基于关系挖掘和注意力的多维时序异常检测

胡智超, 余翔湛, 刘立坤, 张　宇, 于海宁

网络空间安全学院 哈尔滨工业大学 哈尔滨 中国 150001

摘要　在当前信息化的时代, 多维时序数据的异常检测应用广泛, 常用于云服务器、在线服务、系统日志、工业物联网以及智能交通等场景下的状态监控和数据分析中。相比于单一维度的时间序列, 多维时序更加符合实际的场景需求。比如云服务器的关键性能指标, 包括主机 CPU、内存、磁盘 IO 以及网络流量等, 均从不同角度反应了系统状态, 同时彼此之间又存在着关系。传统的时序异常检测方法对这种影响关系考虑不足或者难以高效挖掘这种序列间的隐式关系, 给传统方法在多维时序数据中的应用带来了挑战。本文针对现有方法存在的不足, 提出了基于关系挖掘和注意力机制的异常检测算法 TSAN。该方法首先提出了端到端的序列关系挖掘方法, 通过节点嵌入表示的相似性和图结构来挖掘序列之间关系, 并结合 top-k 和阈值机制来修剪关系图确保其简洁性, 接着利用因果推断生成序列间的因果关系图作为遮罩层, 提高关系图的可解释性和有效性。然后, TSAN 设计了时空注意力网络, 使用时间和空间维度的联合注意力机制来处理混合时空上下文, 用于关系挖掘后的多维时序预测。最后, 提出了异常阈值自动计算方法, 减少了多维时间序列场景下的超参设置, 并且引入最大异常容忍率来排除异常数据的影响, 提高了算法的鲁棒性。从实验结果可以看出, TSAN 在数据集 MSL 和 SMD 上取得了最优的 F1 值, 相比次优方法分别提升了 0.9%(MSL)和 2.3%(SMD), 并且在所有对比方法中具有最小的跨数据集性能波动, 说明了 TSAN 对多维时序数据的异常检测是有效且稳定的。

关键词　异常检测; 多维时间序列; 关系挖掘; 注意力; 机器学习

中图法分类号　TP309.2　DOI 号 10.19363/J.cnki.cn10-1380/tn.2024.11.07

Relation Mining and Attention Based Anomaly Detection for Multivariate Time Series

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

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

Abstract　In the current era of information technology, anomaly detection of multivariate time series is widely used for status monitoring and data analysis in scenarios such as cloud servers, online services, system logs, industrial IoT and intelligent transportation. Compared to single dimensional time series, multivariate time series are more in line with the actual scenario requirements. For example, the key performance indicators of cloud servers, including CPU, memory, disk IO and network traffic information, all reflect the system status from different perspectives and have a relationship with each other. However, traditional anomaly detection methods do not sufficiently consider such influential relationships or are difficult to efficiently mine the implicit relationships between sequences, posing a challenge for the application of traditional methods in multivariate time series. To addresses these limitations, this paper proposes TSAN, an anomaly detection method based on relation mining and attention mechanism. The method first introduces an end-to-end sequence relation mining method. It mines the relation between sequences through the similarity of node embedding and graph structure, and combines top-k and threshold mechanisms to prune the relationship graph to ensure its simplicity, followed by using causal inference to generate the causal graph of sequences as mask layer to improve the interpretability and effectiveness. TSAN then designs a temporal-spatial attention network using a joint attention mechanism to handle the mixed contexts for multivariate timeseries prediction after relation mining. Finally, an automatic calculation method for anomaly threshold is designed to reduce the hyper-parameter settings in multivariate time series scenarios. Besides, TSAN introduces the maximum anomaly tolerance rate to reduce the influence of anomalous data and improves the robustness. From the experimental results, it can be seen that TSAN achieves the best F1 score on the MSL and SMD datasets, with an improvement of 0.9% (MSL) and 2.3% (SMD) respectively compared to the sub-optimal methods and has the smallest cross-dataset performance fluctuations among all the compared methods, indicating that TSAN is effective and stable for anomaly detection of multivariate time series.

Key words　anomaly detection; multivariate time series; relation mining; attention; machine learning

通讯作者: 余翔湛, 研究员, Email: yxz@hit.edu.cn。

本课题得到国家重点研发计划项目(No. 2018YFB1800702)资助。

收稿日期: 2022-10-28; 修改日期: 2022-12-13; 定稿日期: 2024-09-24

1 引言

在当前信息化的时代, 随着各行各业中智能化技术的应用以及大数据的不断发展, 与数据和网络相关的安全问题不断增多。在众多的数据分析与安全防御场景中, 异常检测是一项重要的研究热点[1-2], 常见的应用包括: 入侵检测[3]、故障检测[4]、恶意软件检测以及日志异常检测[5]等。

不同于常规模式下的问题和任务, 异常检测针对的是少数罕见的事件, 具有独特的复杂性, 使得一般的机器学习技术难以取得较好的效果[6]。首先, 异常是未知的, 与许多未知因素有关, 直到发生时才为人所知, 比如网络攻击、信用卡欺诈和网络入侵等[7]; 其次, 异常是不规则的, 一类异常可能表现出与另一类异常完全不同的异常特征; 最后, 异常数据中存在着严重的类别不均衡的情况, 异常通常是罕见的数据实例, 正常实例占数据的绝大部分[8]。

在各类场景的大数据中, 多维时间序列是广泛存在的一种, 对其进行数据分析和异常检测变得越来越重要, 近年来引起大量学者关注, 常用于网络安全、社交网络和智能交通、生态环境等应用中。

(1) 网络安全: 主机状态监控和服务器防护等应用非常依赖日志数据和系统监控数据, 其中有大量多维时间序列。因此有必要在多维时序数据中检测异常并挖掘攻击模式, 以部署实时保护模型[9]。

(2) 社交网络: 社交网络表现出高度的动态性和随时间演变的特性, 其图结构上的多维时间序列分析常用于行为预测、社区检测等任务中[10]。

(3) 智能交通: 道路交通网络是典型的图结构, 有许多人工智能的应用都是基于其产生的时空数据。例如: 交通流量预测、拥堵检测、交通事故检测、自动驾驶等[11]。

(4) 生态环境: 环境监控与治理具有典型的地理空间特征, 很多应用都依赖于地理空间数据和环境指标的监控或检测数据。其中, 水质预测、污染检测和污染溯源等应用正在快速发展[12]。

相比于单一维度的时间序列, 多维时间序列更加符合实际的应用场景, 而且序列之间往往存在着影响关系。根据来源场景的不同, 影响关系分为显式关系和隐式关系。显示关系一般是由先验知识引入的已知信息, 而隐式关系则需要通过数据挖掘的方法从数据中推断。因此, 对于隐式的关系, 如何有效的挖掘和合理的建模成为一个难点。此外, 在多维时序的分析中, 不仅需要考虑时间上下文, 也要考虑由序列间影响带来的空间上下文。

在多维时序数据的异常检测中, 通常使用RNN、LSTM 等方法处理时间上下文, 但大多数都将时间序列视为独立的, 缺乏对序列间关系的分析。也有研究使用 GNN 或者 GNN 与 RNN 的组合处理由关系引入的空间上下文, 但是并没有充分的利用序列之间和时间维度的混合上下文, 序列间隐式的关系结构也不够清晰, 可解释性也较弱。针对现有方法存在的不足, 本文提出了基于关系挖掘和注意力的异常检测算法 TSAN(Temporal-Spatial Attention Network), 并做了如下几个方面的工作:

(1) 提出了端到端的序列关系挖掘方法。该方法通过序列节点嵌入表示的相似性定义序列间的关系, 结合 top-k 和阈值机制修剪关系图确保其简洁性, 并通过因果推断生成序列间的因果关系图作为遮罩层, 提高了关系图的可解释性和有效性。

(2) 构建了时空注意力网络。随着关系图的生成, 在原有的时间上下文之外引入了隐含的空间上下文。TSAN 使用时间和空间维度的联合注意力机制, 通过消息传递的信息融合方法来处理混合时空上下文, 用于关系挖掘后的多维时序预测。

(3) 设计了异常阈值自动计算方法。基于验证集数据为时间序列自动计算异常阈值, 减少了多维时间序列场景下的超参设置。引入最大异常容忍率来排除异常数据的影响, 提高了算法的鲁棒性。最后, 将本文算法与同类算法进行对比, 通过实验证明了本文所提算法的有效性。

本文后续部分安排如下: 在第 2 章梳理研究现状; 在第 3 章首先给出问题具体的数学描述, 然后介绍实现框架; 第 4 章首先介绍序列关系挖掘方法和混合上下文的时空展开, 然后描述基于时空注意力的时序预测与异常检测方法; 第 5 章通过真实数据集上的实验验证所提方法的有效性; 第 6 章对全文进行总结, 并提出未来的研究方向。

