但也可能带来较高的误报率，因此需要对初步筛检到的异常结果进行取舍。本文采用了“去缓”的自适应异常修剪方法来减少误报[19]。具体地，提取各异常序列中子序列异常得分最大的值 $s_a^{\max\_i}$ 并按降序排列成一维数组 $S_a^{\max}=\{s_a^{\max\_1},s_a^{\max\_2},\cdots,s_a^{\max\_n}\}$，如图 13 所示，分别计算它们的下降速率 $p^i$，即

$$p^i=\frac{s_a^{\max\_i}-s_a^{\max\_i+1}}{s_a^{\max\_i}} \tag{11}$$

```
==========================================
各异常序列最大异常得分组成的一维数组:
[6.39527321 3.0638957   2.65504932 1.88564658 1.64663184 1.58686531
 1.57014203 1.55396664  1.53537798 1.53491271 1.50970483 1.5084095
 1.5056684  1.49328852  1.48411036 1.48336411 1.48024154 1.47927916
 1.46258616 1.45728815  1.4446857  1.43816912 1.43363023 1.42924666
 1.42514348 1.42142856  1.42045951 1.41605663 1.41569424 1.40932763
 1.40785849 1.40659261]
==========================================
```

图 13　最大异常得分数组

将验证集数据输入训练好的 VAE-WGAN 模型中，利用超参数搜索方法确定使验证集异常检测性能指标 F1 值达到最大时的裁剪速率阈值 $p$。将 $p^i$ 与设定的裁剪速率阈值 $p$ 进行比较，当第一个没有超过阈值 $p$ 的 $p^i$ 出现时，将它及其后续的最大异常得分 $\{s_a^{\max\_j},i\leqslant j\leqslant n\}$ 所对应的序列重新划归为正常。

图 14(a)为各序列中最大异常得分降序排列后的情况，图 14(b)为最大异常得分下降速率的情况。根据验证集实验结果，将裁剪速率阈值设置为 0.2。从图 14(b)中可以看出，速率超过裁剪阈值的有 A 和 C 共 2 个序列，但是由于 B 序列的异常得分下降速率已经降到阈值以下，因此 B 以及其后的所有序列（包括 C 序列）均重新归为正常。这种方法可直观地理解为异常值大且变化剧烈的序列是真正异常的可能性较大；而对于异常值较大但变化平缓的序列，可能是由于存在新的未被学习过的正常数据，导致模型不能很好地拟合，从而给出较大的异常得分，对于这种情况在异常裁剪时可以被重新归为正常。

