基于 Transformer 的时序数据异常检测方法

徐丽燕[1,2],徐　康[2,3*],黄兴挺[4],李熠轩[3],季学纯[1,2],叶　宁[3]
(1. 智能电网保护和运行控制国家重点实验室,江苏 南京 211106;
2. 南瑞集团(国网电力科学研究院)有限公司,江苏 南京 211106;
3. 南京邮电大学 计算机学院、软件学院、网络空间安全学院,江苏 南京 210003;
4. 南京邮电大学 贝尔英才学院,江苏 南京 210003)

摘　要:近年来,异常检测在电力系统运维、故障诊断等智能运维场景中起到关键作用。其中,深度学习在时序数据异常检测上取得了成功的应用。然而,基于长短期记忆(Long Short-Term Memory,LSTM)等异常检测方法因其序列学习模式中包含递归运算,导致模型难以并行计算,同时长期依赖性会导致模型性能下降。因此,提出了一种基于 Transformer 的时序数据异常检测方法,利用自注意力机制并行训练数据捕获内部有效信息,利用编码-解码框架使用端到端的方式通过时序数据生成异常得分。这个方法能更完整地提取时序数据的上下文关系,精确地捕获时序数据的异常关系。经实验证明,基于 Transformer 的时序数据异常检测方法在 WADI、SWaT、KDDCUP99 与 AIOPS18 等数据集上的异常检测表现出比其他方法更优的性能。

关键词:异常检测;多头自注意力;Transformer;时间序列;深度学习
中图分类号:TP18　　**文献标识码**:A　　**文章编号**:1673-629X(2023)03-0152-09
doi:10.3969/j.issn.1673-629X.2023.03.023

Transformer-based Method of Anomaly Detection for Time Series Data

XU Li-yan[1,2],XU Kang[2,3*],HUANG Xing-ting[4],LI Yi-xuan[3],JI Xue-chun[1,2],YE Ning[3]
(1. State Key Laboratory of Smart Grid Protection and Control,Nanjing 211106,China;
2. NARI Group Corporation/State Grid Electric Power Research Institute,Nanjing 211106,China;
3. School of Computer Science,Nanjing University of Posts and Telecommunications,Nanjing 210003,China;
4. Bell College of Excellence,Nanjing University of Posts and Telecommunications,Nanjing 210003,China)

Abstract:In recent years,anomaly detection plays an important role in the scenarios of Artificial Intelligence for IT Operations (AIOps),such as power grid operation and maintenance,fault diagnosis. With the advantages of universality and strong learning ability,deep learning is widely used in anomaly detection on time series data. However,how to efficiently learn the relationship between a single anomaly and global information of time series data is still an urgent problem to be solved. The sequential learning model like long short-term memory (LSTM) is difficult to compute in parallel due to the large number of recursive operations,while long-term dependence can lead to degradation in model performance. In this paper,we propose a method of anomaly detection for time series data with Transformer. The self-attention mechanism is used to parallelly train the data to capture the effective information,and the encoder-decoder framework is used to transform the time series data into anomaly score in an end-to-end way. This method can not only learn the context of time series data more completely,but also make full use of the feature matrix to conduct anomaly detection. Experiment shows that anomaly detection method of time series data based on transformer has the best performance compared to the baseline methods in the datasets of WADI,SWaT,KDDCUP99 and AIOPS18.

Key words:anomaly detection;multi-head self-attention;Transformer;time series;deep learning

收稿日期:2022-05-22　　**修回日期**:2022-09-23
基金项目:智能电网保护和运行控制国家重点实验室开放课题项目(SGNR0000KJJS2007626);南京邮电大学高水平师资(NY218118,NY219104)
作者简介:徐丽燕(1983-),女,博士,高级工程师,研究方向为电力系统自动化、智能运维等;通信作者:徐　康(1989-),男,博士,CCF 会员(55718M),讲师,研究方向为智能运维、知识图谱和自然语言处理。

0 引　言

为保障网络系统的正常运行,运维人员需要时刻监控海量的数据。其中,关键性能指标(Key Performance Indicator,KPI)异常检测是一个底层核心任务[1],其目标是根据 KPI 变化曲线,从大量数据中找出与预期不符的离群点,也就是系统软硬件服务的异常。随着服务系统的大型化与复杂化,需要监控的关键性能指标呈现多样化、规模化的趋势。利用人工智能技术和机器学习算法对大型网络系统进行异常检测成为智能运维技术发展的必然趋势[2]。深度学习算法利用神经网络自动学习特征,凭借强学习能力与高适应力的优势在异常检测领域变得越来越流行,并已应用于各种任务。研究表明,在异常检测领域,深度学习[3]方法已经完全超越传统方法[4]。

传统深度异常检测方法通过搭建神经网络自动化地完成特征提取并对异常进行量化评估。例如,基于自编码器的方法[5]通过自编码器重构原始数据的误差,评估异常值;基于对抗生成网络的方法[6]根据生成器模拟原始数据,通过原始数据与模拟数据的误差以及判别器的值协同评估数据的异常值。而各种神经网络框架如循环神经网络[7](Recurrent Neural Network,RNN)、长短期记忆网络[8](Long Short-Term Memory,LSTM)在异常检测中的应用,一定程度解决了异常检测时间序列在时间上的相关性问题。

更复杂的组合模型也被应用于时序数据异常检测领域。基于 RNN、LSTM 等循环神经网络和编码-解码框架的网络能捕获长短期的时间序列的趋势[9-10],但 RNN、LSTM 等方法仍然难以很好地捕获时序中的长距离依赖,并且难以并行计算[11]。Li 等人提出了基于生成对抗网络(Generative Adversarial Networks,GAN)的 MAD-GAN 模型[12],这类基于 GAN 重构数据的模型相较于自编码器具有更好的生成能力,但对抗生成网络存在着模式崩溃等问题。最近,图神经网络(Graph Neural Network,GNN)也被应用于多元时间序列[13],通过图结构建模变量之间的关系来预测某个节点的值[14],但这类方法过于依赖于单个时间点之间的关联,而没有有效利用序列的上下文信息。