2 研究现状

近年来, 异常检测被广泛地应用于包括图数据、日志数据、时间序列数据等多样化数据的分析任务中[13]。本文专注于多维时序数据的异常检测, 由于传统时序异常检测方法对多维时序的序列间关系考虑的不足, 本文通过挖掘多维序列关系来提升时序异常检测效果, 核心是时序异常检测、图神经网络以及注意力机制。

2.1 时序异常检测

时序数据中的异常主要指一个时刻的观测值与发展预期不符, 产生的差异过大, 而导致异常。现有

的检测方法主要通过误差来量化这种差异。误差表示数据点的正常值和观测值之间的差异大小, 常用的为 L-P 范数。一般通过重构和预测来完成误差计算, 其中重构一般用于离线数据或难以预测的场景, 使用重构后的数据作为正常值。预测一般用于可预测的场景, 要求可以获得数据的生成模型, 使用预测值作为正常值。

基于重构的方法假设异常点是不可被压缩的或不能从低维映射空间有效被重构的。机器学习中常用的方法的有 PCA、Robust PCA、Random Projection 等降维方法[14-15]。深度学习中, 基于重构的异常检测方法通常包含一个自动编码器, 由编码器和解码器组成, 编码器将原始数据映射到低维特征空间, 而解码器试图从投影的低维空间恢复数据。这两种网络的参数通过重构损失函数来学习。为了使整体重构误差最小化, 保留的信息必须尽可能与输入实例相关。因此, 可以基于自编码器来重构数据, 重构误差大的代表异常程度大。典型案例有全连接自动编码器(Dense AE)、稀疏自动编码器(Sparse AE)、去噪自动编码器(Denoising AE)、收缩自动编码器(Contractive AE)、鲁棒自动编码器(Robust Deep AE)等[16-17]。该类方法的优点是能够通过非线性方法捕捉复杂特征, 试图找到正常实例的一种通用模式, 缺点是如何选择正确的压缩程度, 以及如何解决过拟合的问题。

近年来, 由于生成对抗网络(Generative Adversarial Network, GAN)在生成任务上的优秀表现, 不少研究者将其用于异常检测, 如 TadGAN[18]、MAD-GAN[19]和 TAnoGAN[20]。以 TAnoGAN 为例, 它是一个基于 GAN 的无监督的异常检测方法, 首先将原始时间序列划分为较小的序列, 然后使用 GAN 来学习较小序列的分布。为了处理序列数据, TAnoGAN 在生成器和判别器中都使用了 LSTM 作为基础网络单元。真实数据和生成的数据之间的重构损失包括残差损失和判别损失。一个较小序列的异常得分是这两种损失的加权和。

对于在线的时序数据, 有部分研究者通过统计特征或者数据结构来描述数据集的特性, 用数据对这个统计特性引起的变化来衡量影响大小。异常数据造成的影响要远高于正常数据, 通过影响阈值可筛选出数据集中的异常点。由亚马逊推出的鲁棒随机分割森林算法(Robust Random Cut Forest, RRCF[21])是这类算法的代表。RRCF 将数据点的异常得分视为包含或不包含该点而导致整体树结构发生改变的程度, 是一种针对动态流数据的有效的异常检测方法。它主要是对 IF 进行了优化改进。RRCF 设计了一个稳健的随机切割树数据结构, 并将其作为输入流的草图或概要。然后, 对于任何给定的样本, 当我们将样本添加到树上或从树上移走样本时, 其异常情况可以通过树的变化来测量。

基于预测的异常检测方法使用时间窗内的历史实例预测当前实例来学习特征表示[22-23], 这些特征表示能够捕捉时间或序列的依赖关系, 正常实例通常能够保持良好的依赖关系, 可以很好地被预测, 而异常实例通常会违反这些依赖关系, 使得不可预测[24]。

时序数据上的预测模型通常以多层感知器(Multi-Layer Perceptron, MLP[25])和长短期记忆网络(Long Short-Term Memory, LSTM[26])为基础网络单元。MLP 是一类前馈的人工神经网络, 利用一种反向传播的监督学习技术进行训练, 但是忽略了序列中的时间上下文。LSTM 是 RNN 的最佳变体之一, 利用记忆和门控机制来处理时间上下文, 在针对序列数据的相关任务中表现良好。由于其良好的预测性能, 将 LSTM 用于序列上的异常检测[27-28], 取得了不错的效果。

由于多维时间序列之间关系的存在, 研究者将显式或者隐式的空间上下文引入到时序数据的异常检测中。Geng 等[29]基于显式图结构进行多维时间序列预测, Lim 等[30]利用注意力机制完成多维时间序列预测。基于预测和观测的偏差, 对异常进行判定是这类方法的通用模式。Kieu 等[31]利用原始时间序列的统计特征增强其特征空间, 并使用自编码器和 LSTM 结合完成异常检测。

对于隐式的序列关系, 需要从数据角度进行关系挖掘。为了挖掘数据的空间上下文, Zhao 等[32]使用双层图注意力网络捕获不同序列间的关系, 从而明显地减少了误报。除了在 GNN 的帮助下进行空间上下文的处理外, RNN 在多维时间序列中也被广泛使用。Su 等[33]提出了 OmniAnomaly, 通过随机变量连接和平面归一化流等核心技术, 学习了它们的稳健表征来捕捉多维时间序列的正常模式。然后, OmniAnomaly 通过这些表征重建输入数据, 并通过重建概率确定异常情况。随着图注意力的广泛应用, Deng 等[34]提出了 GDN, 通过引入图来建模序列之间的关系, 同时使用过去时刻的数据作为图节点属性, 并使用 GAT 完成预测和异常检测。

此外, 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[35-36]使用数据迁移和特征迁移来增强异常检测效果, 其中数据

迁移通过数据扩充生成合成数据来扩大训练集数据量, 从而更好地进行正常实例的表征学习, 特征迁移可以从相关问题中提取一些表征层来提高异常检测模型的精度。

目前时序异常检测算法在多维时间序列场景下的应用仍存在一些不足之处:

(1) 缺乏有效的序列关系挖掘方法。异常检测方法在序列间存在影响关系的场景下, 忽略序列关系会导致异常检测效果的下降;

(2) 序列关系挖掘的结果在简洁性和可解释性上还存在提升空间;

(3) 在多维时序预测时, 对时空混合上下文的使用不够充分。

本文提出的 TSAN 结合了因果推断的可解释性以及神经网络的可学习性, 实现了端到端的序列关系挖掘, 旨在通过关系挖掘增强多维时序的异常检测效果, 对现有异常检测方法存在的不足进行改进。

2.2 图神经网络与注意力

图神经网络和注意力机制是本文所提方法 TSAN 进行多维时间序列异常检测的基础。

(1)图神经网络: 卷积神经网络(Convolutional Neural Network, CNN)已被成功用于许多任务, 包括图像分类、语义分割或机器翻译。对于这些具有网格状结构的数据, CNN 有能力提取多尺度的局部空间特征, 并将其结合起来建立高表现力的抽象特征, 通过考虑空间信息为数据产生更好的表示。然而, 许多任务涉及的数据不能用规则网格结构来表示, 例如社交网络和通信网络。Gori 等[37]和 Scarselli 等[38]提出了图神经网络(Graph Neural Network, GNN), 它使用节点之间的消息传递来捕捉图的依赖关系, 因此它可以直接处理更多的图类。相比 CNN, GNN 在不规则领域泛化并应用卷积神经网络。近年来, 图卷积网络(GCN)、图递归网络(GRN)和图注意力网络(GAT)等 GNN 的演化模型在许多深度学习任务上表现出了突破性的性能[39]。

(2)注意力机制: 注意力机制已被广泛用于各种面向序列的任务中。其优点是可以处理各种大小的输入, 关注与输入信息中最相关的部分, 可以更好的提取特征。对于单一序列, 研究者提出了自注意力机制来计算序列的表征。注意力的应用在多个任务中都取得了很好的效果: 如机器阅读、序列表征学习、以及机器翻译等[40]。作为 GNN 和注意力的结合, Velickovic 等[41]提出了 GAT, 一个在图结构化数据上运行的图神经网络架构。它采用了自注意力遮罩层, 用于解决 GCN 对全图节点依赖的缺点。

3 异常检测框架

针对无监督场景下多维时序数据的异常检测, 本节首先对具体问题进行描述, 然后对本文提出方法的框架进行整体的阐述。

3.1 问题描述

多维时间序列具有一致的采样频率和相同的记录条目数, 这意味着对于任意一个采样时刻, 每个序列都有唯一一个观测项。定义 S 为所有时间序列的集合, 一共有 N 个时间序列, $|S|=N$, 则有 $S=\{s_1,s_2,s_3,\ldots,s_N\}$。每个时间序列的长度为 T, 则单个时间序列可记为:

$$s_k=\{s_k^1,s_k^2,s_k^3,\ldots,s_k^T\},\forall 1\leqslant k\leqslant N. \tag{1}$$

对于任意一个时刻, 都有 N 个观测项。每个序列的观测项均是 F 维的向量, 表示序列在某个时刻观测到的 F 个指标值, 记为:

$$s_k^t=\{m_1,m_2,\ldots,m_F\},\forall 1\leqslant k\leqslant N,1\leqslant t\leqslant T. \tag{2}$$