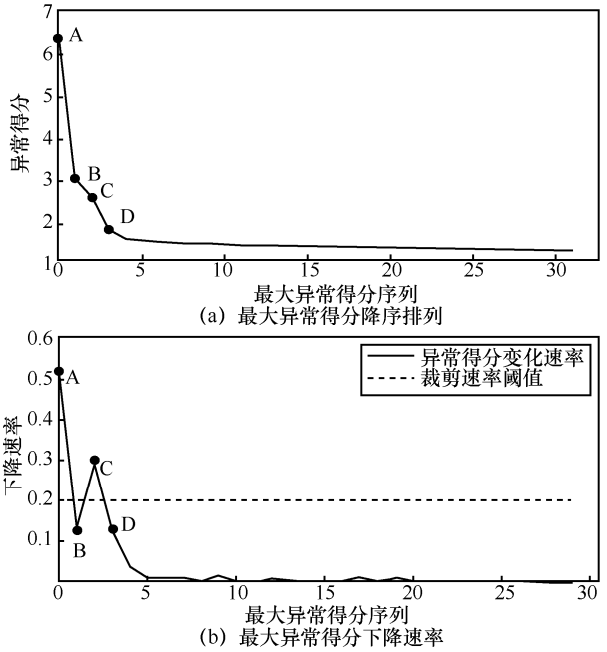



图 14　利用下降速率裁剪异常

### 3.4.4　异常检测结果

对比 TadGAN、MAD-GAN 和 VAE-WGAN 模型在公开数据集上的检测性能，比较结果如表 3 所示。由表 3 可知，VAE-WGAN 在 KDD99-sub、SMAP 和 MSL 这 3 个数据集上的精确率以及 F1 值均为最高，并且在 4 个数据集上的总评性能最好。MAD-GAN 在 4 个数据集上的 F1 值总评最低，可能是由于 MAD-GAN 为单生成对抗网络结构，相比于具有双生成对抗网络结构的 TadGAN 和 VAE-WGAN 模型，异常检测性能更差；MAD-GAN 在 SWaT 数据集的表现较好，这是由于 MAD-GAN 原本就是在 SWaT 数据集上经过了多次优化调整出来的模型。TadGAN 在 SMAP、MSL、SWaT 数据集上的表现较好，但在 KDD99-sub 数据集上

的检测性能较差，说明 TadGAN 不太适合捕捉网络流量中数据特征。虽然 VAE-WGAN 在 SWaT 数据集表现稍弱于 MAD-GAN，但在其他 3 个数据集上表现更加突出，说明 VAE-WGAN 不仅具有出色的检测性能，对于未知异构时间序列数据也具有较强的适用性和泛化能力。

**表 3　　各异常检测模型性能比较**

| 模型 | 数据集 | 精确率 | 召回率 | F1 值 |
|---|---|---|---|---|
| TadGAN | KDD99-sub | 0.376 7 | 0.156 3 | 0.220 9 |
| | SMAP | 0.668 6 | 0.533 4 | 0.593 4 |
| | MSL | 0.613 7 | 0.666 9 | 0.639 2 |
| | SWaT | 0.631 6 | 0.468 7 | 0.538 1 |
| | 总评 | 0.614 0 | 0.475 2 | 0.524 4 |
| MAD-GAN | KDD99-sub | 0.236 7 | 0.442 9 | 0.308 5 |
| | SMAP | 0.263 8 | 0.283 7 | 0.273 4 |
| | MSL | 0.567 9 | 0.354 3 | 0.436 4 |
| | SWaT | 0.897 8 | 0.723 2 | 0.801 1 |
| | 总评 | 0.450 2 | 0.432 2 | 0.428 3 |
| VAE-WGAN | KDD99-sub | 0.784 6 | 0.569 4 | 0.659 9 |
| | SMAP | 0.701 2 | 0.804 5 | 0.749 3 |
| | MSL | 0.741 2 | 0.609 9 | 0.669 2 |
| | SWaT | 0.823 1 | 0.686 4 | 0.748 6 |
| | 总评 | 0.762 5 | 0.667 6 | 0.706 7 |

## 4　结束语

本文提出了基于 VAE-WGAN 架构的多维时间序列异常检测模型，利用 VAE 作为 WGAN 的生成器，解决传统生成对抗网络缺少自编码器训练困难的问题；利用 WGAN 的判别器指导 VAE 调整训练参数，增加自编码器训练的稳健性有利于减少过拟合。使用 Wasserstein 距离作为模型拟合分布与输入样本数据真实分布的差异性度量，可有效地避免传统方法利用 $f$ 散度训练时存在的“梯度失稳”和“模式崩溃”风险；利用 VAE 的概率编码器模拟潜在变量分布而非输入变量本身，解决了多维异构数据难以统一计算的问题，增强了模型对各时间序列数据的泛化能力。利用滑动窗口将时间序列划分为多个子序列，有利于发现序列中上下文异常；使用滑动窗口自适应的阈值设定方法，有助于提升异常序列的召回率；使用“去缓”的裁剪技术，对异常得分变化小的时间子序列做“纠正”处理，提升了异常判定的准确率。

通过对比 VAE-WGAN、TadGAN 和 MAD-GAN 这 3 种异常检测模型在 KDD99-sub、SMAP、MSL 和 SWaT 这 4 个时间序列数据集上的异常检测性能可知，基于 VAE-WGAN 的多维时间序列异常检测模型能够提供稳定的梯度信息，有较强的稳定性和稳健性，且对多维异构数据有较强的泛化能力，尤其是针对网络流量数据的检测性能明显高于其他模型。如何与安全事件关联分析技术相结合，将本文方法拓展到基于网络流量的入侵检测系统是下一步的研究方向。

## 参考文献：


[1] AHMAD S, LAVIN A, PURDY S, et al. Unsupervised real-time anomaly detection for streaming data[J]. Neurocomputing, 2017, 262: 134-147.