尽管已经提出了很多基于深度学习的异常检测方法,但现有的方法仍有两点不足:

(1)现有的深度学习方法建模序列数据时因其长期依赖性导致模型性能下降。

(2)现有的时序深度学习模型中因其包含递归运算导致模型难以并行计算。

针对以上两点不足,提出一种基于 Transformer[15]的时序数据异常检测方法(Devformer)。利用多头自注意力机制[16]并行化地处理当前上下文环境中的全部数据,自动化地提取每个时间节点数据的长距离依赖特征;利用编码-解码框架[17],建立一个端到端的异常得分模型,得到每个时刻异常倾向的量化评估。

相较已有方法,这篇文章的主要贡献有:

(1)将 Transformer 引入异常检测领域,有效解决异常检测时间序列上下文依赖与神经网络并行化运算问题。

(2)在 WADI、SWaT、KDDCUP99 与 AIOPS18 数据集上的异常检测实验中,Devformer 表现优于现有的时序数据异常检测的对比方法。

1 相关工作

这部分将主要介绍时序数据的异常检测与 Transformer。

1.1 时序数据的异常检测

异常监测是对数据集中不匹配预期模式的离群点进行识别,属于不均衡数据的分类问题[18]。异常检测的核心就在于建立一个合适的模型,自动学习数据的深层次特征,得到正常和异常的决策边界,以区分正常和异常样本。

时序数据指以一定的时间间隔持续捕获的一系列指标数据。网络系统的流量变化,银行的量化交易,工业物联网的环境监控都具体表现为时间序列数据,是系统正常工作与性能变化的最直观反映。因此,通过监测时序数据的变化,实现异常检测是智能运维领域的重要内容。基于时序数据普遍表现出数据量大、监测指标多、异常种类多样的特点,更适合使用学习能力强的深度学习方法实现时序数据的异常检测。

传统的统计学方法通过移动差分自回归[19]等模型分解并平稳化时间序列,异常检测具有明显季节性与趋势性的非平稳时间序列,但由于数据维度灾难的原因,难以捕捉高维或非线性关系,对异常与正常分界模糊的数据分类困难。支持向量机[20]、决策树和随机森林[21]的分类方法与 XGBoost[22]等回归算法解决了上述问题,较快且较好地拟合非线性或高维数据,一定程度避免了过拟合的问题,提高了模型的泛化能力。但传统的关键性能指标异常检测方法并没有充分考虑时间序列的上下文信息,难以捕获异常对全文信息的依赖。

在基于机器学习的异常检测方法中,需要对原始数据进行特征工程,提取样本特征。特征的选取决定了异常检测的效果,但却取决于相关领域的专家经验,导致检测效果并不稳定。同时,异常检测时间序列数据量大且异常种类多的特点,决定了机器学习算法在异常检测领域的局限性。因此,引入对大规模高维数据、非线性参数学习能力更强的深度学习异常检测

方法。

1.2 Transformer

Transformer 作为一种新型的神经网络框架,利用自注意力机制学习数据的长距离依赖关系,利用编码-解码框架实现输入与输出的端到端运算。

注意力是有限的,并且数据中总有冗余信息。Transformer 的自注意力机制解决了如何将有限的注意力聚焦在有效信息。自注意力机制捕获数据的长距离依赖关系,以并行运算模式使数据对不同距离数据的依赖关系学习能力相同,解决了 LSTM 等方法因包含递归运算而导致模型难以并行计算,同时长期依赖性会导致模型性能下降的问题。

Transformer 最初被应用于机器翻译[15],用于学习句内单词的相关性。Transformer 框架凭借学习长距离依赖关系的优势,较好地适应了时间序列中当前数据值依赖于历史数据的特征。同时,Dai 等人引入段级递归机制,有效地解决了 Transformer 并行运算机制中分批次投入导致的上下文碎片问题[23]。各项研究表明,Transformer 在序列数据处理任务表现出较优的性能。Wu 等人通过改变编码器的输入与解码器的输入、输出模式将 Transformer 用于时序数据的预测[24];Bryan Lim 等人在时序预测任务中,将 LSTM 应用于 Transformer 的编码器并结合门限机制提高了时序预测的精度[25]。

而近年来,Transformer 也被应用于时序数据异常检测的任务中。Chen 等人在构建的模型中使用 Transformer 架构建模时序数据的依赖关系[26];Li 等人提出的基于生成对抗网络的模型也使用了 Transformer 组件构成 GAN 模型的生成器,来挖掘时序数据的深层信息[27];Wang 等人将 Transformer 与变分自编码器(Variational Auto-Encoder, VAE)结合起来,在多元时序数据上进行异常检测[28];Xu 等人同样通过 Transformer 对每个时间点计算自注意力权重来得到点与序列之间的关系[29];这些工作展现了 Transformer 在时序数据处理上的优秀性能。