给定一个数据集 $D_{train}=\{s_1,s_2,s_3,\ldots\}$ 作为训练数据。异常检测的任务是学习一个映射 $f:\mathcal{X}\to\mathcal{Y}$, 其中输入 $\mathcal{X}=D_{train}$ 是在时刻集合 T_{train} 上观测到的时间序列, $\mathcal{Y}$ 是与每个时间刻度相关的标签集。在无监督的异常检测方法中, 一般假设训练数据仅由正常的数据实例组成。

在测试数据集 $D_{test}=\{(s_1)',(s_2)',(s_3)',\ldots\}$ 上进行检测, 其中 D_{test} 来自相同的 N 个时间序列, 但在不同的时间刻度集 T_{test} 上采样。检测算法产生的标签长度为 T_{test}, 标签表示每个时间点是否为异常点, 即 $f(t)\in\{0,1\}$, 如果 $f(t)=1$ 那么时间 t 是异常点。

基于预测的异常检测方法, 推理结果的输出经过两个阶段, 预测和异常判定。如公式(3)所示:

$$\begin{cases}\widehat{\mathcal{X}} &= p(\mathcal{X}),\\ \text{score} &= \delta(\mathcal{X},\widehat{\mathcal{X}}),\end{cases} \tag{3}$$

其中 p 为预测函数, 预测多维序列在每个时刻的观测值。δ 为异常评分函数, 根据每个时刻的观测值和预测值计算该时刻的异常得分。如果时刻 t 的异常得分大于异常阈值 τ, 即 $\text{score}(t)>\tau$, 则 t 时刻为异常点。所以, 函数 f 可以有 δ 和 p 复合而成, 即:

$$f=\delta(\mathcal{X},p(\mathcal{X})). \tag{4}$$

3.2 实现框架

本文提出的异常检测方法 TSAN 在序列关系挖掘的基础上, 通过对多维时序数据的预测, 计算序列的预测值与其实际观测值的偏差, 从而判断时刻是否异常。

TSAN 多维时序异常检测方法的框架如图 1 所示, 主要包括关系挖掘、时空预测以及异常检测三个部分。

(1) **关系挖掘**: 首先, 为每个单独的序列建立可学习的嵌入表示。然后, 以序列为顶点, 序列关系为边, 通过节点相似性、修剪策略和因果推断等方法构建序列间的关系图。

(2) **时空预测**: 基于不同的注意力机制, 分别处理序列对相邻节点和历史数据的依赖。通过两种注意力机制的融合, 形成时空注意力, 并使用它对具有混合时空上下文的图序列进行预测。

(3) **异常检测**: 首先, 根据每个时刻的观测值和预测值, 为数据集中的每个时刻计算异常得分。然后, 根据自动异常阈值 τ 计算得到每个时刻的异常标签。

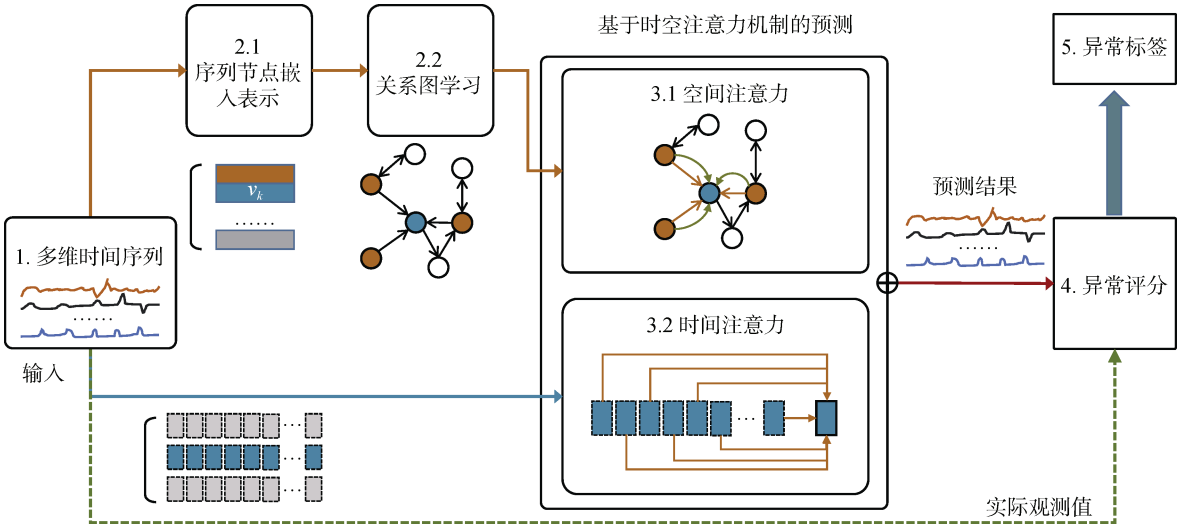

图 1 多维时序数据异常检测框架

Figure 1 Framework of anomaly detection for multivariate time series

4 异常检测方法实现

本文针对多维时间序列的异常检测, 首先需要挖掘序列间的关系。然后, 利用时空注意力机制处理由序列关系引入的时空上下文, 并进行多维时序数据的预测。最后, 使用自动阈值的方法计算每个时刻的异常标签, 完成异常检测。

4.1 序列关系挖掘

存在关联影响关系的序列之间具有特征上的相似性或者行为上的相似性, 如观测数值大小、变化趋势或其它统计特征等。每个序列有自己的属性特征和观测序列值, 同时序列间存在关联影响关系, 由于图适用于对象及其关系的建模。因此使用图来描述这样的关系模型。

由于不同节点的特征及行为并不一样, 因此关系图是有向图。记序列关系图为 $G=(V,E)$, 其中 V 是所有顶点的集合, E 是所有边的集合, 邻接矩阵为 $\boldsymbol{A}$。$A_{ij}=1$ 表示节点 i 与 j 存在一条有向边, 即序列 i 对序列 j 存在影响。为每个节点建立可学习的嵌入表示, 记为: $v_i=R^d$, 是 d 维的实数向量。

序列之间的相似度由节点向量的余弦相似度定义。对于任意两个节点 v_i 和 v_j, 它们之间的夹角记为 θ, 则节点间的相似度计算公式如下:

$$s(i,j)=\cos\theta=\frac{v_i^{\top}v_j}{\|v_i\|\ \|v_j\|},\forall 1\leqslant j\leqslant N. \tag{5}$$

节点间相似度的大小决定了节点间关联关系的强弱。以关联关系的强弱为优先级, 每个节点只选择有限个节点作为相邻节点(有向边起点)。为了去除无关的以及作用较小的边, 本文使用 top-k 和阈值混合机制获取修剪后的邻接矩阵。记 filter 为筛选函数, 它从集合中选取最多 K 个元素, 并且值不小于阈值。那么, 可以得到邻接矩阵 $\boldsymbol{A}$ 的计算公式:

$$A[ji]=1\{j\in \text{filter}(\{s(i,j):j\in\mathcal{N}_f(i)\},\tau_g)\}, \tag{6}$$

其中 $\mathcal{N}_f(i)$ 表示筛选之前节点 i 的相邻节点集合。τ_g 是筛选阈值。

基于 top-k 及阈值的修剪机制示意图如图 2 所示。在样例中, 设置 $K=3$, 关系阈值 $\tau_g=0.05$。对于图中的每个节点, 最多有 K 个边指向该节点, 表示空间关系信息的输入。图中虚线连接的边表示相似度值低于阈值 τ_g, 需要被裁剪。以节点 1 为例, 节点 2,3,6 均有 1 条边指向节点 1, 表示序列 2,3,6 对序

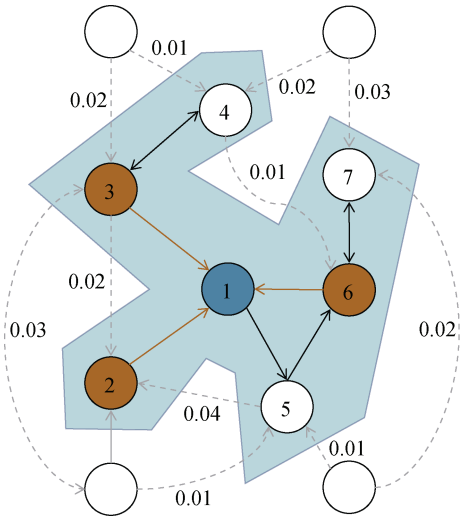

图 2　关系图修剪机制示意图

Figure 2　An example of prune mechanism

列 1 有影响, 同时由于有向边 (1,5) 的存在, 序列 1 对序列 5 有影响。通过 top-k 和阈值的修剪, 最终生成的关系图包括节点: $\{1,2,3,4,5,6,7\}$ 以及有向边: $\{(2,1),(3,1),(6,1),(4,3),(3,4),(1,5),(5,6),(7,6),(6,7)\}$。