[2] KINGMA D P, WELLING M. Auto-encoding variational Bayes[J]. arXiv Preprint, arXiv: 1312.6114, 2013.

[3] GOODFELLOW I J, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial nets[C]//Proceedings of the 27th International Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2014: 2672-2680.

[4] CHANDOLA V, BANERJEE A, KUMAR V. Anomaly detection: a survey[J]. ACM computing surveys (CSUR), 2009, 41(3): 1-58.

[5] AN J, CHO S. Variational autoencoder based anomaly detection using reconstruction probability[J]. SUN Data Mining Center, 2015, 2: 1-18.

[6] XU H W, CHEN W X, ZHAO N W, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications[C]//Proceedings of the 2018 World Wide Web Conference. Republic and Canton of Geneva: International World Wide Web Conferences Steering Committee, 2018: 187-196.

[7] CHEN W X, XU H W, LI Z Y, et al. Unsupervised anomaly detection for intricate KPIs via adversarial training of VAE[C]//Proceedings of IEEE INFOCOM 2019 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2019: 1891-1899.

[8] POL A A, BERGER V, GERMAIN C, et al. Anomaly detection with conditional variational autoencoders[C]//Proceedings of 2019 18th IEEE International Conference on Machine Learning and Applications. Piscataway: IEEE Press, 2019: 1651-1657.

[9] 杨宏宇, 王峰岩, 吕伟力. 基于无监督生成推理的网络安全威胁态势评估方法[J]. 清华大学学报(自然科学版), 2020, 60(6): 474-484.
YANG H Y, WANG F Y, LYU W L. Network security threat assessment method based on unsupervised generation reasoning[J]. Journal of Tsinghua University (Science and Technology), 2020, 60(6): 474-484.

[10] MEMARZADEH M, MATTHEWS B, AVREKH I. Unsupervised anomaly detection in flight data using convolutional variational auto-encoder[J]. Aerospace, 2020, 7(8): 115.

[11] 郭敏, 曾颖明, 于然, 等. 基于对抗训练和 VAE 样本修复的对抗攻击防御技术研究[J]. 信息网络安全, 2019(9): 66-70.
GUO M, ZENG Y M, YU R, et al. Research on defense technology of adversarial attacks based on adversarial training and VAE-repairing[J]. Netinfo Security, 2019(9): 66-70.

[12] BEULA RANI B J, SUMATHI M E L. Survey on applying GAN for anomaly detection[C]//Proceedings of 2020 International Conference on Computer Communication and Informatics (ICCCI). Piscataway: IEEE Press, 2020: 1-5.

[13] THOMAS S, PHILIPP S, SEBASTIAN M W. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery[C]//Proceedings of Information Processing in Medical Imaging. Berlin: Springer, 2017, 39(2): 1703-1721.

[14] BASHAR M A, NAYAK R. TAnoGAN: time series anomaly detection with generative adversarial networks[C]//Proceedings of 2020 IEEE Symposium Series on Computational Intelligence. Piscataway: IEEE Press, 2020: 1778-1785.

[15] LI D, CHEN D C, JIN B H, et al. MAD-GAN: multivariate anomaly detection for time series data with generative adversarial networks[C]//Artificial Neural Networks and Machine Learning - ICANN 2019. Berlin: Springer, 2019: 214-232.

