

基于 PSO 和外部知识的时序数据异常检测^①

丁美荣, 王昭泓, 郑辛茹, 张迎春

(华南师范大学 软件学院, 佛山 528225)

通信作者: 王昭泓, E-mail: 2021024168@m.scnu.edu.cn



摘要: 在时间序列数据的异常检测中, 单一模型往往只提取与自身模型结构相关的时序特征, 从而容易忽略其他特征. 同时, 面对大规模的时序数据, 模型难以对时序数据的局部趋势进行建模. 为了解决这两个问题, 本文提出一种基于粒子群优化算法 (particle swarm optimization, PSO) 和外部知识的异常检测模型 PEAD. PEAD 模型以深度学习模型作为基模型, 引入快速傅里叶变换生成的外部知识来提高基模型对局部趋势的建模能力, 随后 PEAD 模型以 Stacking 集成学习的方式训练基模型, 再使用 PSO 算法对基模型的输出加权求和, 对加权求和后的重构数据进行异常检测, PSO 算法能够让模型的最终输出共同关注时序数据的全局特征和时间特征, 丰富模型提取的时序特征, 从而提高模型的异常检测能力. 通过对 6 个公开数据集进行测试, 研究结果表明 PEAD 模型在大部分数据集上表现良好.

关键词: 时间序列数据; 异常检测; 快速傅里叶变换; Stacking 集成学习; 粒子群优化算法

引用格式: 丁美荣, 王昭泓, 郑辛茹, 张迎春. 基于 PSO 和外部知识的时序数据异常检测. 计算机系统应用, 2024, 33(2): 83-93. <http://www.c-s-a.org.cn/1003-3254/9426.html>

Anomaly Detection of Time Series Data Based on PSO and External Knowledge

DING Mei-Rong, WANG Zhao-Hong, ZHENG Xin-Ru, ZHANG Ying-Chun

(School of Software Engineering, South China Normal University, Foshan 528225, China)

Abstract: In the anomaly detection of time series data, a single model often only extracts temporal features related to its model structure and thus tends to ignore other features. At the same time, facing large-scale temporal data, it is difficult for models to model local trends in temporal data. To address these two issues, this study proposes an anomaly detection model called PEAD based on particle swarm optimization (PSO) and external knowledge. The PEAD model uses a deep learning model as the base model and introduces external knowledge generated by the fast Fourier transform to improve the modeling ability of the base model for local trends. Subsequently, the PEAD model trains the base model through Stacking ensemble learning and then uses the PSO algorithm to sum the weighted output of the base model. The weighted sum of the reconstructed data is used for anomaly detection. The PSO algorithm enables the final output of the model to focus on the global and temporal features of the temporal data and enriches the temporal features extracted by the model, thereby improving its anomaly detection ability. By testing six publicly available datasets, the research results show that the PEAD model performs well on most of the datasets.

Key words: time series data; anomaly detection; fast Fourier transform (FFT); Stacking ensemble learning; particle swarm optimization (PSO) algorithm

① 基金项目: 广东省普通高校人工智能重点领域专项 (2019KZDZX1033)

收稿时间: 2023-07-24; 修改时间: 2023-10-09; 采用时间: 2023-10-25; csa 在线出版时间: 2023-12-27

CNKI 网络首发时间: 2023-12-28

时间序列数据,又称时序数据,是按照时间顺序进行排列的一组集合^[1]。当研究者们通过观察时序数据发现其中一个数据和其他数据相差过大时,则认为该时间点发生了异常,根据异常点数量的不同,可以将异常数据分为点异常和集合异常^[2],不同的异常情况带来的安全隐患不同,如果不进行及时处理,那么最后有可能带来重大的经济损失,因此,研究者们通过各种方法检测出时序数据的异常部分,这个过程被称为时序数据异常检测^[3]。时序数据异常检测作为数据分析领域的一个重要研究方向,它常被应用于各种工程应用,例如城市车辆轨道运行情况^[4]、航天器遥测数据^[5]和大气排污^[6]等,工程师们通过对时序数据发生异常的片段进行处理,从而提高系统的可靠性和准确性。

随着互联网和物联网的发展,越来越多的传感器数据被实时采集和存储,这些数据往往是多维庞杂的。所以如何从这些数据中提取有价值的信息,现已成为数据分析领域的重要研究问题之一。