为了保证关系图的有效性, 使用因果关系图作为遮罩层对关系图进行限制。本文使用 PC 算法[42]来学习用于遮罩限制的因果关系图。PC 算法首先确定节点间的依赖关系, 但不确定方向, 即先生成一个无向图, 然后再确定依赖方向, 从而把无向图扩展为完全部分有向无环图(Completed Partially Directed Acyclic Graph, CPDAG), 具体步骤如下:

(1) 生成关系骨架图: 对于 G 中的两个相邻点 i 和 j, 如果 i 和 j 如果能在给定节点 k 时条件独立, 则删除 i 和 j 的边。我们需要对任意两个节点进行条件独立性检验, PC 算法采用了 Fisher Z Test 作为条件独立性检验方法。这样会得到一个无向图, 图中的无向边表示它连接的两个节点之间有因果关系。

(2) 从骨架生成完全部分有向图: 对于所有具有相邻节点 k 的不相邻节点对 i 和 j, 如果 k 不属于 i 和 j 的分割集, 将 $i-k-j$ 方向设定为 $i\to k\leftarrow j$。为了得到更多的有向边, 重复以下几条规则对无向边进行方向设置:

规则 1: 对于 $j-k$, 如果存在 $i\to j$, 且 i 与 k 不相邻, 则方向设定为 $j\to k$;

规则 2: 对于 $i-j$, 如果存在 $i\to k\to j$, 则方向设定为 $i\to j$;

规则 3: 对于 $i-j$, 如果 $i-k\to j$, $i-l\to j$, 且 k 与 l 不相邻, 则方向设定为 $i\to j$;

规则 4: 对于 $i-j$, 如果 $i-k\to l$ 和 $k\to l-j$, 且 k 与 l 不相邻, 那么方向设定为 $i\to j$。

最终, 得到序列关系的因果图。对于由公式(6)得到的邻接矩阵, 其中的任意一条有向边, 如果在因果图中依然保持有向边, 那么序列关系图中两个序列存在有向边, 否则删除。

4.2　图序列时空展开

在多维时间序列中, 对序列进行预测时, 因为其所处上下文同时包括时间和空间, 所以输入包括: (1)该序列中相对于当前时刻而言若干个过去时刻的观测数据, 即历史影响数据; (2)同一时刻其它序列观测的数据, 即关系影响数据。因此预测模型的核心之一是要同时对数据的时间和空间两种不同上下文进行描述。

从混合上下文的角度, 通过时空依赖的展开对时序数据的预测进行解释。图 3 描述了多维时间序列的时空展开, 每个时刻的多维时间序列数据都被转换为关系图的形式, 每个序列具有不同的节点属性。以时间序列 1 在时刻 $t-k$ 为例子进行展开说明, 关系图中的时间序列在受到其他时间序列的影响时, 这些时间序列的边都指向它, 并且以绿色线条表示额外的上下文信息输入。

在图 3 中, 左右两侧为时空角度, 中间部分是时空展开角度。其中, 中间部分的左侧为时间维度展开, 中间部分的右侧为空间维度展开。说明如下:

(1) 时空视图: 每个时刻都是一个图, 图的结构在不同的时刻保持不变, 但每个节点的观测值会发生变化。因此, 多维时间序列的转换输入是一个时空视图中的图序列。

(2) 时间维度展开: 在时间维度和空间维度上展开时刻 $t-k$ 后, 仅在时间视图中描述序列。对于节点 1, 它是一个单一的时间序列, 与其他时间序列没有关系。因此可以只关注时间维度的影响。

(3) 空间维度展开: 在时间维度和空间维度上展开时刻 $t-k$ 后, 仅在空间视图中描述序列。在空间视图中, 对于节点 1 来说, 它只通过关系边与节点 $\{2,3,6\}$ 有结构性的连接关系, 与过去的时间点没有关系, 和非连接节点 $\{4,5,7\}$ 之间也没有直接的关联关系。

根据上述展开视角的分析, 本文采用基于消息传递的方法将一个节点的时空上下文融入到时间序列的预测中。在关系图中, 聚合来自其邻居的“消息”, 在时间序列中, 聚合来自过去依赖时刻的“消息”。

上下文信息融合包括两个阶段: “消息”和“聚合”。每个节点或者时刻将创建一个消息, 随后发送

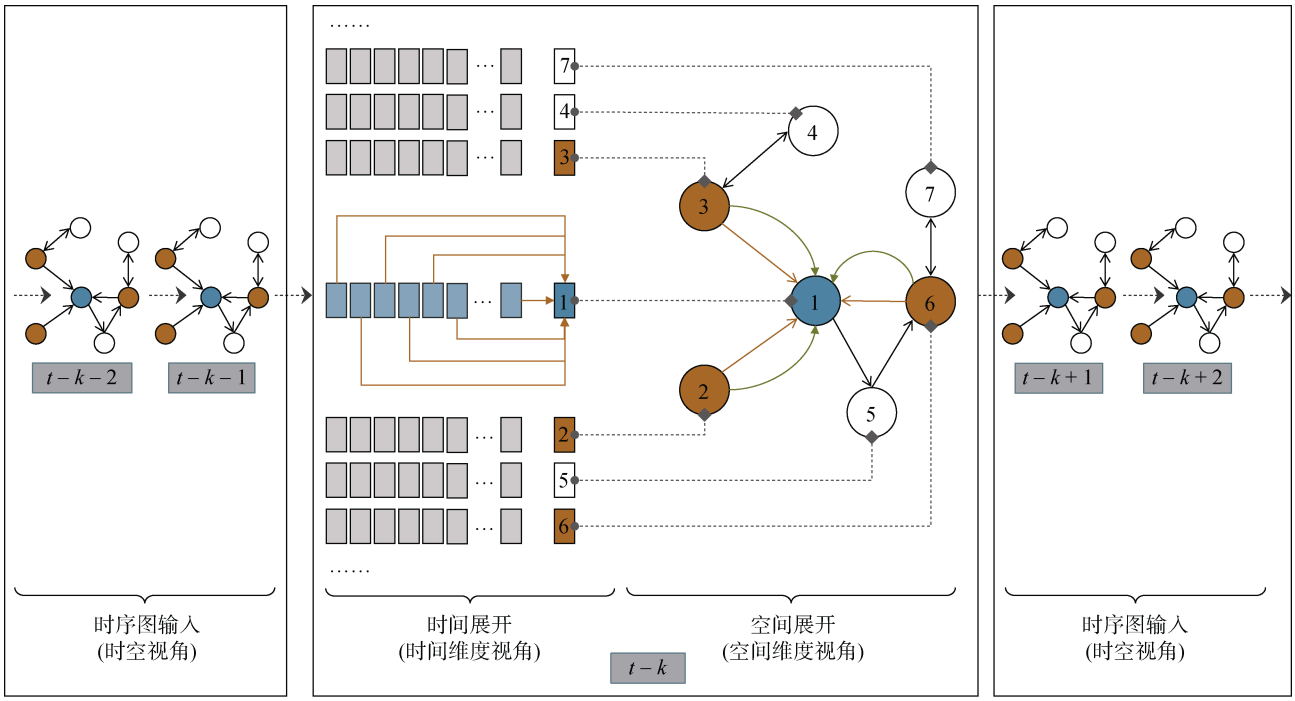

图 3　多维时间序列的时空展开图

Figure 3　Expansion view of multivariate time series

给其他节点或时刻。通常使用一个线性层将节点特征转化为消息。记 v 为消息接收目标，$\boldsymbol{\eta}$ 为属性特征，$\mathcal{A}$ 返回消息来源集合，则消息传递神经网络的第 l 层产生的消息 $\mathbf{m}$ 为:

$$\mathbf{m}_u^{(l)} = \mathrm{MSG}^{(l)}(\boldsymbol{\eta}_u^{(l-1)}), u \in \{\mathcal{A}(v) \cup v\}. \tag{7}$$

然后，v 将汇总所有接收到的消息以获得新的特征表示:

$$\boldsymbol{\eta}_v^{(l)} = \mathrm{AGG}^{(l)}(\mathbf{m}_u^{(l)}, u \in \{\mathcal{A}(v) \cup v\}, \mathbf{m}_v^{(l)}). \tag{8}$$

将这一机制应用于时空场景，用 $s_i^{k(l)} \in \mathcal{R}^F$ 表示系列 i 时间刻度 k 在层 l 中的特征，消息传递神经网络的常见形式是:

$$s_i^{k(l)} = \gamma^{(l)}(s_i^{k(l-1)}, \square_{(j,t)\in\mathcal{A}(i,k)} \phi^{(l)}(s_i^{k(l-1)}, s_j^{t(l-1)})), \tag{9}$$