2 基于 Transformer 的异常检测方法

时序型的多维 KPI 数据及其对应的异常标签为本模型的输入和输出。该文使用滑动窗口方法处理原始时序数据获取窗口数据。

本模型输入的时间序列数据处理为 T 个长度为 m 的滑动窗口,每个滑动窗口数据记作 $\mathbf{Series}_t$,为模型输入的基本单位。以图 1 为例,模型将 5 条数据处理为 3 个长度为 3 的滑动窗口。本模型的编码器将输入的 $\mathbf{Series}_{1\cdots T}$ 编码为中间表示 $\boldsymbol{C}_{1\cdots T}$,$\boldsymbol{C}_t = f_e(\boldsymbol{x}_{t1}, \boldsymbol{x}_{t2}, \cdots, \boldsymbol{x}_{tm})$,其中 $\boldsymbol{x}_{t1}, x_{t2}, \cdots, \boldsymbol{x}_{tm}$ 为 $\mathbf{Series}_t$ 的每个时间点的多维 KPI 向量数据,$f_e(\cdots)$ 为 Transformer 编码器[15]。接着,使用解码器 $\varphi_t = f_d(C_t, \mathbf{Series}_t)$,将中间表示 C_t 和 $\mathbf{Series}_t$ 作为 Transformer 解码器 $f_d(\cdots)$ 的输入,输出为异常得分 $\boldsymbol{\Phi}_t$。使用分数生成器生成规范化的异常得分 $\mathrm{dev}(\boldsymbol{\Phi}_t)$,模型假设 $\mathbf{Series}_t$ 的异常得分满足高斯分布,其中 Margin 为正常值和异常值的阈值,μ 和 σ 为高斯分布的均值和标准差。$\mathbf{Series}_t$ 中的异常得分表示时间窗口内最后一个时间点数据的异常得分。

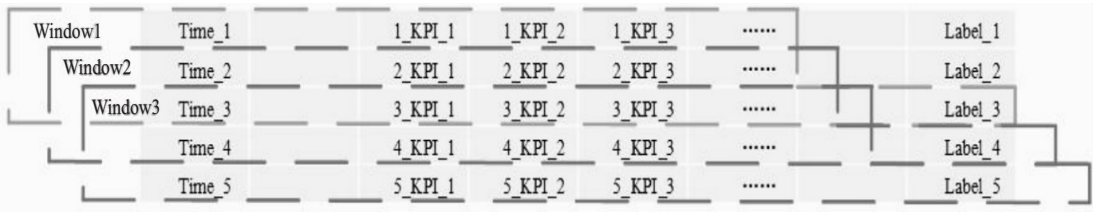

图 1 滑动窗口预处理

本模型训练基于 Transformer 的异常检测模型,主要分为三个模块:(1)编码器模块,输入为预处理后的窗口数据,输出数据的中间表示;(2)解码器模块,输入为预处理后的窗口数据和编码器的中间表示,输出窗口数据的初始异常分数;(3)分数生成器,将初始异常分数通过分数生成器生成规范化的异常得分,最后通过阈值判别数据中的异常,通过该异常得分模型,当滑动窗口的异常得分 $\boldsymbol{\Phi}_t$ > Margin 时,判断滑动窗口的最后一条数据为异常。具体流程如图 2 所示。

2.1 Devformer 编码器

Devformer 编码器主要分为多头自注意力机制,残差与归一化网络两个模块。

2.1.1 多头注意力机制

在 Devformer 的编码器中,首先将 $\mathbf{Series}_t$ 投入多头自注意力网络,捕捉内部序列的依赖关系,计算 $\mathbf{Series}_t$ 内部注意力矩阵 $\boldsymbol{A}_t^e$。

注意力机制有三个输入:**Query**,**Key**,**Value**。由 **Query** 索引键值 **Key**,计算两者相关性,再加权 **Value** 得到注意力矩阵。

$$\mathrm{Attention}(\mathbf{Query}, \mathbf{Key}, \mathbf{Value}) = \mathrm{softmax}\left(\frac{\mathbf{Query} \times \mathbf{Key}^{\mathrm{T}}}{\sqrt{d_k}}\right) \times \mathbf{Value} \quad (1)$$

定义三个矩阵 $\boldsymbol{W}_Q$、$\boldsymbol{W}_K$、$\boldsymbol{W}_V$ 分别作为 **Query**、**Key**、**Value** 取相同值 $\mathbf{Series}_t$ 时的权重,注意力机制能够捕

捉序列内部的依赖关系。

$$\boldsymbol{A}_t^e = \text{Attention}(\boldsymbol{W}_Q\textbf{Series}_t, \boldsymbol{W}_K\textbf{Series}_t, \boldsymbol{W}_V\textbf{Series}_t) \tag{2}$$

将多维的时序数据切片分别训练,并将多次注意力机制结果用前馈神经网络拼接。多头的机制能够提高模型的训练效率,有助于捕捉更丰富的特征。

$$\text{Multihead}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{concat}[\textbf{head}_1,\cdots,\textbf{head}_n]\boldsymbol{W}_o \tag{3}$$

$$\textbf{head}_i = \text{Attention}(\boldsymbol{Q}_i,\boldsymbol{K}_i,\boldsymbol{V}_i) \tag{4}$$

其中, n 为头的数量, i 为头的索引。

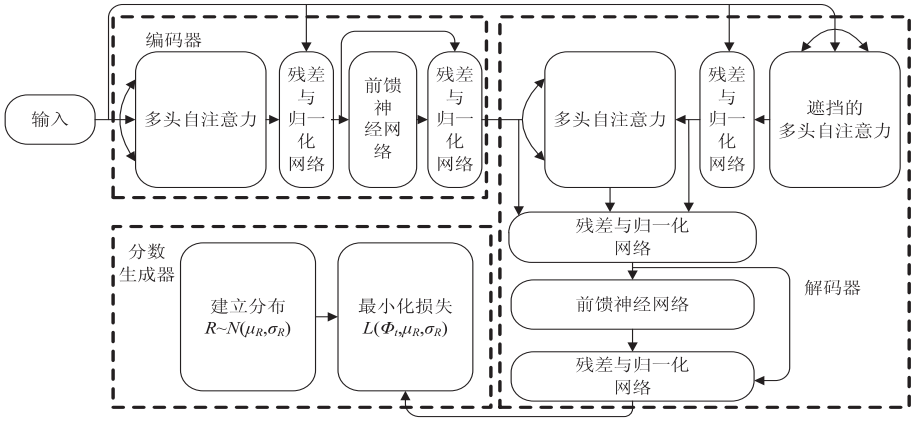

图 2　基于 Transformer 的时序数据异常检测框架

2.1.2　残差与归一化网络

Devformer 的编码器中,在多头自注意力模块后添加残差与归一化网络,将多头自注意力网络的输入 $\textbf{Series}_t$ 与输出 $\boldsymbol{A}_t^e$ 叠加,解决退化问题,使学习的自注意力矩阵更稳定。