早期研究人员使用统计分布的方法对时序数据进行异常检测,例如 K-means 模型^[7]将相似点聚集成簇,多数点形成的簇被视为正常,远离正常簇的点被视为异常,尽管 K-means 计算高效,但它难以处理多维数据。经典方法自回归求和滑动平均模型 (autoregressive integrated moving average model, ARIMA) 的变体也常被用于检测异常数据^[8],但很少用于多维时序数据的异常检测,并且异常检测的准确率仍需提高。Nakamura 等人^[9]提出的 MERLIN 模型通过改进时间序列不一致的方法解决了用户选择子序列长度敏感性问题,并在大量的时序数据中有效地找到所有可能的结果,但是子序列的排列容易受到其他维度的影响,以至于在多维时序数据上表现不佳。为了解决传统方法难以处理多维时序数据的问题,研究者们引入深度学习的方法。

相比传统方法,深度学习更适用于大量多维数据。其中, Hundman 等人^[10]提出长短期记忆无参阈值异常检测 (long short term memory and no parameter exception detection threshold, LSTM-NDT)。该方法利用 LSTM 得到重建误差,并通过无参阈值的方法判断该点是否为异常。LSTM-NDT 不需要人工干预,节省了大量人力成本,但 LSTM-NDT 单一的架构难以关注到时序数据其余特征。为了让模型关注到更多的时序特征,丰富时序预测结果,研究者们将多种深度学习模型进行融合,其中 Su 等人^[11]提出一种多模型融合的模

型 OmniAnomaly, OmniAnomaly 将门控循环单元 (gated recurrent unit, GRU) 和变分自动编码器 (variational auto-encoders, VAE) 相结合,它可以处理变量之间显式的时间关联,从而提高数据重建的鲁棒性,同时, OmniAnomaly 使用一种新的阈值自动检测方法: 阈值峰值 (peak of threshold, POT), POT 可以高效准确地检测异常。虽然 OmniAnomaly 利用 GRU 关注到时序数据之间的时间特性^[12],但往往会忽视时序数据的全局特性。为了捕获时序数据的全局特征, Tuli 等人^[13]提出一种基于 Transformer 的异常检测 (Transformer networks for anomaly detection, TranAD) 模型, TranAD 采用 Transformer 和 GAN 相结合的方式,将 Transformer 作为判别器和生成器进行对抗式训练以提高模型特征提取的稳定性,同时 TranAD 使用对抗训练一阶段的焦点得分作为外部知识,增强了模型对异常点的关注度,提高模型对时序数据局部趋势的建模能力,得益于 Transformer 模型的结构, TranAD 在众多数据集上表现良好,但因为 TranAD 是以 Transformer 为主的深度模型,所以 TranAD 不容易关注到时序数据的时间特征。

为了使模型能够同时关注到时序数据的多种特征,研究者们将目光投向 Stacking 集成学习, Stacking 集成学习^[14]是指结合所有基模型的一种训练方式,先训练不同的基模型,然后所有基模型的输出使用投票法或全连接等方式生成最终结果。使用 Stacking 集成学习可以让不同的基模型学习到不同的时序特征,降低过拟合风险,从而提高模型的整体性能。陈仲磊等人^[15]提出一种基于集成学习的异常检测框架,使用预训练生成式转换器 (generative pre-trained Transformer, GPT)、GRU、随机森林作为集成学习的基模型,基模型 GRU 作为预测的主要网络, GPT 利用注意力机制提高模型预测准确率,两个基模型共同关注时序数据的全局和时间特征,提高模型异常检测能力,随机森林对整体输出进行宏观调控提升模型稳定性。但是过于复杂的集成模型难以训练大规模数据,并且计算开销大。

综合以上研究存在的问题,本文提出一种基于 PSO 和外部知识的时序数据异常检测模型 (anomaly detection of time series based on PSO and external knowledge, PEAD),其中为了提高基模型对时序数据局部趋势的建模能力, PEAD 将快速傅里叶变换 (fast Fourier transform, FFT) 生成的外部知识引入基模型,另外,为了使得 PEAD 模型可以同时关注时序数据时

间特征和全局特征, PEAD 使用 PSO^[16]对基模型进行 Stacking 集成学习, 通过减少基模型数量和使用传统方法集成, 保证模型结构简洁高效的同时能够有效处理大规模数据。