其中 $\square$ 表示一个可微的、置换不变的函数，如求和、平均值或最大值，γ 和 ϕ 表示可微调的函数，如多层感知机。

4.3　基于时空注意力的预测

通过时空展开的分析，对时空混合上下文中的数据进行了建模。异常检测将基于“预测”和“异常评分”完成。TSAN 设计了时空注意力用于时空混合上下文场景下的预测和异常检测。注意力机制使得模型可以关注到输入数据中的有效部分，这有助于解释上下文扩展中的依赖关系，还有利于完成异常的可追溯性，并提供一些可解释性。

对于任何两个观察值 s_i^t 和 s_j^k，一个共享的注意力机制 $\mathbf{a}$ 计算注意力系数并应用 LeakyReLU 非线性变换。

$$e_{\mathbf{a}}(s_i^t, s_j^k) = \mathrm{LeakyReLU}(\mathbf{a}^\top [\Theta s_i^t \| \Theta s_j^k]), \tag{10}$$

其中 Θ 是可学习的线性变换，将输入特征转换成更高层次的特征: $\mathcal{R}^F \to \mathcal{R}^{F'}$，$F'$ 表示转换后的特征维度，通常 $F' \neq F$。注意力机制 $\mathbf{a}$ 由一个单层前馈神经网络来实现: $\mathcal{R}^{F'} \times \mathcal{R}^{F'} \to \mathcal{R}$。注意力系数 $e_{\mathbf{a}}(s_i^t, s_j^k)$ 表示 s_j^k 对于 s_i^t 的重要性，等同于 s_i^t 针对 s_j^k 分配的注意力大小。

需要注意的是，$\mathbf{a}$ 只在同一维度上共享，时间和空间维度上应用不同的机制 $\mathbf{a}$。最后，TSAN 将时间和空间的注意力融合以完成预测。

对于第 i 个时间序列的第 t 时刻，即 s_i^t，时空注意力算子在时间和空间两个维度上分配注意力。时间维度上的窗口大小为 w，表示通过过去 w 个时刻的数据预测下一个时刻的数据。$\mathcal{W}(t)$ 表示 t 时刻过去 w 个时刻的集合。$\mathcal{N}(i)$ 是节点集合，这个集合中的所有节点均有一条有向边指向节点 i。在应用时空注意力时，在两个不同的维度上，注意力被分配到的对象集合是不同的:

(1) 在时间维度上，s_i^t 需要分配注意力的对象集合是 $\mathcal{S}_{temporal}(i,t) = \{s_i^k, \forall k \in \mathcal{W}(t)\}$。因此，在时间维度上，基于时间注意力进行预测的公式为:

$$(s_i^t)'_{temporal} = \alpha_i^{t,t}\Theta s_i^t + \sum_{k\in\mathcal{W}(t)} \alpha_i^{t,k}\Theta s_i^k, \tag{11}$$

其中, 归一化的注意力系数 $\alpha_i^{t,k}$ 使用注意力机制 $\mathbf{a}_1$ 计算得到:

$$\alpha_i^{t,k} = \frac{\exp(e_{\mathbf{a}_1}(s_i^t, s_i^k))}{\sum_{k\in\mathcal{W}(t)\cup\{t\}} \exp(e_{\mathbf{a}_1}(s_i^t, s_i^k))}. \tag{12}$$

(2) 在空间维度上, s_i^t 需要分配注意力的对象集合是 $\mathcal{S}_{spatial}(i,t) = \{s_j^t, \forall j \in \mathcal{N}(i)\}$。因此, 在空间维度上, 基于空间注意力的预测公式为: 。

$$(s_i^t)'_{spatial} = \alpha_{i,i}^t\Theta s_i^t + \sum_{j\in\mathcal{N}(i)} \alpha_{i,j}^t\Theta s_j^t, \tag{13}$$

其中, 归一化的注意力系数 $\alpha_{i,j}^t$ 使用注意力机制 $\mathbf{a}_2$ 计算得到:

$$\alpha_{i,j}^t = \frac{\exp(e_{\mathbf{a}_2}(s_i^t, s_j^t))}{\sum_{j\in\mathcal{N}(i)\cup\{i\}} \exp(e_{\mathbf{a}_2}(s_i^t, s_j^t))}. \tag{14}$$

最终, 使用平均值函数 MEAN 融合时间和空间注意力, 应用时空注意力预测得到的结果为:

$$(s_i^t)' = \mathrm{MEAN}((s_i^t)'_{temporal}, (s_i^t)'_{spatial}). \tag{15}$$

预测模型的损失函数使用均方差(Mean Squared Error, MSE), 如下:

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{T_{train} - w}\sum_{t=w}^{T}\sum_{i=1}^{N}\|(s_i^t)' - (s_i^t)\|_2^2. \tag{16}$$

4.4 异常评分

基于预测的异常检测模型, 通常在它们的假设中, 训练集和验证集都是正常数据, 不包含异常数据。因此可以从验证集中利用预测误差自动计算出异常评分的阈值。为了增加模型的鲁棒性, TSAN 增加“最大异常容忍率”, 来处理验证集中可能存在的少量异常数据。

从预测模型的角度, 异常描述了数据的预测偏离观测的程度。在多维序列中, 每个序列的数据分布彼此不同, 所以应具有单独的异常得分阈值。阈值由验证集上该序列的预测误差决定。

$$E_i^t = |(s_i^t)' - (s_i^t)|. \tag{17}$$

由每个时间点的预测误差组成的序列为 E, 最大异常容忍率为 η, $E_{\max}$ 和 $E_{\min}$ 分别代表最大和最小误差值。找到误差阈值的过程描述为算法 1。

算法 1. find_error_threshold (E , η).

输入: 误差序列 E, 最大异常容忍率 η

输出: 误差阈值 E_τ

1 $E \leftarrow$ 将 E 按照降序排列

2 $\gamma_1 = 0$, $\gamma_2 = 0$, L = len(E)

3 记录最大最小值: $E_{\max} = E[0]$, $E_{\min} = E[L-1]$

4 FOR $i = 0$ to $L-1$ DO

5 IF $E[i] \neq E[i+1]$ and $i/L \leq \eta$ THEN

6 $\gamma_1 = i, \gamma_2 = i+1$

7 END

8 END

9 计算误差阈值 $E_\tau = (E[\gamma_1] + E[\gamma_2])/2$

10 RETURN E_τ

为了确定一个时刻是否异常, 首先根据预测结果对每个序列的异常情况进行评分。异常得分阈值是对每个序列单独计算的, 并采用 Min-Max 方法进行归一化。

$$\tau_i = \begin{cases} \dfrac{E_\tau - E_{\min}^i}{E_{\max}^i - E_{\min}^i}, & E_{\max}^i > E_{\min}^i, \\ 0, & E_{\max}^i = E_{\min}^i. \end{cases} \tag{18}$$

因此, 每个时间时刻的异常得分为:

$$\mathrm{score}_i(t) = \frac{E_i^t - E_{\min}^i}{E_{\max}^i - E_{\min}^i}. \tag{19}$$

所以, 可以得到每个时间点的异常标签:

$$y_i(t) = \max(\mathrm{sign}(\mathrm{score}_i(t) - \tau_i), 0). \tag{20}$$

对于时刻 t, 当有任意时间序列发生异常时, 则认为多维时间序列整体是异常的, 因此最终的多维序列异常标签计算方式如下:

$$y(t) = \max(\{y_i(t) : \forall i \in \{1, 2, \ldots, N\}\}). \tag{21}$$

5 实验结果及分析

本节首先介绍实验用到的数据集、基准方法和相关设置, 然后对多维时间序列上的异常检测结果进行对比分析。

5.1 实验数据集

本文使用两个数据集 Server Machine Data(SMD)[33] 和 Mars Science Laboratory Rover(MSL)[13]进行多维时间序列异常检测。SMD 是从服务器收集的, 包括许多主机状态监测指标。MSL 是从航天器上收集的, 包括大量的遥感监控数据。表 1 显示了数据集的实体和监控指标信息[33]。

每个数据集均包含了训练集和测试集, 是从相同来源采样的多指标时序数据。实验中, 我们在训练集中划分一定比例的数据作为验证集。相比于训练集, 测试集包含了每个时刻是否是异常的标签。

表 1　多维时间序列数据集信息(SMD 和 MSL)

Table 1　Multivariate time series datasets

数据集	实体	维度	指标信息
SMD	28	38	CPU 负载, 网络流量, 内存使用率等
MSL	27	55	遥感监控数据: 辐射、温度、功率以及计算活动等。

5.2　基准与实验设置

本文选择 5 种方法: LSTM、GCN_LSTM、GAT_LSTM、GDN[34]和 OmniAnomaly[33]作为基准方法与 TSAN 进行对比, 这些方法的特点总结如下:

(1) LSTM: 本文基于 LSTM 构建了一个预测模型, 将任意时刻的 N 个序列观测项当作是 1 个观测项的 $N*F$ 个属性值进行预测, 忽略序列之间的关联关系。

(2) GCN_LSTM: 本文基于 LSTM 和 GCN[43]构建的预测模型, 使用 GCN 对序列关系图进行空间上下文处理。GCN 网络层为每个时刻的关系图生成嵌入表示, 之后图嵌入表示的序列将作为 LSTM 的输入, 使用最后一个时刻的输出作为图序列的预测, 最后通过一个线性层作为解码器将预测结果转为多维时间序列的预测。

