

UMTS-Mixer: 基于时间相关性和通道相关性的 时间序列异常检测^①



孙怡阳^{1,2}, 陈志德^{1,2}, 冯晨³, 朱可欣⁴

¹(福建师范大学 计算机与网络空间安全学院, 福州 350117)

²(福建省网络安全与密码技术重点实验室 (福建师范大学), 福州 350117)

³(福州职业技术学院 信息工程系, 福州 350108)

⁴(台湾中山大学 平行处理实验室, 高雄 510275)

通信作者: 陈志德, E-mail: zhidechen@fjnu.edu.cn

摘要: 多变量时间序列的异常检测是一个具有挑战性的问题, 要求模型从复杂的时间动态中学习信息表示, 并推导出一个可区分的标准, 该标准能从大量正常时间点识别出少量的异常点. 但在时间序列分析中仍存在多变量时间序列复杂的时间相关性和高维度使得异常检测性能较差的问题, 针对上述问题, 本文提出了一种基于 MLP (multi-layer perceptron) 架构的模型 (UMTS-Mixer), 由于 MLP 的线性结构对顺序敏感, 将其用来捕获时间相关性和跨通道相关性. 大量实验表明 UMTS-Mixer 能够有效地检测时间序列异常, 并在 4 个基准数据集上的表现更好, 同时, 在 MSL 和 PSM 两个数据集上取得了最高的 $F1$, 分别为 91.35%, 92.93%.

关键词: 异常检测; 时间相关性; 通道相关性; MLP

引用格式: 孙怡阳, 陈志德, 冯晨, 朱可欣. UMTS-Mixer: 基于时间相关性和通道相关性的时间序列异常检测. 计算机系统应用, 2024, 33(1): 127-133. <http://www.c-s-a.org.cn/1003-3254/9347.html>

UMTS-Mixer: Time Series Anomaly Detection Based on Temporal Correlation and Channel Correlation

SUN Yi-Yang^{1,2}, CHEN Zhi-De^{1,2}, FENG Chen³, ZHU Ke-Xin⁴

¹(College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China)

²(Fujian Provincial Key Lab of Network Security & Cryptology (Fujian Normal University), Fuzhou 350117, China)

³(Department of Information Technology Engineering, Fuzhou Polytechnic, Fuzhou 350108, China)

⁴(Parallel Processing Lab, Taiwan Sun Yat-sen University, Kaohsiung 510275, China)

Abstract: Anomaly detection in multivariate time series is a challenging problem that requires models to learn information representations from complex temporal dynamics and derive a distinguishable criterion that can identify a small number of outliers from a large number of normal time points. However, in time series analysis, the complex temporal correlation and high dimensionality of multivariate time series will result in poor anomaly detection performance. To this end, this study proposes a model based on MLP (multi-layer perceptron) architecture (UMTS-Mixer). Since the linear structure of MLP is sensitive to order, it is employed to capture temporal correlation and cross-channel correlation. A large number of experiments show that UMTS-Mixer can detect time series anomalies and perform better on the four benchmark datasets. Meanwhile, the highest $F1$ is 91.35% and 92.93% on the MSL and PSM datasets, respectively.

Key words: anomaly detection; temporal correlation; channel correlation; multi-layer perceptron (MLP)

① 基金项目: 国家自然科学基金 (62277010); 福建省自然科学基金 (2020J01132452); 福厦泉国家自主创新示范区协同创新平台项目 (2022FX6)

收稿时间: 2023-06-28; 修改时间: 2023-07-27; 采用时间: 2023-08-08; csa 在线出版时间: 2023-10-27

CNKI 网络首发时间: 2023-10-31

1 引言

现实生活中无处不存在数据,无时无刻也不在产生数据,而这些随时间变化的连续数据就是时序数据,如金融数据,天气数据,交通数据,运动数据等.这些时序数据中通常会存在异常数据,在金融数据中,异常数据可能对应着欺诈事件的发生,在工业数据中,异常数据也可能对应着机器故障的发生,因此,对时序数据中的异常数据进行检测以避免带损失就极具意义.但是异常通常是不常见,并且隐藏在大量的正常数据中,这也使得数据标记变得困难且昂贵,因此,现在的大量工作都是基于无监督的时间序列异常检测,本工作也是基于此.