残差网络将 $\textbf{Series}_t$ 与 $\boldsymbol{A}_t^e$ 叠加,通过增加恒等映射,在不影响学习效果的前提下,一定程度解决梯度消失问题。

$$\boldsymbol{A}_t^e = \boldsymbol{A}_t^e + \textbf{Series}_t \tag{5}$$

归一化网络基于 $\textbf{Series}_t$ 的均值与方差,不断调整神经网络中间输出,使编码器输出 $\boldsymbol{C}_t$ 更加稳定,起到正则化作用。

$$\boldsymbol{A}_t^e = \text{LayerNorm}(\boldsymbol{A}_t^e) \tag{6}$$

在前馈神经网络后添加残差与归一化网络有同样的效果。

2.2　Devformer 解码器

Devformer 解码器主要分为后遮挡的自注意力机制、残差与归一化网络、注意力解码机制三个模块。

$\textbf{Series}_t$ 投入后遮挡的自注意力网络,学习遮挡的注意力矩阵 $\boldsymbol{A}_t^e$。将注意力矩阵的上三角置 0,使第 i 条数据的异常得分只受前 $i-1$ 条数据影响, $\textbf{Series}_t(1 \leqslant t \leqslant T)$ 的异常得分只受 $\{\textbf{Series}_1,\textbf{Series}_2,\cdots,\textbf{Series}_{t-1}\}$ 的影响,防止数据泄露,如公式(7)、(8)所示。

$$\text{MaskedAttention}(\textbf{Query},\textbf{Key},\textbf{Value}) = \text{softmax}\left(\frac{\textbf{Query}\times\textbf{Key}^{\mathrm{T}}}{\sqrt{d_k}} + \textbf{mask}\times-\text{inf}\right)\times\textbf{Value} \tag{7}$$

$$\boldsymbol{A}_t^d = \text{MaskedAttention}(\boldsymbol{W}_Q\textbf{Series}_t, \boldsymbol{W}_K\textbf{Series}_t, \boldsymbol{W}_V\textbf{Series}_t) \tag{8}$$

其中, **mask** 为与 $\textbf{Series}_t$ 相同维度的单位上三角矩阵, $-\text{inf}$ 的含义为负无穷。在进入激活函数前, **Query** 和 **Key** 的点积结果需要先经过 **mask** 层。**mask** 层将注意力矩阵的上三角置为负无穷,对应激活后的注意力系数趋于 0,使当前时刻数据的异常得分与未来时刻无关。

取输出 $\boldsymbol{A}_t^d$ 作为注意力机制 **Query**,取编码器输出 $\boldsymbol{C}_t$ 作为注意力机制的 **Key** 与 **Value**,学习 $\boldsymbol{C}_t$ 对 $\boldsymbol{A}_t^d$ 的注意力分配矩阵。

$$\boldsymbol{C}_t^d = \text{Attention}(\boldsymbol{W}_Q\boldsymbol{A}_t^d, \boldsymbol{W}_K C_t, \boldsymbol{W}_V\boldsymbol{C}_t) \tag{9}$$

最后通过前馈神经网络进行降维处理,输出单维的 Φ_t,作为 $\textbf{Series}_t$ 最后一条数据 x_{tm} 异常得分。

$$\varphi_t = \text{ReLU}(\text{Linear}(\boldsymbol{C}_t^d)) \tag{10}$$

解码器中的每个注意力网络与前馈神经网络都加入了残差与归一化网络。

2.3　Devformer 分数生成器

Devformer 以多头自注意力机制的编码-解码框架学习异常得分之后,假设所有数据的异常得分满足高斯分布,构造一个损失函数。基于异常检测时间序列异常数量少而种类多的特征,该文采用偏差网络框架[30],通过高斯分布获取参考分数,建立基于 Z 得分的损失函数。

Devformer 由解码器获得异常得分 $f_d(\boldsymbol{C}_t,\textbf{Series}_t)$ 后,采用先验驱动的方法,从标准高斯分布中随机抽取一定数量的样本,并取样本的平均值、标准差作为参考分数。

$$r_1,r_2,\cdots,r_m \sim N(\mu_R,\sigma_R) \tag{11}$$

$$\mu \approx \mu_R = \frac{1}{m}\sum_{i=1}^{m} r_i \tag{12}$$

$$\sigma \approx \sigma_R = \frac{1}{m}\sum_{i=1}^{m}(r_i - \mu_R)^2 \tag{13}$$

其中，$r_i(1 \leqslant i \leqslant m)$ 为随机选取的一组数据的异常得分，m 为数据量，μ_R 为先验数据集参考分数的数学期望，σ_R 为参考分数的方差。

$$\mathrm{dev}(\varphi_t) = \frac{\varphi_t - \mu_R}{\sigma_R} \tag{14}$$

引入 $L(\varphi_t, \mu_R, \sigma_R)$ 作为损失函数：

$$L(\varphi_t, \mu_R, \sigma_R) = (1-y)\,|\,\mathrm{dev}(\varphi_t)\,| + y\max(0, a - \mathrm{dev}(\varphi_t)) \tag{15}$$

其中，y 为 $\mathbf{Series}_t$ 最后一条数据的异常标记。

在基于 Z 得分建立的损失函数中，正常数据的异常得分会向 0 靠近，异常数据的异常得分会远离 0，或保持大于 a 的值不变，以此实现了正常数据和异常数据异常得分的区分。在异常检测过程中，时序数据 x_t 投入已训练的模型，将输出对应的异常得分 $\boldsymbol{\Phi}_t$。将异常得分在 $[0, a]$ 范围内的数据归为正常，否则为异常。