1 理论基础

1.1 粒子群优化算法

Kennedy 等人^[17]受到鸟类觅食规律的启发, 经过多年研究最终形成粒子群优化算法, 又称粒子群算法, 粒子群算法的本质在于集体的信息共享机制。

如图 1 所示, 假设森林中的鸟类进行随机移动, 目的是为了搜索整个森林中最大食物, 每个鸟类通过初始方向进行随机搜索, 在第 1 次搜索结束后, 鸟类可能会接近最大食物, 也可能会接近少量食物, 又或者偏离食物, 搜索结束后鸟类会记录当前搜索到的最大食物和位置, 然后通过共享机制将食物和位置共享给整个鸟类群, 这样鸟类群就知道哪个位置的食物最多, 鸟类在随后的每一次搜索通过自身记录的最大食物位置和全体最大食物位置进行移动, 从而使整个鸟类群向最大食物的位置聚拢。

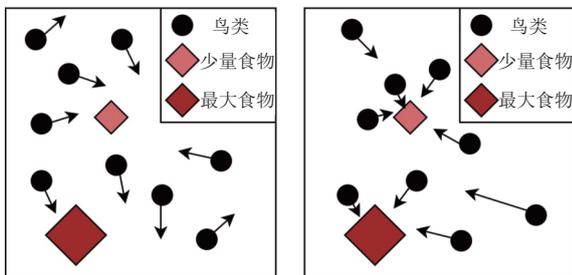


图 1 鸟类捕食过程

将上述鸟类捕食过程对应到粒子群算法中, 鸟类表示粒子, 食物表示适应值, 鸟类位置表示一个局部解, 鸟类当前搜索到的最大食物表示局部最大适应值, 最大食物表示全局最大适应值, 鸟类当前搜索到的最大食物位置表示局部最优解, 最大食物位置表示全局最优解. 本研究的目的是通过粒子群算法, 找到两个模型输出的最优权重, 也就是最大食物位置, 通过加权求和的方式给不同的模型分配权重, 让模型可以同时关注到时序数据的不同特征。

1.2 快速傅里叶变换

快速傅里叶变换是离散傅里叶变换的一种快速计算方法, 本质上是使用分治策略, 将数据规模为 N 的数

据分成多个区间进行离散傅里叶变换, 变换过程中利用数据的周期性和对称性删减掉不必要的重复计算, 从而加快计算速度, 使得离散傅里叶变换的时间复杂度从 $O(N^2)$ 下降到 $O(N \log N)$ ^[18]。

本研究使用 FFT 算法, 将时序数据从时域映射到频域, 然后保留高频的数据, 最后使用逆 FFT 生成高频时序数据, 高频时序数据将作为外部知识和原始数据结合, 最后生成带有高频信息的输入数据, 相比原始数据, 带有高频信息的输入数据更加关注时序数据中变化明显的区域, 从而增强基模型对局部趋势的建模能力。

2 基于 PSO 和外部知识的异常检测模型

本文基于 PSO 和外部知识的异常检测模型算法流程如算法 1 所示。

算法 1. 基于 PSO 和外部知识的异常检测模型

- 1) 输入: 训练集 $T = \{x_1, x_2, \dots, x_N\}$;
- 2) 数据预处理: 对 T 使用归一化和滑动窗口得到 W_N ;
- 3) FFT 生成外部知识: 对 W_N 使用 FFT 模块生成新的外部知识 W'_N ;
- 4) 基模型 Transformer 生成重构数据 O_1 : W'_N 和 W_N 矩阵按列合并生成 I , I 输入基模型 Transformer 的编码层 Encoder, T 输入解码层 Decoder, 经过 $\text{Decoder}(\text{Encoder}(I), T)$ 生成重构数据 O_1 ;
- 5) 基模型 LSTM 生成重构数据 O_2 : T 输入基模型 LSTM 生成重构数据 O_2 ;
- 6) for $i=1$ to n : 步骤 4) 和步骤 5) 经过迭代训练后分别得到 O'_1 和 O'_2 , 损失函数使用均方误差;
- 7) PSO 生成最终重构数据 O : 利用 PSO 算法求出 O'_1 和 O'_2 的最优权重 P_w , 适应函数使用均方误差, 然后使用 P_w 对 O'_1 和 O'_2 进行加权求和生成最终重构数据 O ;
- 8) 输出: 对 O 和 T 使用均方误差得到损失 L , 损失 L 输入 POT 生成结果 Y , $Y=1$ (if $L > \text{POT}(L)$).