无监督时间序列异常检测是一个有挑战性的工作,在这个复杂的工作面前,传统的机器学习算法很难取得好的表现,于是越来越多的深度学习方法出现,以期好的完成这个工作.无监督时间序列异常检测工作的难点在于训练数据中包含正常数据和异常数据,对模型学习数据的特征信息会产生不利影响,同时,还需要模型习得一个好的判据,以此来区分正常数据和异常数据.因此,上述两个问题就是解决无监督时间序列异常检测的重点,也是难点所在.由于时间序列具有时间模式的特性且 RNN 具有序列建模的能力,很自然地就想到使用 RNN 建模,但 RNN 的局限在于模型更注重的是当前时间点的数据信息和前一时间点的数据信息的关系,而当前时间点的数据信息并非只与前一时间点的数据信息有关,往往与前一段时间点的数据信息有关,这就限制了 RNN 的特征表示能力.随着近年来 Transformer 模型^[1]在计算机视觉和自然语言处理等领域的不错表现,许多学者开始尝试将 Transformer 模型运用在时间序列数据中, CAT^[2]、DCT-GAN^[3]、anomaly Transformer^[4]、TranAD^[5]等工作对 Transformer 模型针对时间序列的特性做了各种改进,提升了其计算性能,也得到了好的表现.但最近 Zeng 等人^[6]的工作质疑了基于 Transformer 的模型对时间序列建模的有效性,其实验表明 Transformer 中的核心模块 self-attention 在时序数据建模上并没有取得作用. self-attention 模块是为了提取长序列的语义信息,其特征提取是没有考虑顺序的,但时序建模中顺序是一个重要因素.尽管使用 Transformer 架构的模型增加了 positional embedding 和 temporal embedding 模块弥补时序的顺序特殊性,但其实验得出的结论是这些模块并没有保留时序数据的顺序.与此相反,线性模型体现出了对顺序关系的极度敏感,调换输入数据的顺序会对结果产生非常大的变

化,同时,线性模型也较好地保留了趋势性和周期性.基于 Zeng 等人^[6]的工作, MTS-Mixers^[7]、TSMixer^[8]基于线性结构的模型在时间序列预测任务上取得优于基于 Transformer 模型的效果,也验证了线性结构在时间序列数据上的有效性.鉴于此,本文提出了 UMTS-Mixer 模型,其中并没有用到 RNN 和 self-attention 结构,而只用到了线性层.通过线性层的堆叠,先缩放后扩展(类似编解码器的结构),在数据的时间相关上学习特征表示;然后依旧是线性层的堆叠,先缩放后扩展,在数据的通道相关上学习特征表示,使模型从两个维度学习,得到好的信息表征.

本文的贡献如下:(1) 本文提出了基于时间相关性和通道相关性的时间序列异常检测方法,通过对顺序敏感的线性模型来习得数据在时间和通道两个维度上信息表征,使正常数据区别于异常数据,更容易为模型所表示.(2) 对 4 个数据集进行了实验,结果表明本方案与基线方法相比,提升了对异常的检测准确率,在 F1 值和召回率两个异常检测指标上优于基线方法.

本文第 2 节介绍了相关工作.第 3 节介绍本文提出的基于时间相关性和通道相关性的时间序列异常检测方法.第 4 节在 4 个现实数据集上评估本文的方法.第 5 节总结本文的工作.

2 相关工作

无监督多变量时间序列异常检测作为一个重要的现实问题,得到了广泛的关注.有大量的基于神经网络的方法被提出,这些方法大致可以分为两类,一类是基于自回归的方法,另一类则是基于重构的方法.

基于自回归的方法,通常将过去时间段的数据作为模型的输入,模型产生的输出与当前时间点的数据的差值作为预测误差,通过预测误差检测异常.基于自回归的方法有时序预测模型 ARIMA^[9],该方法需要将时间序列转换为平稳、突变较少的序列,使用前一时间段内的样本点来预测下一时刻的样本输入值. VAR^[10]扩展了 ARIMA,并根据滞后相关的协方差预测未来.自回归模型也可以替换为 LSTM^[11].