3 Devformer 模型训练方法

算法 1 给出了基于偏差网络的 Transformer 算法的异常得分模型建立方法。

算法 1：Devformer 模型训练方法。

输入：n 维的异常检测时间序列 $\boldsymbol{X} \in \boldsymbol{R}^n$ 的集合；

输出：异常评分网络 $\boldsymbol{\Phi}_t \in \boldsymbol{R}$。

预处理数据集建立 T 个长度为 m 的滑动窗口集合 $\{\mathbf{Series}_t\}$；/* m 为滑动窗口长度 */

(1) 随机初始化编码器向量 $\{\boldsymbol{C}_t\}$；

(2) for i = 1 to epochs do

(3) for j = 1 to nb_batch do

(4) 从训练集随机抽取 batch_size 个数据投入训练；

(5) 将 m 维序列切成 n 片，每片 m/n 维，分别训练后将结果拼接；/* m 为 n 的整数倍 */

(6) 编码器编码所有的 $\mathbf{Series}_t$ 为中间向量 $\boldsymbol{C}_t$；

(7) 解码器解码 $\mathbf{Series}_t$、$\boldsymbol{C}_t$，获得异常得分 $\boldsymbol{\Phi}_t$

(8) 梯度下降最小化损失函数 $L(\varphi_t, \mu_R, \sigma_R)$；

(9) end for

(10) end for

(11) return 0

在数据集中随机采集固定数量的正常数据和异常数据，保证模型单次投入的数据组中正常数据和异常数据数量相等，训练足够数量的异常数据，解决时序数据集异常类别不平衡的问题。

4 实验结果与分析

该文由 Python3.8 与 Tensorflow2.3.0 搭建了基于 Transformer 的时序数据异常检测模型，并利用 SWaT、WADI 和 KDDCUP99、AIOPS18 四个数据集评价该模型的性能。表 1 为数据集基本信息。

表 1 数据集基本信息

Dataset	SWaT	WADI	KDDCUP99	AIOPS18
Dimension	51	123	34	6
Training Size	16 097	9 075	21 242	53 408
Testing Size	9 700	6 208	12 186	37 264
Train Rate	0.277 2	0.257 0	0.179 9	0.069 3
Test Rate	0.074 3	0.136 9	0.516 4	0.106 0

表 1 中，Train Rate 表示训练集中异常数所占比例，Test Rate 表示测试集中异常数所占比例。

SWaT 与 WADI 数据集分别来自新加坡科技与设计大学网络安全中心的配水系统与水安全处理系统的攻击检测。SWaT 数据集的 KPI 来自对应系统的网络流量与 51 个传感器与执行器的数据，WADI 数据集的 KPI 来自配水系统 123 个传感器与执行器的数据。

KDDCUP 99 数据集来自 1999 年 KDDCUP 竞赛的比赛数据，模拟了 US 空军局域网环境，并监控了 34 类关键性能指标，其异常来自局域网受到的模拟攻击；AIOPS18 数据集来自 2018 年 AIOPS 挑战赛的比赛数据集，该数据集由时间戳、一维的关键性能指标与异常标记组成，体现了智能运维在时间序列异常检测领域的代表形式。

上述数据集，长时间、短间隔地记录多维的关键性能指标（如网络流量、传感器与执行器的所有值）与异常检测标签，符合模型异常检测的时序要求与深度学习的大数据量与高维数据的要求。

实验中，偏差网络损失函数的置信度边界值（confidence margin）设置为 5.0，高斯分布数据量设置为 5 000 时，实验结果较优。

设置模型训练次数 epochs 为 800，单次训练投入数据组批数 nb_batch 为 10，一批数据组的数据量 batch_size 为 30。

实验使用准确率（Precision）、召回率（Recall）和 F1 分数衡量模型异常检测的效果：

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{16}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{17}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{18}$$

其中,TP 为正确检测的异常数,FP 为检测为异常的正常数,FN 为检测为正常的异常数。

异常检测的目标是尽可能完全地检测出异常,在进行实验与模型评估时,更加重视模型在召回率指标上的表现。因此,在保证 F1 分数更高的同时,该文主要使用召回率作为模型的性能衡量指标。

4.1　异常检测结果分析

将基于偏差网络的 Transformer 异常检测方法分别与主成分分析异常检测方法(PCA)、随机森林异常检测方法(Random Forest)、基于偏差网络的长短期记忆(LSTM)异常检测方法、文献[12]中基于生成对抗网络的重构方法 MAD-GAN、文献[30]中基于偏差网络的深度学习模型 Devnet、文献[14]中基于图注意力网络的 GDN 模型进行比较。其中,MAD-GAN 将数据映射到潜在空间,并通过潜在空间重构数据,计算重构损失得到异常得分;Devnet 结合高斯先验分布计算偏差损失,通过端到端方式直接优化模型;GDN 将数据的每个特征维度作为图神经网络的一个节点,通过节点相似度学习图结构,结合注意力机制计算每个节点的异常得分。

具体实验结果如表 2 所示,表 2 中加粗部分表示单个数据集中最优的性能。

表 2　不同方法的性能比较

Datasets	Methods	Precision	Recall	F1
SWaT	PCA	0.571 4	0.800 0	0.666 6
	RandomForest	0.974 4	0.550 7	0.703 7
	LSTM	0.603 2	0.778 1	0.679 6
	MAD-GAN	0.833 3	0.454 5	0.588 2
	Devnet	0.999 7	0.778 1	0.875 1
	GDN	0.949 9	0.641 5	0.765 8
	Devformer	0.700 6	**0.850 2**	0.768 2
WADI	PCA	0.920 0	0.605 3	0.730 1
	RandomForest	0.571 4	0.965 5	0.717 9
	LSTM	0.487 4	0.780 0	0.599 9
	MAD-GAN	0.135 6	0.727 3	0.228 9
	Devnet	0.471 2	0.202 1	0.282 9
	GDN	0.864 7	0.355 9	0.505 3
	Devformer	0.423 5	**0.982 4**	0.591 9
KDDCUP 99	PCA	0.923 1	0.461 5	0.615 4
	RandomForest	0.888 9	0.186 0	0.307 6
	LSTM	0.603 2	0.778 1	0.679 6
	MAD-GAN	0.962 8	0.710 6	0.817 8
	Devnet	0.908 8	0.951 3	0.929 6
	GDN	0.803 1	**0.999 7**	0.890 8
	Devformer	0.998 2	0.955 9	0.976 6
AIOPS18	PCA	0.166 7	0.200 0	0.181 8
	RandomForest	0.342 9	0.444 4	0.387 1
	LSTM	0.176 4	0.280 8	0.216 7
	MAD-GAN	0.285 7	0.363 6	0.320 0
	Devnet	0.275 4	0.306 7	0.290 2
	GDN	0.393 1	0.329 9	0.358 7
	Devformer	0.504 5	**0.470 6**	0.487 0