本文基于 PSO 和外部知识的异常检测模型 PEAD 结构图如图 2 所示, 整体结构分为 6 个部分, 首先对原始时序数据 T 进行预处理操作, 经过归一化和滑动窗口后得到处理后的时序数据 W_N . 其次, 将预处理时序数据输入 FFT 模块生成关注高频局部趋势的外部知识 W'_N . 接下来是集成模型训练部分, 第 1 个基模型是 Transformer, W'_N 和 W_N 合并生成待输入数据 I , I 输入 Transformer 的编码层, 因为 Transformer 具有关注全局特征的能力, 所以需要额外的外部知识来补足它局部信息模糊的缺陷, 这里引入包含高频数据的外部知识, 可以让 Transformer 更好地弥补局部信息的细节特征, 再将原始数据 T 输入解码层, Transformer 模型生成重构数据 O_1 ; 另一个基模型是 LSTM, 原始数据 T 输入

LSTM, 每个 LSTM 子模块预测每个时间点的数据 y , 所有 y 形成集合作为重构数据 O_2 . 然后对基模型 Transformer 和 LSTM 分别进行迭代训练, 得到训练后的重构数据 O'_1 和 O'_2 , 训练过程中使用的损失函数是均方误差. 之后将 O'_1 和 O'_2 输入到 PSO 模块, PSO 会根据适应函数生成 O'_1 和 O'_2 的最优权重, 接着对 O'_1 和 O'_2 使用加权求和生成最终的重构数据 O , 因为 LSTM 生成的重构数据带有时序数据的时间特征, Transformer 生成的重构数据带有时序数据的全局特征, 所以经过加权求和后的 O 共同关注了时序数据的时间和全局特征, 而且使用 PSO 模块不需要集成学习交叉验证等方法, 整体结构简洁高效. 最后对 O 和 T 使用均方误差生成 L , L 输入 POT 模块, POT 模块生成阈值自动判断时序数据点是否异常, 大于阈值的时间点设为异常点 1, 正常点为 0, POT 模块最终生成只包含 0 或 1 的集合 Y .

第 2 节剩下部分将会详细介绍模型的具体实现.

2.1 数据预处理

在进行异常检测之前, 首先对时序数据进行预处理操作. 以多维时序数据为例, 假设时间长度为 N , 时序数据可以表示如下:

$$T = \{x_1, x_2, \dots, x_N\} \quad (1)$$

其中, x_n 表示第 n 个时间点的时序数据, $x_n \in R^m, n \in [1, N]$. 其中 m 代表维度至少为 1 的正整数, 如果 $m=1$, 则说明输入的数据是单维时序数据.

接下来对数据进行归一化处理, 并且设置滑动窗口, 归一化如下:

$$x_n = \frac{x_n - \min(T)}{\max(T) - \min(T) + \varepsilon} \quad (2)$$

其中, $\min(T)$ 、 $\max(T)$ 分别表示时序数据 T 的最小值和最大值, ε 是一个常量, 用来防止分母为 0 的情况, 经过归一化后, $x_n \in [0, 1)$.

滑动窗口经常被用于处理时序数据^[13], 具体地, 使用滑动窗口对归一化时序数据进行切片, 切片大小取决于窗口大小 q .

$$W_N = \{x_{N-q+1}, \dots, x_N\} \quad (3)$$

2.2 FFT 生成外部知识

滑动窗口 W_N 直接输入 FFT 模块进行傅里叶变换:

$$f_j = \sum_{n=n-q+1}^N x_n e^{-i\omega n} \quad (4)$$

FFT 内部生成频率 $f_j, j \in [0, n/2 + 1]$, 其中 n 是非负整数. 频率 f_j 在进行逆 FFT 之前, 需要去除低频数据, 保留前 k 个最大频率点, 其他频率点设置为 0, 得到 f'_j .

$$f'_j = \begin{cases} f_j, & j \in \max(f_j, k) \\ 0, & j \notin \max(f_j, k) \end{cases} \quad (5)$$