基于重建的方法,通常将当前时间点的数据作为模型的输入,模型对数据降维,然后还原到原始维度重建数据,模型输入和输出之间的差值作为重建误差,通过重建误差检测异常.基于重建方法的模型一般通过编解码器的结构来实现,自编码器也被称为自相关神经网络,具有非线性降维能力.为了捕获更多的信息并

能更好的表示信息,对传统的自编码器做了相应特性的改变,如稀疏自动编码器^[12]、去噪自动编码器^[13]、卷积自动编码器^[14].VAE (variational autoencoder)^[15]将神经网络自动编码器与 mean-field variational Bayes 相结合.Park 等人提出了 LSTM-VAE 模型^[16],该模型替换了原始 VAE 中的 FFN,采用 LSTM 主干进行时间建模,LSTM 使用去噪自动编码进行训练以提高表示能力.Su 等人提出的 OmniAnomaly^[17]进一步扩展了具有归一化流的 LSTM-VAE 模型,并使用重构概率进行检测.随着 Transformer 兴起,将 Transformer 中的结构引入时序数据的建模中,提出了 anomaly Transformer,同时使用高斯核学习先验关联,用自注意力学习序列关联,并用 MINMAX 方法突出关联差异,但该方法未考虑变量之间的关系.TranAD 是另一个基于 Transformer 的异常检测模型,它具有自调节和对抗训练,在处理长序列时,训练和测试效率很高,同时保持稳定.虽然已经有部分基于 Transformer 的方法取得了不错的表现,但其复杂的架构和其核心模块在顺序结构建模的无效性,不得不重新对其进行审视.

因此,针对上述问题,本文通过采用线性模型对数据的时间维度和通道维度建模,从而使得模型对正常数据有强大的表征能力,区别于异常数据的难以表征,以此来进行正常数据和异常数据的区分,达到异常检测的目的.

3 模型构造

3.1 问题描述

给定一个多变量时间序列数据的实例,表示为 $X_t = [x_1, x_2, x_3, \dots, x_n] \in \mathbb{R}^{n \times c}$,其中 n 为时间长度, c 为变量长度,多变量时间序列异常检测的目标就是通过学习过去时间的数据特征信息,从而能识别出当前时刻的数据是正常数据还是异常数据,表示为 $y_{cur} = f_{AD}(x_{cur}) \in \{0, 1\}$,其中 x_{cur} 为当前时刻的数据, f_{AD} 为异常检测方

法, y_{cur} 为当前时刻数据的正异常标签,0 表示当前时刻为正常数据,1 则表示为异常数据.由于是基于无监督任务,所以训练数据中不存在数据的正异常标签,本文通过重建误差来使模型学习数据的特征信息,然后用训练好的模型在测试数据集给出正异常的判断,最后通过模型给出的正异常判断和测试数据集中的正异常标签进行对比,以此来评估模型.

3.2 UMTS-Mixer 模型

Zeng 等人^[6]通过实验证明了线性模型能更好地捕获时间依赖关系,因此本文提出了一种线性结构模型 UMTS-Mixer 以实现高效的时间序列的异常检测.模型的整体架构如图 1 所示.

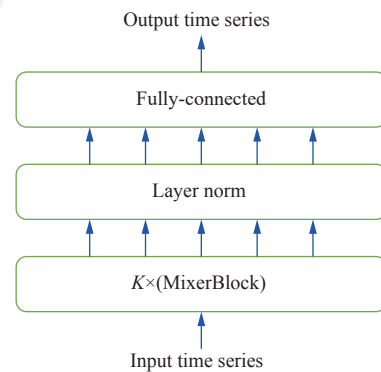


图 1 UMTS-Mixer 整体架构

UMTS-Mixer 模型主要由 MixerBlock 模块构成,MixerBlock 模块包含两个子模块分别为 time-mixer 和 channel-mixer,其中 time-mixer 模块用来学习时序数据中的时间依赖关系,channel-mixer 模块用来学习时序数据中变量之间的依赖关系,而 time-mixer 和 channel-mixer 模块组成核心是 MLP,layer norm 层对数据进行归一化处理,time-mixer 与 channel-mixer 的衔接中需要完成 T 转置操作,使得模型可以交替进行对时间依赖和变量依赖学习,MixerBlock 模块的结构如图 2.