(3)GAT_LSTM: 本文基于 LSTM 和 GAT[41]建立的预测模型, 其设计方式与 GCN_LSTM 相同。但是使用 GAT 代替 GCN 处理空间上下文。

(4) GDN: 将结构学习方法与图神经网络相结合, 并利用注意力权重为异常检测结果提供可解释性。它使用历史时间窗口内的数据作为图节点的属性特征进行预测。通过图注意力机制处理空间上下文, 在多维时间序列异常检测中表现良好。

(5) OmniAnomaly: 设计了一个用于多变量时间序列异常检测的随机递归神经网络。在考虑时间依赖性和随机性的同时, 学习潜在的表示, 以捕获多维时间序列的正常模式。然后使用这些表征来重建输入数据, 并使用重建概率来确定某个时刻是否是异常, 在多种场景下具有很好的鲁棒性。

表 2 展示了 LSTM、GCN_LSTM、GAT_LSTM 和 TSAN 这 4 种算法针对时空上下文的递进式的处理关系, 在是否考虑时空上下文以及采用何种上下文处理策略上是不同的。通过这样的设计, 验证上下文与处理机制是否有效。

本文实验环境基于 Python 3.9.7 搭建, 方法使用了 pytorch 1.8.0 的 CUDA 11.1 版本, 以及 pytorch-geometric 1.5.0。实验的设置说明如下:

(1) 在实验中, 使用 10%的训练集数据作为验证集。对于所有方法, 在将序列转化为模型输入时, 设置滑动窗口大小为 15, 步长为 5。

表 2　上下文与处理机制

Table 2　Context and mechanism

方法	时间上下文		空间上下文	
	是否考虑	处理机制	是否考虑	处理机制
LSTM	是	LSTM	否	无
GCN_LSTM	是	LSTM	是	GCN
GAT_LSTM	是	LSTM	是	Attention
TSAN	是	Attention	是	Attention

(2) 在 GDN 方法中, 每个序列最大的关联序列数量限制设置为 $K=20$。

(3) 对于 GCN_LSTM, GAT_LSTM 和 TSAN, 均采用 top-k 及阈值机制对关系图模型进行修剪。设置 $K=20$, $\tau_g=0.05$ 用于移除影响较小的关系连接边。因果图生成时的显著性水平设置为: $\alpha=0.05$。

(4) 对于 LSTM, GCN_LSTM, GAT_LSTM 和 TSAN, 异常评分的阈值均使用算法 1 和公式(18)进行计算, 并且设置最大异常容忍率 $\eta=0.001$。

由于异常检测中的正例数量远少于反例数量, 并且使用者更关心正例的识别, 所以采用 F1 分数(F1-Score, F1), 召回率(Recall, R), 精确率(Precision, P)作为异常检测结果的评价指标:

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}} \tag{22}$$

$$\text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{23}$$

$$\text{F1}=\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \tag{24}$$

5.3　结果分析

TSAN 进行多维时序异常检测时, 首先会挖掘出序列间的关系, 结果表现为有向的关系图。以 SMD 中的一台主机上的序列关系挖掘结果为例, 一共有 38 个序列, 除去 7 个无变化的序列数据, 生成的关系图中一共有 31 个节点, 部分关系图如图 4 所示。

在 MSL 和 SMD 数据集上, 基准方法与 TSAN 执行异常检测得到的 F1 值如表 3 所示。其中异常检测方法在每个类别取得的 F1 值是在该类别下所有数据序列上异常检测 F1 值的均值。图 5 展示了不同角度的性能对比。

从结果可以看出, 在两个数据集上, 本文提出的 TSAN 方法都取得了最好的 F1 值。

(1) 在数据集 MSL 上, TSAN 的 F1 值为 0.907, 相比于次优的 OmniAnomaly, 提升约 0.9%。TSAN 的精准率为第二优, 高于平均值 0.821, 除了最优的

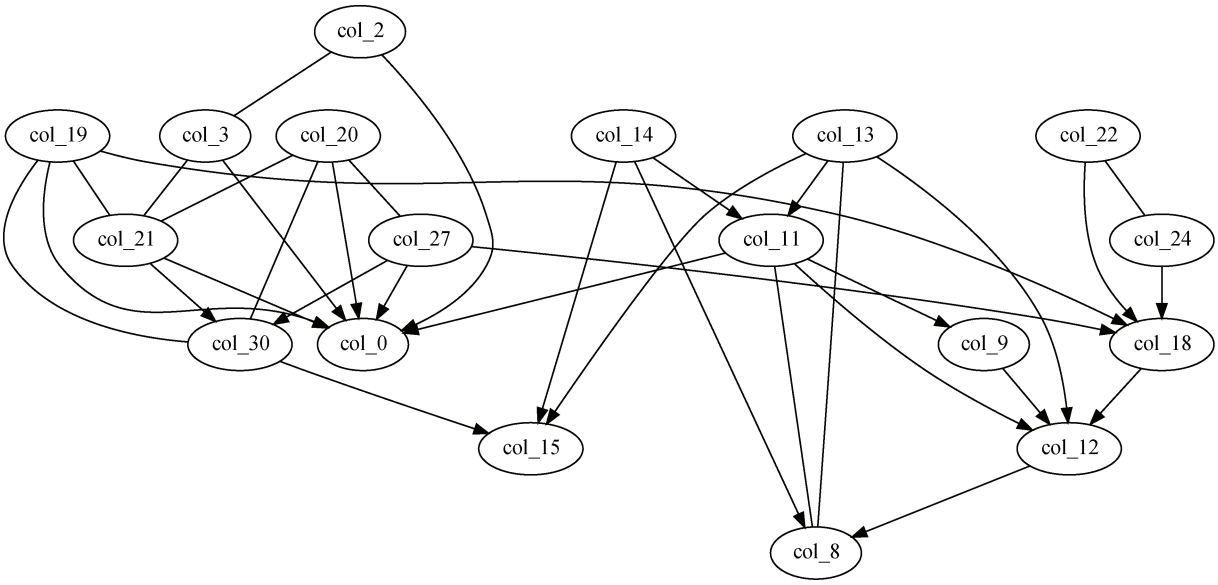

图 4　序列关系挖掘的部分结果(SMD-machine-2-8)

Figure 4　Part of relation mining result (SMD-machine-2-8)