实验结果表明,相较于其他方法,Devformer 能够较好地分析出时间序列的长距离依赖关系,能够全面检测出存在的异常,在召回率方面有着显著的优势。

对于 SWaT 数据集,Devformer 拥有最高的召回率,相较于召回率第二的主成分分析方法,提升了高达 0.050 2 的得分,提升率达 6.275%;同样,在特征数量更多的 WADI 数据集上,Devformer 的召回率依然最高,达到了 0.982 4,相比于其他对比方法,召回率得分有着平均 0.376 4 的提升;对于 KDDCUP99 数据集,Devformer 的召回率也超过了 0.95,在保证可观的召回率的同时,还拥有出色的准确率(得分超过 0.99),F1 值也超过了 0.97,相较于其他方法,三项指标均为

最高水准,体现出了模型优秀的综合性能;同样,在 AIOPS18 数据集上,Devformer 的三项指标均为最高,在 Recall 值上有着平均 0.194 7 的提升。

经分析发现,Devformer 的性能在大规模数据集中更优。并且,Devformer 在特征维度较高的 WADI 与仅包含六个特征的 AIOPS18 数据集上,都有着优秀的性能表现,说明了该模型对不同的特征维度均具有良好的适应性。Devformer 在 WADI 数据集中准确率过低,在 SWaT 数据集中召回率不够高,主要因为训练数据量不足。各深度方法对于 AIOPS18 数据集的训练效果均不理想,是因为该数据集的关键性能指标维度过低导致的数据中包含的特征信息过少,可以通过抽样重复训练的方法提升训练效果。

4.2 不同窗口大小下 Devformer 性能分析

Devformer 中窗口大小(即序列长度)通过影响模型捕获单个异常对其他数据依赖的最大距离,而对模型性能影响巨大。为了分析不同序列长度对模型准确率、召回率与 F1 的影响,本实验控制其他变量不变,通过设置不同的序列长度,分析序列长度对模型性能的影响。实验结果如图 3 所示。

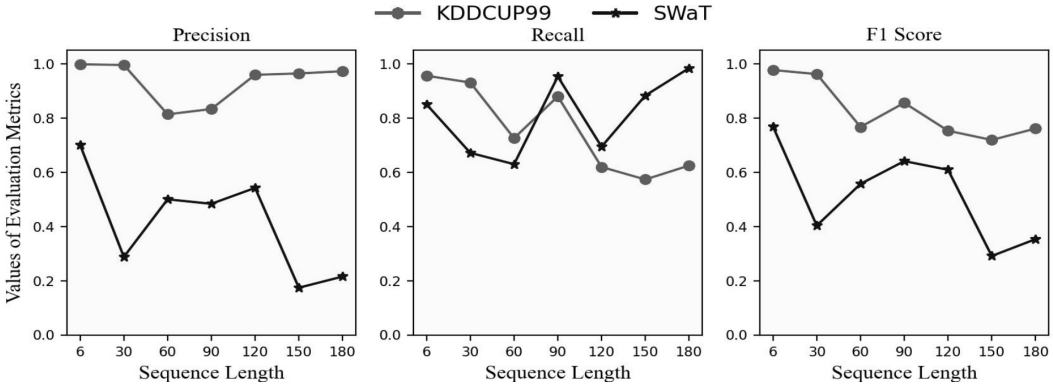

图 3 序列长度对模型性能的影响

4.3 Devformer 的鲁棒性实验

在实际应用中,训练用数据集是有噪声的,即无法将所有的异常标注。Devformer 的偏差网络更关注异常并重复训练阳性数据,减少了假阴型样本的干扰。为了证明 Devformer 对假阴型数据的抗干扰能力,在训练集中分别增加 0%、2%、5%、10%、20% 的假阴性样本,比较 Devformer 分别与主成分分析异常检测方法(PCA)、随机森林异常检测方法(Random Forest)、文献[12]的 MAD-GAN、文献[30]中的 Devnet、基于偏差网络的长短期记忆(LSTM)异常检测方法在有噪声的 KDD 训练集中的性能。实验结果如图 4 所示。

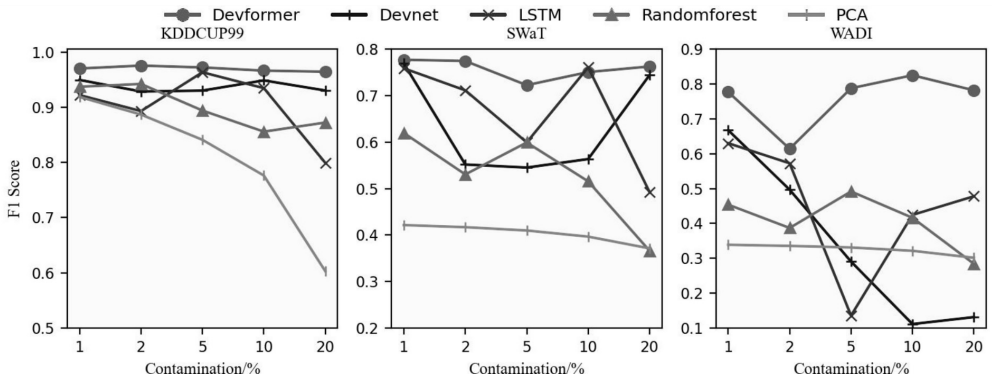

图 4 污染训练集对模型性能的影响

实验证明,Devformer 具有较好的鲁棒性,即使训练集中有较多的假阴型样本,对模型性能也不会有明显影响。这是因为 Devformer 采集大量正常数据的异常得分建立正态分布作为参考分数,即使数据中有少量假阴型样本作为异常数据,对模型学习到的分布影响不大。