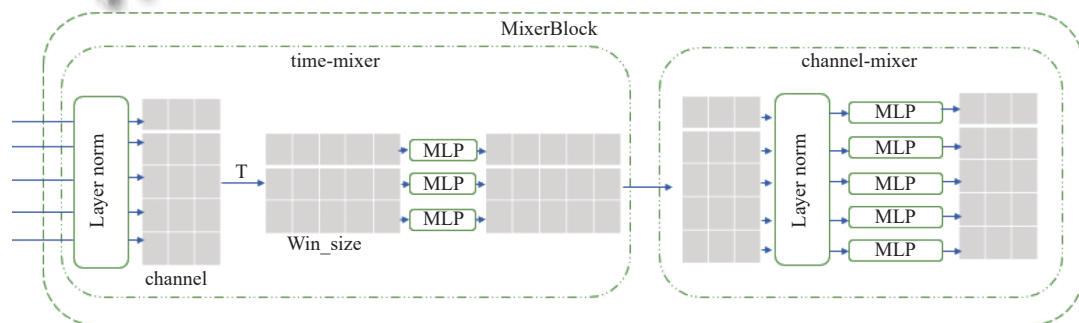


图 2 MixerBlock 模块结构图

time-mixer 与 channel-mixer 中最基本的组成单元 MLP 的构成如图 3 所示. MLP 其中包含了 fully-connected 层, GELU 激活层, dropout 层, 最后由一个 fully-connected 层结束. 训练模式下时间序列重建的过程如下:

$$\begin{cases} X_{in}^T = MLP(rearrange(norm(X_{in}))) + X_{in} \\ X_{in}^C = MLP(norm(rearrange(X_{in}^T))) + X_{in}^T \\ X_{out} = Linear(norm(X_{in}^C)) \end{cases} \quad (1)$$

其中, X_{in} 表示输入模型的时间序列数据, $norm$ 表示 layer norm 层, $rearrange$ 表示时间维度和变量维度的一个转换, X_{in}^T 表示学习时间依赖关系后所得到的数据, X_{in}^C 表示学习变量之间依赖关系后得到的数据, 最后通过一个线性层映射到输入维度, 得到输出 X_{out} . 通过重建误差, 使 $loss$ 值降低, 更新模型参数, $loss$ 函数如式 (2) 所示. 训练过程如算法 1 所示.

$$loss = \|X_{in} - X_{out}\|_F^2 \quad (2)$$

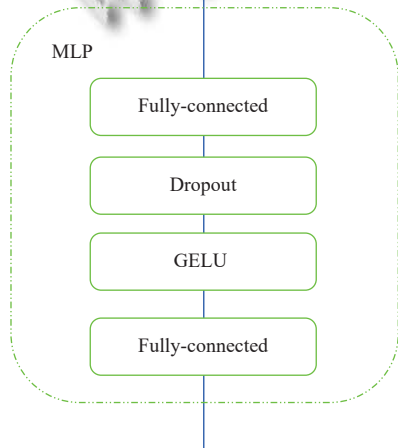


图 3 MLP 构成图

训练模型, 算法流程如算法 1.

算法 1. 训练算法

Input: 输入样本 X_{in} .
Output: 输出值 X_{rec} .

1. for $i = 1$ to epoch do
2. for $j = 1$ to batch do
3. for $k = 1$ to K do
4. $X^k = MixerBlock(X^{k-1})$
5. end for
6. $Z^k = norm(X^k)$
7. $X_{rec} = Linear(Z^k)$
8. 更新模型参数
9. end for
10. end for

算法 1 中 epoch 表示模型迭代训练次数, 每个 epoch 将数据划分为等份的若干个 batch, 1 个 batch 数据通过计算 $loss$ 值和优化器更新 1 遍模型参数, K 表示 UMTS-Mixer 模型中有 K 个 MixerBlock 模块.