[16] CHEN X H, DENG L W, HUANG F T, et al. DAEMON: unsupervised anomaly detection and interpretation for multivariate time series[C]//Proceedings of 2021 IEEE 37th International Conference on Data Engineering. Piscataway: IEEE Press, 2021: 2225-2230.

[17] 戴文倩. 基于生成对抗网络的流量预测研究[D]. 上海: 上海师范大学, 2019.
DAI W Q. Traffic prediction method for network based on generative adversarial network[D]. Shanghai: Shanghai Normal University, 2019.

[18] 潘一鸣, 林家骏. 基于生成对抗网络的恶意网络流生成及验证[J]. 华东理工大学学报(自然科学版), 2019, 45(2): 344-350.
PAN Y M, LIN J J. Generation and verification of malicious network flow based on generative adversarial networks[J]. Journal of East China University of Science and Technology, 2019, 45(2): 344-350.

[19] FERDOWSI A, SAAD W. Generative adversarial networks for distributed intrusion detection in the Internet of things[C]//Proceedings of 2019 IEEE Global Communications Conference. Piscataway: IEEE Press, 2019: 1-6.

[20] 刘芃. 基于 Text-GAN 的加密流量识别关键技术研究[D]. 南京: 南京邮电大学, 2020.
LIU P. Research on encrypted traffic identification based on Text-Gan[D]. Nanjing: Nanjing University of Posts and Telecommunications, 2020.

[21] 邹福泰, 谭越, 王林, 等. 基于生成对抗网络的僵尸网络检测[J]. 通信学报, 2021, 42(7): 95-106.
ZOU F T, TAN Y, WANG L, et al. Botnet detection based on generative adversarial network[J]. Journal on Communications, 2021, 42(7): 95-106.

[22] CHEN N G, LI C M. Hyperspectral image classification approach based on Wasserstein generative adversarial networks[C]//Proceedings of 2020 International Conference on Machine Learning and Cybernetics (ICMLC). Piscataway: IEEE Press, 2020: 53-63.

[23] WEI X, GONG B Q, LIU Z X, et al. Improving the improved training of Wasserstein GANs: a consistency term and its dual effect[J]. arXiv Preprint, arXiv:1803.01541, 2018.

[24] GEIGER A, LIU D Y, ALNEGHEIMISH S, et al. TadGAN: time series anomaly detection using generative adversarial networks[C]//Proceedings of 2020 IEEE International Conference on Big Data (Big Data). Piscataway: IEEE Press, 2020: 33-43.

[25] 王星, 霍纬纲. Wasserstein 自编码器异常检测模型[J]. 计算机工程与设计, 2020, 41(11): 3249-3254.
WANG X, HUO W G. Wasserstein autoencoder anomaly detection models[J]. Computer Engineering and Design, 2020, 41(11): 3249-3254.

[26] LARSEN A B L, SØNDERBY S K, WINTHER O, et al. Autoencoding beyond pixels using a learned similarity metric[C]//Proceedings of the 33rd International Conference on Machine Learning. [S.l.]: JMLR, 2016:1558-1566.

[27] NIU Z J, YU K, WU X F. LSTM-based VAE-GAN for time-series anomaly detection[J]. Sensors (Basel, Switzerland), 2020, 20(13): 3738.

[28] 张璇, 程敏熙, 肖凤平. 利用 Origin 对数据异常值的剔除方法进行比较[J]. 实验科学与技术, 2012, 10(1): 74-76, 118.
ZHANG X, CHENG M X, XIAO F P. Origin used in comparison the methods of eliminating the excrescent data[J]. Experiment Science and Technology, 2012, 10(1): 74-76, 118.

[作者简介]

**段雪源**（1981− ），男，河南开封人，海军工程大学博士生，主要研究方向为人工智能、信息处理、网络安全。

**付钰**（1982− ），女，湖北武汉人，博士，海军工程大学教授、博士生导师，主要研究方向为信息安全、人工智能。

**王坤**（1981− ），女，河南信阳人，海军工程大学博士生，主要研究方向为信息安全。