4.4 不同 Transformer 层数下 Devformer 性能分析

在 Devformer 中设置了不同数量的 Transformer 层,模型表现出不同的性能,图 5 显示了三个数据集 WADI、SWaT 和 KDDCUP99 在不同 Transformer 层数的模型中的性能变化。

实验结果表明,该模型在 SWaT 数据集、KDDCUP99 数据集以及 WADI 数据集 Transformer 层数分别设置为 3 层、5 层和 1 层时表现最佳。

当 Transformer 层数过多时,神经网络过于复杂,会导致过拟合的问题;Transformer 层数过少,可能导致简单的神经网络难以学习高维度数据中复杂的异常模式。

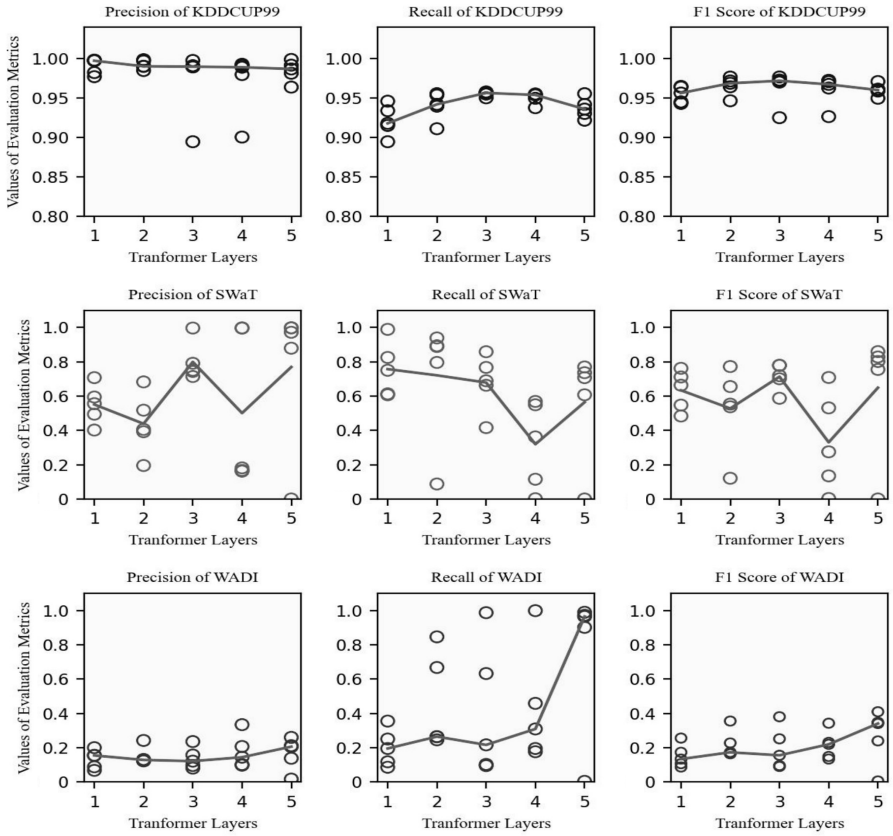

图 5　Transformer 层数对模型性能的影响

5　结束语

该文提出了一种用于时间序列的基于 Devformer 的深度异常检测方法。首先,通过基于时间窗口的方法预处理时序数据;其次,通过 Transformer 和基于 Z 得分的偏差网络建立异常得分模型。实验结果表明,相比于现有深度学习异常检测方法,该方法对于大规模时间序列的异常检测有更好的召回率、稳定性和抗干扰性。

由于 Devformer 的神经网络较为复杂,模型训练时间偏长,且若训练数据规模过小,模型效果偏差。Devformer 中神经网络的简化与提优是很重要的研究课题,可以提高模型训练效率以及对小规模数据的适应能力。

参考文献:

[1] PEI D,ZHANG S L,PEI C H. Intelligent operation and maintenance based on machine learning[J]. Communications of CCF,2017,13(12):67-73.

[2] DANG Y N,LIN Q W,HUANG P. AIOps:real-world challenges and research innovations[C]//Proc of 2019 IEEE/ACM 41st international conference on software engineering. Redmond:IEEE,2019:4-5.

[3] 郭丽丽,丁世飞. 深度学习研究进展[J]. 计算机科学,2015,42(5):28-33.

[4] PANG G,SHEN C,CAO L,et al. Deep learning for anomaly detection:a review[J]. ACM Computing Surveys,2021,54(2):38.

[5] 李贝贝,彭　力,戴菲菲. 结合马氏距离与自编码器的网络流量异常检测方法[J]. 计算机工程,2022,48(4):133-142.

[6] ZENATI H, ROMAIN M, FOO C S, et al. Adversarially learned anomaly detection[C]//Proc of 2018 IEEE international conference on data mining (ICDM). Piscataway: IEEE,2018:727-736.

[7] NANDURI A,SHERRY L. Anomaly detection in aircraft data using recurrent neural networks (RNN)[C]//2016 integrated communications navigation and surveillance (ICNS). Kyoto:IEEE,2016:5C2-1-5C2-8.

[8] 陈运文,吴　飞,吴庐山,等. 基于异常检测的时间序列研究[J]. 计算机技术与发展,2015,25(4):166-170.