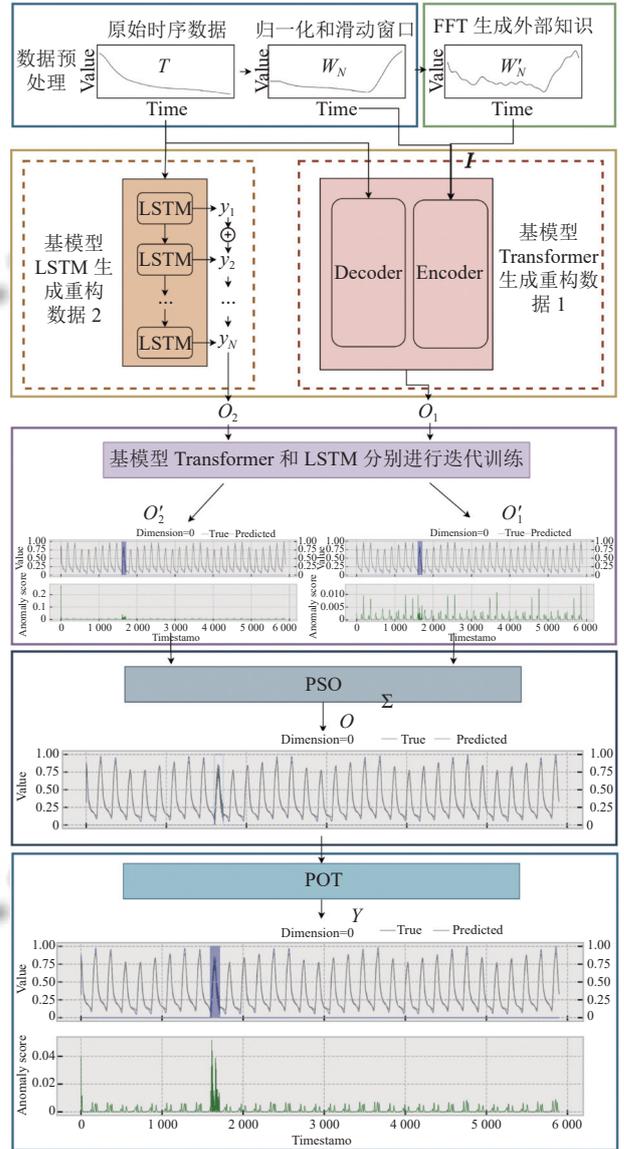


图 2 PEAD 模型结构图

最后, f'_j 通过逆 FFT, 得到新的时序数据 W'_N .

$$W'_N = \sum_{j=0}^{n/2+1} f'_j e^{i\omega j} \quad (6)$$

新的时序数据 W'_N 由高频数据生成, 它保留了原始数据的局部趋势. 如图 3 所示, 时序数据在经过 FFT 模块之后生成高频曲线 (虚线), 高频曲线相比原始曲线

(实线), 在时间跨度上更趋向于局部细节的描绘, 这一点有利于构建时序数据的局部趋势。

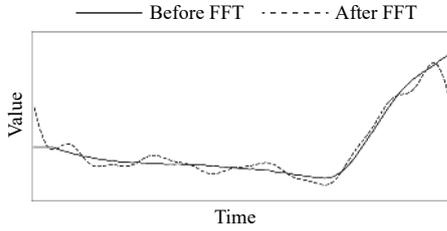


图3 时序数据 FFT 前后对比

2.3 集成模型训练部分

2.3.1 基模型 Transformer 生成重构数据 1

实验使用的基模型之一是 Transformer, 它常被用于各种自然语言处理任务^[19], 由于时序数据属于文本数据的一种, 因此可以使用 Transformer 处理时序数据。如图 4 所示, 基模型 Transformer 分为两个部分, 一个是编码层 Encoder, 另一个是解码层 Decoder。外部知识 W'_N 和预处理后的数据 W_N 结合得到 I , 将其输入到 Encoder。

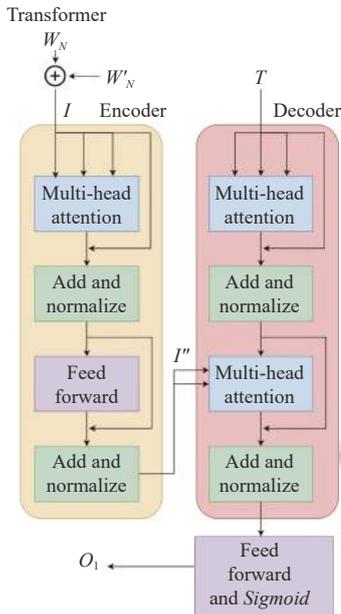


图4 Transformer 模型结构图

$$I = W_N \oplus W'_N \quad (7)$$

$$I' = \text{Norm}(I + \text{MultiHeadAtt}(I, I, I)) \quad (8)$$

$$I'' = \text{Norm}(I' + \text{FeedForward}(I')) \quad (9)$$

其中, “ \oplus ”表示矩阵按列拼接, MultiHeadAtt 表示多头注意力机制, 用来捕获时序数据的全局特征^[19], 矩阵

I 和 MultiHeadAtt 的输出进行残差连接, Norm 是指层归一化, 用来稳定模型的输出^[19]. FeedForward 是前馈神经网络, 本质上是一层全连接神经网络, 用来控制模型输出的维度. I' 经过 FeedForward 再经过残差连接和 Norm 后, 最终 I'' 作为 Encoder 的输出。