测试模型, 异常检测算法流程如算法 2.

算法 2. 测试算法

Input: 训练数据 X_{train} , 验证数据 X_{val} , 测试数据 X_{test} .

1. for $i = 1$ to batch do
2. $pred_{train} = UMTS-Mixer(X_{train})$
3. $loss_{train} = pred_{train} - X_{train}$
4. end for
5. for $i = 1$ to batch do
6. $pred_{val} = UMTS-Mixer(X_{val})$
7. $loss_{val} = pred_{val} - X_{val}$
8. end for
9. $thresh = np.percentile(loss_{train}, loss_{val}, 99)$
10. for $i = 1$ to batch do
11. $pred_{test} = UMTS-Mixer(X_{test})$
12. $loss_{test} = pred_{test} - X_{test}$
13. if $loss_{test} > thresh$
14. 判定为异常
15. else
16. 判定为正常
17. end for

算法 2 中 X_{train} 表示训练数据, X_{val} 表示验证数据, X_{test} 表示测试数据, 先将训练数据输入模型, 得到模型重建的训练数据, 然后计算出训练数据和重建数据之间的误差值, 验证数据重复训练数据的操作, 得到验证数据和重建数据之间的误差值, 将得到的训练数据的误差值和验证数据的误差值按比例划分, $thresh$ 为误差值最高 1% 与剩下 99% 的分界线, 最后计算测试数据的误差值, 将误差值高于 $thresh$ 的数据判定为异常数据, 误差值低于或等于 $thresh$ 的数据判定为正常数据.

$$thresh < \|pred_{test} - X_{test}\|_F^2 \quad (3)$$

$$thresh \geq \|pred_{test} - X_{test}\|_F^2 \quad (4)$$

4 实验

本文在 4 个基准数据集上评估 UMTS-Mixer 的异常识别效果. 4 个数据集分别是 SMD、PSM^[18]、MSL、SMAP, 以下是对 4 个实验数据集的描述: (1) SMD 是从一家大型互联网公司收集的服务器数据, 具有 38 个通道, 异常占比为 4.2%. (2) PSM 是从 eBay 的多个应用服务器节点内部收集的数据信息, 具有 26 个通道, 异常占比为 27.8%. (3) MSL 和 SMAP 都是来自 NASA^[11]

的公共数据集, 分别有 55 个通道和 25 个通道, 异常占比分别为 10.5% 和 12.8%。其中包含从航天器监控系统的事件报告得出的遥测异常数据。数据集具体信息表 1 所示。

表 1 4 个数据集具体信息

数据集	通道数	训练样本	验证样本	测试样本	异常占比
MSL	55	46653	11664	73729	0.105
SMD	38	566724	141681	708420	0.042
PSM	25	105984	26497	87841	0.278
SMAP	25	108146	27037	427617	0.128

4.1 实现细节

本文采用滑动窗口来获得一组子序列。对于 SMD 数据集, 滑动窗口的大小为 200, 滑动步长为 100, 其他 3 个数据集滑动窗口大小为 100, 滑动步长为 1。阈值 *thresh* 被确定为使标记为异常的验证数据集的 *r* 比例数据。对于最终结果, 本文设置 SMD 的 *r* = 0.5%, 其他数据集设置为 1%。训练模型时, 用到了提前停止训练方法, 当连续 3 个 epoch 的验证集 *loss* 没有降低时, 则停止训练, 并将 *loss* 值最低的模型作为最终的模型, 同时, 每个 epoch 的学习率会根据 epoch 的变化而变化。在异常检测时, 使用调整策略^[19], 如果检测到某个连续异常段中的时间点, 则认为该异常段中的所有异常被准确检测。这种策略是合理的, 因为一个异常时间点会导致警报, 将使整个时间段被注意到^[20]。UMTS-Mixer 包含 3 个 MixerBlock 模块, 本文将隐藏状态 *dmodel* 的通道数设置为 512, 使用初始学习率为 10^{-4} 的 Adam 优化器^[21], 批量大小为 64。所有实验均以 Python 3.8 作为编程基础, PyTorch 1.10.0、cuda 11.1 实现和测试。使用 Windows 10、Intel(R) Core(TM) i7-10750H CPU、RTX3060、16 GB 内存的计算机运行程序。