[9] ZHANG C,SONG D,CHEN Y,et al. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data[C]//Proceedings of the AAAI conference on artificial intelligence. Hawaii: AAAI, 2019: 1409-1416.

[10] ZHANG Y,CHEN Y,WANG J,et al. Unsupervised deep anomaly detection for multi-sensor time-series signals[J]. IEEE Transactions on Knowledge and Data Engineering, 2021,Early Access:10.1109/TKDE.2021.3102110.

[11] AUDIBERT J,MICHIARDI P,GUYARD F,et al. USAD:

unsupervised anomaly detection on multivariate time series [C]//Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. Virtual Event: ACM, 2020: 3395-3404.
[12] LI D, CHEN D, JIN B, et al. MAD-GAN: multivariate anomaly detection for time series data with generative adversarial networks [C]//International conference on artificial neural networks. Munich: Springer, 2019: 703-716.
[13] ZHAO H, WANG Y, DUAN J, et al. Multivariate time-series anomaly detection via graph attention network [C]//2020 IEEE international conference on data mining (ICDM). Sorrento: IEEE, 2020: 841-850.
[14] DENG A, HOOI B. Graph neural network-based anomaly detection in multivariate time series[C]//Proceedings of the AAAI conference on artificial intelligence. Virtual: AAAI, 2021: 4027-4035.
[15] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Proc of the 31st international conference on neural information processing systems. Long Beach: ACM, 2017: 6000-6010.
[16] 王宇欣,方浩宇,张 伟,等. 注意力机制在情感分析中的应用研究[J]. 计算机技术与发展,2022,32(4):193-199.
[17] 袁非牛,章 琳,史劲亭,等. 自编码神经网络理论及应用综述[J]. 计算机学报,2019,42(1):203-230.
[18] VARUN C, ARINDAM B, VIPIN K. Anomaly detection: a survey[J]. ACM Computing Surveys, 2009, 41(3): 1-58.
[19] MOAYEDI H Z. Arima model for network traffic prediction and anomaly detection[C]//Proc of international symposium on information technology. Piscataway: IEEE, 2008: 1-6.
[20] LI K L, HUANG H K, TIAN S F, et al. Improving one-class SVM for anomaly detection[C]//Proc of 2003 international conference on machine learning and cybernetics. Piscataway: IEEE, 2003: 3077-3081.
[21] MUNIYANDI A P, RAJESWARI R, RAJARAM R. Network anomaly detection by cascading k-means clustering and C4.5 decision tree algorithm [J]. Procedia Engineering, 2012, 30: 174-182.
[22] HENRIQUES J, CALDEIRA F, CRUZ T J, et al. Combining k-means and XGBoost models for anomaly detection using log datasets[J]. Electronics, 2020, 9(7): 1164.
[23] DAI Z H, YANG Z L, YANG Y M, et al. Transformer-XL: attentive language models beyond a fixed-length context [C]//Proc of the 57th conference of the association for computational linguistics. Stroudsburg: ACL, 2019: 2978-2988.
[24] WU N, GREEN B, XUE B, et al. Deep transformer models for time series forecasting: the influenza prevalence case[J]. arXiv: 2001.08317, 2020.
[25] LIM B, SERCAN O A, LOEFF N, et al. Temporal fusion transformers for interpretable multi-horizon time series forecasting[J]. arXiv: 1912.09363, 2019.
[26] CHEN Z, CHEN D, ZHANG X, et al. Learning graph structures with transformer for multivariate time series anomaly detection in iot[J]. IEEE Internet of Things Journal, 2021, 9(12): 9179-9189.
[27] LI Y, PENG X, ZHANG J, et al. DCT-GAN: dilated convolutional transformer-based GAN for time series anomaly detection[J]. IEEE Transactions on Knowledge and Data Engineering, 2021, Early Access: 10.1109/TKDE.2021.3130234.
[28] WANG X, PI D, ZHANG X, et al. Variational transformer-based anomaly detection approach for multivariate time series [J]. Measurement, 2022, 191: 110791.
[29] XU J, WU H, WANG J, et al. Anomaly transformer: time series anomaly detection with association discrepancy[J]. arXiv: 2110.02642, 2021.
[30] PANG G, SHEN C, VAN DEN HENGEL A. Deep anomaly detection with deviation networks [C]//Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. Anchorage: ACM, 2019: 353-362.

(上接第 151 页)

[6] 王子豪,郭 玲. 基于改进 LM 算法的摄像机标定研究[J]. 工业控制计算机,2014,27(10):94-96.
[7] 汪首坤,赵金枝,姜 明,等. 基于圆形阵列标定板的张氏相机标定法[J]. 北京理工大学学报,2019,39(8):859-863.
[8] 张宏峰,倪受东,赵 亮. 基于麻雀搜索算法的摄像机标定优化方法[J]. 激光与光电子学进展,2021,58(22):2215004.
[9] 彭 梦,蔡自兴. 基于多约束误差函数的 2 维激光雷达和摄像机标定方法[J]. 机器人,2014,36(6):662-667.
[10] 王俊威,西 勤,冯其强,等. 基于抗差岭估计的相机标定方法[J]. 测绘工程,2018,27(9):64-68.
[11] ABDEL-AZIZ Y I, KARARA H M. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry [J]. Photogrammetric Engineering & Remote Sensing, 2015, 2(5): 103-107.
[12] ZHANG Z Y. A flexible new technique for camera calibration[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(11): 1330-1334.
[13] WANG Q, WANG Z, SHANG Z. Parameter calibration of a vision sensor with a geometric similarity constraint [J]. Measurement-Science and Technology, 2019, 31(3): 1-8.
[14] LENZ R K, TSAI R Y. Techniques for calibration of the scale factor and image center for high accuracy 3D machine vision metrology[J]. IEEE Trans on PAMI, 1998, 10(5): 713-720.
[15] WEI G, MA S. Implicit and explicit cam-era calibration: theory and experiments[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994, 16(5): 469-480.