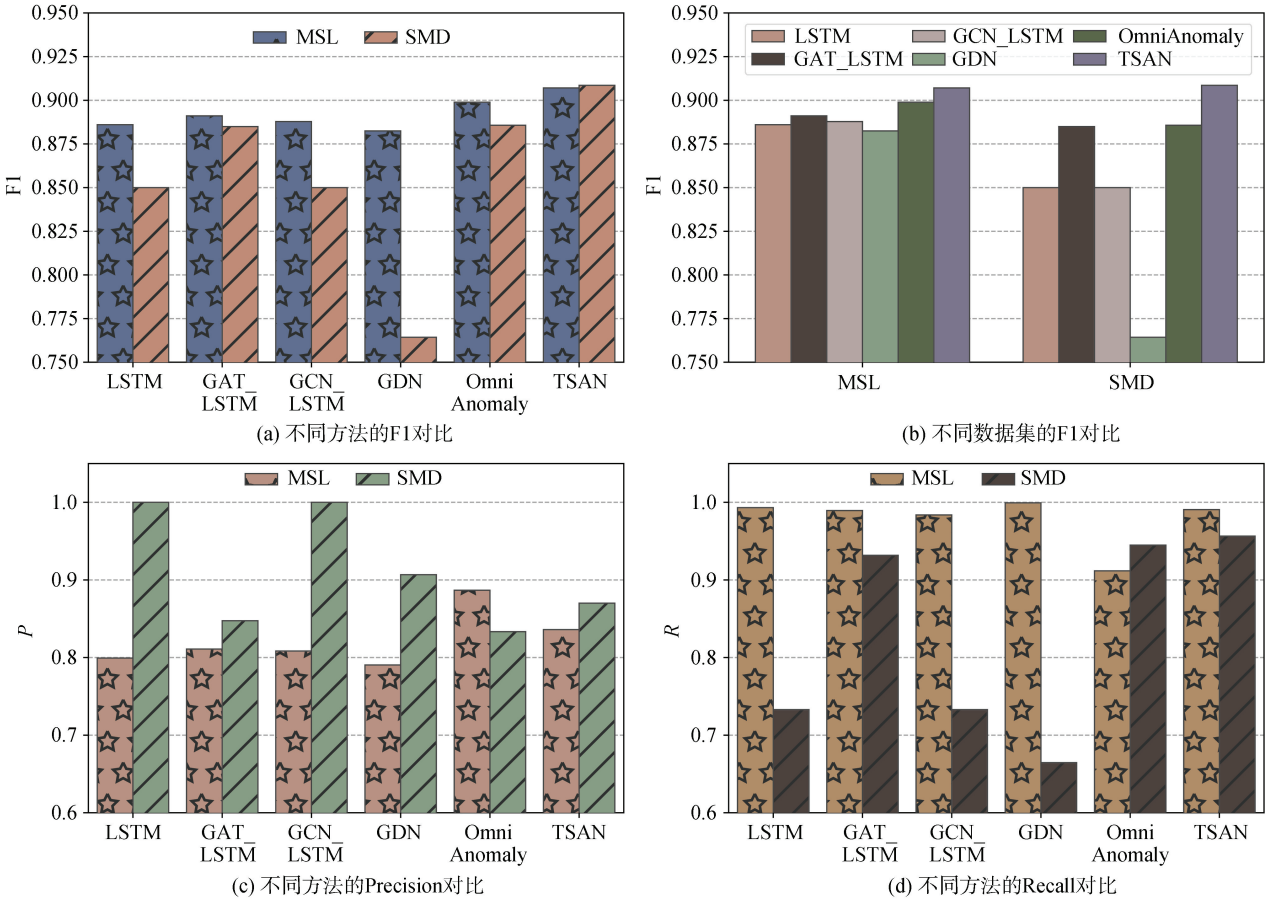

图 5　异常检测结果性能对比

Figure 5　Comparison of anomaly detection result

OmniAnomaly 和 TSAN 外, 其余方法的精准率都低于该平均值。但是, 在召回率指标上, OmniAnomaly 低于其他方法, 其他方法的召回率值均高于 0.980, GDN 取得最高的 0.999, TSAN 的召回率值为 0.990, 非常接近。

(2) 在数据集 SMD 上, TSAN 是唯一达到 0.9 以上的方法, 它的 F1 值相较于其他方法均有不小提升。相较于效果次优的 OmniAnomaly 的 0.885, 提升

表 3　TSAN 与基准方法的检测结果对比

Table 3　Performance of baselines and TSAN

数据集	方法	F1-score	Precision	Recall
MSL	LSTM	0.88604	0.79929	0.99312
	GCN_LSTM	0.88782	0.80842	0.98375
	GAT_LSTM	0.89107	0.81096	0.98937
	GDN	0.88245	0.79041	**0.99937**
	OmniAnomaly	0.89890	**0.88670**	0.91170
	TSAN	**0.90706**	0.83597	0.99062
SMD	LSTM	0.85000	**1.00000**	0.73291
	GCN_LSTM	0.85000	**1.00000**	0.73291
	GAT_LSTM	0.88495	0.84745	0.93167
	GDN	0.76428	0.90677	0.66459
	OmniAnomaly	0.88570	0.83340	0.94490
	TSAN	**0.90855**	0.87005	**0.95652**

约 2.3%, 相较于 GDN 的 0.764, 提升约 14.4%。虽然 LSTM 和 GCN_LSTM 取得了最好的精准率, 但是它们的召回率较低仅为 0.732, 最高的召回率为 TSAN 的 0.956。

(3) 从表 2 展示的时空上下文建模对比可知上下文递进关系: LSTM → GAT_LSTM → TSAN。方法的整体效果是不断提升的, 以综合指标 F1 为例, 不考虑序列间关系的 LSTM(MSL 为 0.886 和 SMD 为 0.850), 使用 GAT 挖掘空间上下文的 GAT_LSTM (MSL 为 0.891 和 SMD 为 0.884), 使用时空注意力机制处理时空上下文的 TSAN(MSL 为 0.907 和 SMD 为 0.908), 异常检测的性能改善明显。因此, 对于多维时间序列数据的挖掘, 设计合理的空间上下文处理方法是有必要的。

(4) 考虑时空上下文并且采用了注意力机制的 GAT_LSTM 和 TSAN 都取得了不错的 F1, 在两个数据集上都是优于或者接近次优。因此可以得出结论, 在 MSL 和 SMD 两个数据集中, 多维时间序列之间存在着影响关系, 即同时存在着时间和空间上下文的影响。在时空上下文的场景中, 注意力机制的应用提升了模型的异常检测性能。

取每个方法在两个数据集上取得的评价指标的差异进行比较。图 6 显示了 6 种方法在两个数据集上的指标变化。从性能波动的对比可以看出, TSAN 的性能变化最小, $\Delta F1 = 0.0015$, $\Delta P = 0.0340$, $\Delta R = 0.0341$。性能的变化说明如下:

(1) GAT_LSTM、OmniAnomaly 和 TSAN 在不同数据集上的同一指标差距较小, 具有较高的稳定性。而且不仅仅是 F1, 它们的精准率和召回率波动也很小。其余 3 种方法的同一指标差距较大, 稳定性稍差。这说明空间上下文对于时间序列是否异常的判断是有益的。

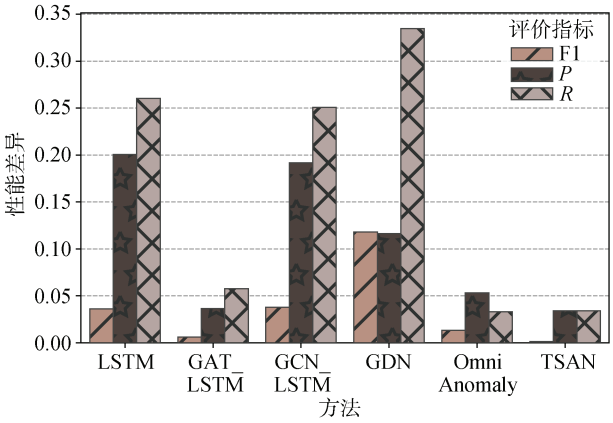

图 6　跨数据集异常检测性能变化对比

Figure 6　Performance variation of different methods across datasets MSL and SMD

(2) 从 LSTM 和 GCN_LSTM, LSTM 和 GAT_LSTM 两组对比来看, 它们在时间维度上均采用 LSTM, 当模型不处理空间上下文时, 波动较大。这说明处理空间上下文有助于提升模型的稳定性。

(3) 从 GCN_LSTM 和 GAT_LSTM 的对比来看, 在本文的应用场景下, 基于消息聚合并且采用注意力机制计算权重的 GAT 的稳定性高于 GCN。

(4) 从 LSTM, GAT_LSTM, TSAN 的递进角度来看, 它们在不同数据集上的指标差距越来越小, 因此在时间和空间维度上利用注意力机制处理上下文, 可以有效的提升模型的稳定性。

5.4　讨论

从实验结果可以看出, TSAN 适用于序列之间存在影响关系的多维序列。在其工作流程中, 异常检测分为序列预测和异常评分两个独立的阶段, 异常评

分并不会对序列预测产生影响。因此, TSAN 不仅可以用于异常检测, 还可以应用于多维序列的预测。

对于多维时间序列, TSAN 序列间关系的构建和应用包括两个方面: (1)基于节点相似度定义、应用 top-k 及阈值机制修剪以及通过节点因果图限制得到多维时间序列关系图; (2)在处理上下文的过程中从邻域节点融合数据时分配的注意力权重。

从节点关系的角度, 关系图从行为和属性特征决定了序列节点之间的关联关系, 这种关系在训练阶段是可学习的并且随着参数优化不断变化, 但对于数据集来说最终的结果是固定的, 不随时间变化而变化。从权重的角度, 时空注意力描述了节点间影响的强弱, 并且这种强弱关系是随时间动态变化的。通过分析具体时刻节点消息聚合分配权重的大小来判断异常的来源和原因, 为异常检测的溯源分析提供了一定的支持。注意力机制和因果关系图的限制使得挖掘到的序列关系具有一定的可解释性。

对象之间的关系可以归纳为显式关系和隐式关系两种。在本文的研究中, 并没有先验知识说明多维序列之间的影响关系, TSAN 使用相似性和因果推断挖掘序列间的关系, 从而完成隐式空间上下文的建模。当序列间存在已知关联时, 可以从中分析可能存在的空间上下文关系。比如, 服务器集群中, 各主机存在着网络连接, 数据传输方向可表示影响关系。这说明可以基于合理的场景来构建显式的空间上下文。无论是隐式还是显式的空间上下文, 在时空场景中, 使用时空注意力机制等方法对空间上下文进行分析都可以提升模型的性能和稳定性。

6 结论

本文提出了一种多维时间序列上的关系挖掘与异常检测方法 TSAN。方法以序列节点的相似性为基础, 结合剪枝策略和因果推断的关系作为约束, 构建了简洁有效的关系图, 并利用神经网络的特性实现了端到端的关系自动挖掘。在关系图序列上, 提出了基于时空注意力机制的预测模型, 通过融合时间上下文和关系上下文, 提高了异常检测结果的有效性和稳定性。从实验结果可以看出, TSAN 在测试数据集上取得了最优的 F1 指标, 相比次优方法分别提升了 0.9%(MSL)和 2.3%(SMD), 并且在所有对比方法中具有最小的跨数据集性能波动。这说明了 TSAN 对多维时序数据的异常检测是有效且稳定的。作为下一步工作, 我们将探索如何在具有显式和隐式混合空间上下文的多维时间序列场景中应用 TSAN, 并且优化提高 TSAN 在大数据集上的性能。

参考文献

[1] Ruff L, Kauffmann J R, Vandermeulen R A, et al. A Unifying Review of Deep and Shallow Anomaly Detection[J]. *Proceedings of the IEEE*, 2021, 109(5): 756-795.