4.2 对比实验

本文将模型与 9 个基线进行比较, 包括以下模型: anomaly Transformer (2022)、TranAD (2022)、InterFusion (2021)、BeatGAN (2019)、OmniAnomaly (2019)、LSTM-VAE (2018)、THOC (2020)、LSTM (2018)、VAR (1976); 其中 InterFusion (2021)、THOC (2020) 是当年发表的最先进的深度模型。

4.3 评估指标

通过准确率 (*P*)、召回率 (*R*) 和 *F1* 值来评估一个异常检测算法的好坏, *TP* 表示算法将异常样本识别出来的个数, *FN* 表示算法将异常样本识别为正常样本的

个数, *FP* 表示算法将正常样本识别成异常样本的个数。根据公式理解, *P* 表示算法识别为异常的样本中有多少比例是识别正确的, *P* 越接近 1 则说明算法误报越少, *P* 越接近 0 说明算法误报越多, *R* 表示算法在所有的异常样本中有多少比例是将异常样本识别为异常的, 也即算法识别出异常样本的比例, *R* 越接近 1 则说明算法漏报越少, *R* 越接近 0 则说明算法漏报越多, *F1* 值则是 *P* 和 *R* 综合考虑, 权衡出的准则, *F1* 越接近 1 说明算法的效果越好。

$$\begin{cases} P = \frac{TP}{TP+FP} \\ R = \frac{TP}{TP+FN} \\ F1 = \frac{2P \times R}{P+R} \end{cases} \quad (5)$$

4.4 结果分析

为了衡量本文模型 UMTS-Mixer 的有效性, 模型在 4 个数据集上做了检验, 并与 9 个基准模型进行对比, 对比结果如表 2、图 4、图 5 所示。表 2 中, 其中加粗表示该项的最大值。由表 2 可知, 本文提出的模型在 MSL、PSM 这两个数据集上, *F1* 值为最高, 取得了最好的表现, 在 MSL、SMAP 数据集中, 模型的准确率也是对对比模型中最高的, 在 SMD、SMAP 数据集中, 模型的 *F1* 值分别位于对比模型的第 3 位和第 4 位。在 MSL、SMD 这两个数据集上, 本模型取得的准确率和召回率都是比较接近的, 说明模型的 *thresh* 取值是较为准确的, 但模型在 SMAP、PSM 这两个数据集中准确率和召回率相差较远, 说明模型没有找到更好的 *thresh* 分界线, 若准确率高于召回率 (相差 10% 左右) 说明模型更倾向于判定为异常, 若准确率低于召回率 (相差 10% 左右) 说明模型更倾向于判定为正常。从图 4 也能更直观地看出本文提出的模型要优于基线模型。图 6 给出了 PSM 数据集时间序列、已知的异常数据点、模型判定的异常数据点, 其中 *feature_0* 表示时间序列数据, Model-pred 表示模型判定的异常数据点, Known-label 表示已知的异常数据点, Known-label 与 Model-pred 的匹配度越高, 说明模型的异常检测效果越好。Known-label 与 Model-pred 不匹配的时间点, 说明模型在该时间点进行了误判或漏判, Model-pred 不在 Known-label 区域代表误判, Known-label 区域中是 *feature_0* 代表漏判。从图 6 中的匹配度可以得出模型在 PSM 数据集上异常检测的表现是优异的。

表2 本文提出的模型与9个基线在4个数据集上的性能对比(%)