[2] Soldani J, Brogi A. Anomaly Detection and Failure Root Cause Analysis in (Micro) Service-Based Cloud Applications: A Survey[J]. *ACM Computing Surveys*, 2023, 55(3): 1-39.

[3] 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: 102419.

[4] Garg S, Kaur K, Kumar N, et al. Hybrid Deep-Learning-Based Anomaly Detection Scheme for Suspicious Flow Detection in SDN: A Social Multimedia Perspective[J]. *IEEE Transactions on Multimedia*, 2019, 21(3): 566-578.

[5] Pajouh H H, Javidan R, Khayami R, et al. A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks[J]. *IEEE Transactions on Emerging Topics in Computing*, 2019, 7(2): 314-323.

[6] Chang yen-yu, Li P, Sosic R, et al. F-FADE: Frequency Factorization for Anomaly Detection in Edge Streams[C]. *The 14th ACM International Conference on Web Search and Data Mining*, 2021: 589-597.

[7] Navarro J, Deruyver A, Parrend P. A Systematic Survey on Multi-Step Attack Detection[J]. *Computers & Security*, 2018, 76: 214-249.

[8] Yao Y Y, Wang Z Q, Gan C, et al. Multi-Source Alert Data Understanding for Security Semantic Discovery Based on Rough Set Theory[J]. *Neurocomputing*, 2016, 208: 39-45.

[9] Thomas R, Pavithran D. A Survey of Intrusion Detection Models Based on NSL-KDD Data Set[C]. *2018 Fifth HCT Information Technology Trends*, 2018: 286-291.

[10] Wang M, Wang C K, Yu J X, et al. Community Detection in Social Networks[J]. *Proceedings of the VLDB Endowment*, 2015, 8(10): 998-1009.

[11] Kwak B I, Han M L, Kim H K. Cosine Similarity Based Anomaly Detection Methodology for the CAN Bus[J]. *Expert Systems with Applications*, 2021, 166: 114066.

[12] 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: 104-112.

[13] Hundman K, Constantinou V, Laporte C, et al. Detecting Spacecraft Anomalies Using LSTMS and Nonparametric Dynamic Thresholding[C]. *The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018: 387-395.

[14] Jove E, Casteleiro-Roca J L, Quintián H, et al. A New Method for Anomaly Detection Based on Non-Convex Boundaries with Random Two-Dimensional Projections[J]. *Information Fusion*, 2021, 65: 50-57.

[15] Vaswani N, Bouwmans T, Javed S, et al. Robust Subspace Learning: Robust PCA, Robust Subspace Tracking, and Robust Subspace Recovery[J]. *IEEE Signal Processing Magazine*, 2018, 35(4): 32-55.

[16] Gao H H, Qiu B Y, Barroso R J D, 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, 2023, 10(5): 2978-2990.

[17] Han H G, Zhang H J, Qiao J F. Robust Deep Neural Network Using Fuzzy Denoising Autoencoder[J]. *International Journal of Fuzzy Systems*, 2020, 22(4): 1356-1375.

[18] Geiger A, Liu D Y, Alnegheimish S, et al. TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks[C]. *2020 IEEE International Conference on Big Data*, 2020: 33-43.

[19] Li D, Chen D C, Jin B H, et al. MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks[M]. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019: 703-716.

[20] Bashar M A, Nayak R. TAnoGAN: Time Series Anomaly Detection with Generative Adversarial Networks[C]. *2020 IEEE Symposium Series on Computational Intelligence*, 2020: 1778-1785.

[21] Guha S, Mishra N, Roy G, et al. Robust random cut forest based anomaly detection on streams[C]. *International Conference on Machine Learning*, PMLR, 2016: 2712-2721.

[22] Schmidhuber J. Deep Learning in Neural Networks: An Overview[J]. *Neural Networks*, 2015, 61: 85-117.

[23] Ramaki A A, Khosravi-Farmad M, Bafghi A G. Real Time Alert Correlation and Prediction Using Bayesian Networks[C]. *2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology*, 2015: 98-103.

[24] Yan X D, Zhang H D, Xu X M, et al. Learning Semantic Context from Normal Samples for Unsupervised Anomaly Detection[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, 35(4): 3110-3118.

[25] Farzad A, Gulliver T A. Log Message Anomaly Detection with Fuzzy C-Means and MLP[J]. *Applied Intelligence*, 2022, 52(15): 17708-17717.

[26] Graves A. Long short-term memory[J]. *Supervised sequence labelling with recurrent neural networks,* 2012: 37-45.

[27] Ergen T, Kozat S S. Unsupervised Anomaly Detection with LSTM Neural Networks[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2020, 31(8): 3127-3141.

[28] Ding L Y, Fang W L, Luo H B, et al. A Deep Hybrid Learning Model to Detect Unsafe Behavior: Integrating Convolution Neural Networks and Long Short-Term Memory[J]. *Automation in Construction*, 2018, 86: 118-124.

[29] Geng X, Li Y G, Wang L Y, et al. Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, 33(1): 3656-3663.

[30] Lim B, Arık S Ö, Loeff N, et al. Temporal Fusion Transformers for Interpretable Multi-Horizon Time Series Forecasting[J]. *International Journal of Forecasting*, 2021, 37(4): 1748-1764.

[31] Kieu T, Yang B, Jensen C S. Outlier Detection for Multidimensional Time Series Using Deep Neural Networks[C]. *2018 19th IEEE International Conference on Mobile Data Management*, 2018: 125-134.

[32] Zhao H, Wang Y J, Duan J Y, et al. Multivariate Time-Series Anomaly Detection *via* Graph Attention Network[C]. *2020 IEEE International Conference on Data Mining*, 2020: 841-850.

[33] Su Y, Zhao Y J, Niu C H, et al. Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network[C]. *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019: 2828-2837.

[34] Deng A L, Hooi B. Graph Neural Network-Based Anomaly Detection in Multivariate Time Series[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, 35(5): 4027-4035.

[35] Xu C Q, Wang J L, Zhang J, et al. Anomaly Detection of Power Consumption in Yarn Spinning Using Transfer Learning[J]. *Computers & Industrial Engineering*, 2021, 152: 107015.

[36] Michau G, Fink O. Unsupervised Transfer Learning for Anomaly Detection: Application to Complementary Operating Condition Transfer[J]. *Knowledge-Based Systems*, 2021, 216: 106816.

[37] Gori M, Monfardini G, Scarselli F. A New Model for Learning in Graph Domains[C]. *Proceedings of 2005 IEEE International Joint Conference on Neural Networks*, 2005: 729-734.

[38] Scarselli F, Gori M, Tsoi A C, et al. The Graph Neural Network Model[J]. *IEEE Transactions on Neural Networks*, 2009, 20(1): 61-80.

[39] Zhou J, Cui G Q, Hu S D, et al. Graph Neural Networks: A Review of Methods and Applications[J]. *AI Open*, 2020, 1: 57-81.

[40] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. *Advances in neural information processing systems,* 2017, 30.

[41] Veličković P, Cucurull G, Casanova A, et al. Graph Attention Networks[EB/OL]. 2017: 1710.10903. https://arxiv.org/abs/1710.10903v3.

[42] Kalisch M, Bühlmann P. Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm[J]. *Journal of Machine Learning Research*, 2007, 8: 613-636.

[43] Kipf T N, Welling M. Semi-Supervised Classification with Graph Convolutional Networks[EB/OL]. 2016: 1609.02907. https://arxiv.org/abs/1609.02907v4.

胡智超 2013 年在哈尔滨工业大学软件工程专业获得硕士学位。现在哈尔滨工业大学网络空间安全专业攻读博士学位。研究领域为异常检测、入侵检测、数据安全。Email: hit.huzhichao@gmail.com

余翔湛 哈尔滨工业大学计算学部网络空间安全学院, 研究员。研究领域: 入侵检测、流量分类、数据安全。Email: yxz@hit.edu.cn

刘立坤 哈尔滨工业大学网络空间安全学院助理研究员, 博士。研究领域为网络数据流动监测、网络攻击检测与防御、浏览器指纹追踪与对抗。Email: liulikun@hit.edu.cn

张宇 哈尔滨工业大学网络空间安全学院副教授, 博导。研究领域为互联网基础设施安全、未来网络体系结构、互联网测量。Email: yuzhang@hit.edu.cn

于海宁 2013 年在哈尔滨工业大学信息安全专业获得博士学位。现任哈尔滨工业大学网络空间安全学院副研究员。研究领域为数据安全、隐私计算等。Email: yuhaining@hit.edu.cn