数据集	MSL			SMD			SMAP			PSM		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
VAR	74.68	81.42	77.90	78.35	70.26	74.08	81.38	53.88	64.83	90.71	83.82	87.13
LSTM	85.45	82.50	83.95	78.55	85.28	81.78	89.41	78.13	83.39	76.93	89.64	82.80
LSTM-VAE	85.49	79.94	82.62	75.76	90.08	82.30	92.20	67.75	78.10	73.62	89.92	80.96
BeatGAN	89.75	85.42	87.53	72.90	84.09	78.10	92.38	55.85	69.61	90.30	93.84	92.04
OmniAnomaly	89.02	86.37	87.67	83.68	86.82	85.22	92.49	81.99	86.92	88.39	74.46	80.83
InterFusion	81.28	92.70	86.62	87.02	85.43	86.22	89.77	88.52	89.14	83.61	83.45	83.52
THOC	88.45	90.97	89.69	79.76	90.95	84.99	92.06	89.34	90.68	88.14	90.99	89.54
anomaly Transformer	90.97	88.45	89.69	92.33	85.26	88.65	91.05	80.99	85.72	98.73	83.21	90.31
TranAD	86.37	89.02	87.67	86.16	83.83	84.98	89.32	68.86	77.77	85.31	94.22	89.54
Ours	91.15	91.55	91.35	86.81	88.16	87.48	95.91	69.98	80.92	97.63	88.66	92.93

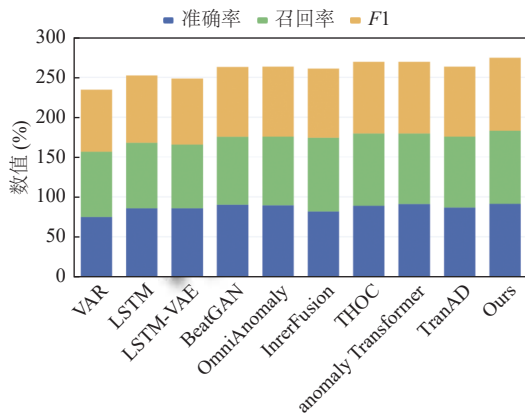


图4 各个模型在MSL数据集上3个评估指标的矩形堆叠图

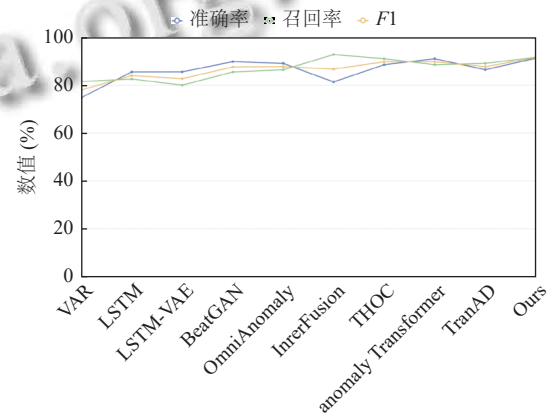


图5 各个模型在MSL数据集上的准确率,召回率,F1

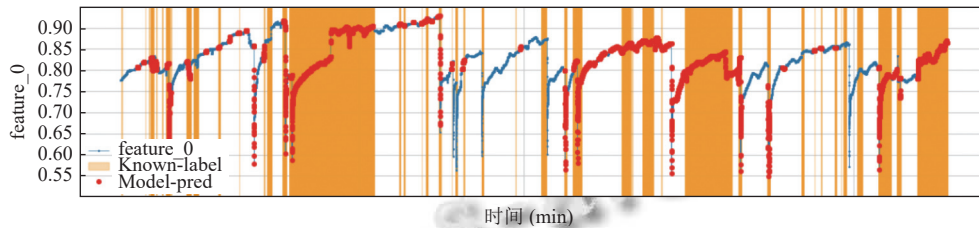


图6 PSM数据集已知异常与模型给定异常对比图

5 结语

本文通过研究无监督多变量时间序列异常检测问题,提出了一种更适合时间序列领域的模型范式,即基于MLP的异常检测模型UMTS-Mixer,由于线性结构对顺序敏感的特性,模型使用MLP学习数据中的时间依赖关系和各变量之间的关联关系获得更好的数据表征.UMTS-Mixer在一组详尽的实证研究上,与其他模型相比,取得更好表现,但依旧存在改进空间,即如何在提高准确率或召回率的同时缩小准确率和召回率的差值.未来工作需要在模型的学习策略上展开大量的研究,以使得模型不仅有好的重建能力,也具备强的正异常界定的能力.

参考文献

- 1 Vaswani A, Shazeer N, Parmar N, *et al.* Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc., 2017. 6000–6010.
- 2 Zhang SM, Liu YC, Zhang XC, *et al.* CAT: Beyond efficient Transformer for content-aware anomaly detection in event sequences. Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Washington: ACM, 2022. 4541–4550.
- 3 Li YF, Peng XY, Zhang J, *et al.* DCT-GAN: Dilated convolutional Transformer-based GAN for time series anomaly detection. IEEE Transactions on Knowledge and

- Data Engineering, 2023, 35(4): 3632–3644. [doi: [10.1109/TKDE.2021.3130234](https://doi.org/10.1109/TKDE.2021.3130234)]
- 4 Xu JH, Wu HX, Wang JM, *et al.* Anomaly Transformer: Time series anomaly detection with association discrepancy. Proceedings of the 10th International Conference on Learning Representations. ICLR, 2022.
 - 5 Tuli S, Casale G, Jennings N R. TranAD: Deep Transformer networks for anomaly detection in multivariate time series data. Proceedings of the VLDB Endowment, 2022, 15(6): 1201–1214. [doi: [10.14778/3514061.3514067](https://doi.org/10.14778/3514061.3514067)]
 - 6 Zeng AL, Chen MX, Zhang L, *et al.* Are Transformers effective for time series forecasting? Proceedings of the 37th AAAI Conference on Artificial Intelligence. Washington: AAAI Press, 2023.
 - 7 Li Z, Rao ZW, Pan LJ, *et al.* MTS-Mixers: Multivariate time series forecasting via factorized temporal and channel mixing. arXiv:2302.04501, 2023.
 - 8 Chen SA, Li CL, Yoder N, *et al.* TSMixer: An all-MLP architecture for time series forecasting. arXiv:2303.06053v5, 2023.
 - 9 Moayedi HZ, Masnadi-Shirazi MA. ARIMA model for network traffic prediction and anomaly detection. Proceedings of the 2008 International Symposium on Information Technology. Kuala Lumpur: IEEE, 2008. 1–6.
 - 10 Anderson O, Kendall M. Time-series. Journal of the Royal Statistical Society (Series D), 1976, 25(4): 308–310.
 - 11 Hundman K, Constantinou V, Laporte C, *et al.* Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. London: ACM, 2018. 387–395.
 - 12 Ng A. Sparse autoencoder. CS294A Lecture Notes, 2011, 72(2011): 1–19.
 - 13 Vincent P, Larochelle H, Bengio Y, *et al.* Extracting and composing robust features with denoising autoencoders. Proceedings of the 25th International Conference on Machine Learning. Helsinki: ACM, 2008. 1096–1103.
 - 14 Noh H, Hong S, Han B. Learning deconvolution network for semantic segmentation. Proceedings of the 2015 IEEE International Conference on Computer Vision. Santiago: IEEE, 2015. 1520–1528.
 - 15 An J, Cho S. Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE, 2015, 2(1): 1–18.
 - 16 Park D, Hoshi Y, Kemp CC. A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. IEEE Robotics and Automation Letters, 2018, 3(3): 1544–1551. [doi: [10.1109/LRA.2018.2801475](https://doi.org/10.1109/LRA.2018.2801475)]
 - 17 Su Y, Zhao YJ, Niu CH, *et al.* Robust anomaly detection for multivariate time series through stochastic recurrent neural network. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Anchorage: ACM, 2019. 2828–2837.
 - 18 Abdulaal A, Liu ZH, Lancewicki T. Practical approach to asynchronous multivariate time series anomaly detection and localization. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. Singapore: ACM, 2021. 2485–2494.
 - 19 Xu HW, Chen WX, Zhao NW, *et al.* Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in Web applications. Proceedings of the 2018 World Wide Web Conference. Lyon: International World Wide Web Conferences Steering Committee, 2018. 187–196.
 - 20 Shen LF, Li ZC, Kwok JT. Timeseries anomaly detection using temporal hierarchical one-class network. Proceedings of the 34th International Conference on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 13016–13026.
 - 21 Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014.

(校对责编: 孙君艳)