对于 Decoder, 直接将原始数据 T 输入 Decoder。

$$T' = \text{Norm}(T + \text{MultiHeadAtt}(T, T, T)) \quad (10)$$

$$T'' = \text{Norm}(T' + \text{MultiHeadAtt}(T', I'', I'')) \quad (11)$$

其中, Encoder 的输出 I'' 作为 Decoder 的 Key 和 Value, 可以让 T' 充分地利用 Encoder 的上下文信息^[19]. 最后将 T'' 输入 FeedForward 和激活函数 Sigmoid , 得到重构数据 O_1 。

$$O_1 = \text{Sigmoid}(\text{FeedForward}(T'')) \quad (12)$$

激活函数 Sigmoid 的作用主要是生成 $[0, 1]$ 范围内的输出, 并使结果标准化。

对基模型 Transformer 使用梯度下降算法优化模型权重, 误差函数使用均方误差, Transformer 经过迭代训练后得到重构数据 O_1 , 损失函数公式如下:

$$L_1 = \|O_1 - T\|_2 \quad (13)$$

2.3.2 基模型 LSTM 生成重构数据 2

另一个基模型使用 LSTM, 原始数据 T 作为输入. LSTM 和 Transformer 不同, 它主要是对 x_n 进行下一步 y_n 的预测, y 的集合作为最终的重构数据 O_2 。

LSTM 有 3 个门控机制, 利用不同的权重来过滤过去的信息, 首先是用来更新输入门的模块:

$$z_n = g(W_z x_n + R_z y_{n-1} + b_z) \quad (14)$$

其中, g 是激活函数, 使用 \tanh 进行激活, 它可以将数值规定到 $[-1, 1]$ 的范围内; W_z 、 R_z 是当前时间点 x_n 和上一个 LSTM 块输出 y_{n-1} 的权重矩阵, b_z 是偏置向量。

输入门: 当前时间点输入 x_n 和上一时间点输出 y_{n-1} 按位乘运算实现信息提取的作用^[12]。

$$i_n = \sigma(W_i x_n + R_i y_{n-1} + P_i \otimes c_{n-1} + b_i) \quad (15)$$

其中, σ 是激活函数 Sigmoid , W_i, R_i, P_i 分别是 x_n, y_{n-1}, c_{n-1} 的权重矩阵, \otimes 表示逐点相乘, b_i 是偏置向量。

c_{n-1} 表示上一个 LSTM 块的单元状态 (隐藏态), 它记录了前 $n-1$ 个时间点的所有信息^[12]。

遗忘门: 时序数据并非每个点都息息相关, 这时就需要遗忘门有选择地遗忘一部分数据, 从而达到更鲁

棒的效果^[12]。

$$f_n = \sigma(W_f x_n + R_f y_{n-1} + P_f \otimes c_{n-1} + b_f) \quad (16)$$

其中, σ 是激活函数 *Sigmoid*, W_f, R_f, P_f 分别是 x_n, y_{n-1}, c_{n-1} 的权重矩阵, b_f 是偏置向量。

隐藏状态: 隐藏状态 c 记录之前所有时间点的所有信息^[12], 它保存了每个时间点的时间特征。

$$c_n = z_n \otimes i_n + c_{n-1} \otimes f_n \quad (17)$$

利用更新模块 z_n 对输入门 i_n 进行有效信息的提取, 利用遗忘门 f_n 有选择地删除前一个隐藏状态 c_{n-1} , 最后得到当前的隐藏状态 c_n 。

输出门: LSTM 块利用当前时间点 x_n 、上一个时间点输出 y_{n-1} 和当前隐藏状态 c_n 综合计算当前的输出, 隐藏状态 c_n 包含了输入门和输出门, 所以输出门实际上是前一时刻和当前时刻信息的汇总。

$$o_n = \sigma(W_o x_n + R_o y_{n-1} + P_o \otimes c_n + b_o) \quad (18)$$

其中, σ 是激活函数 *Sigmoid*, W_o, R_o, P_o 分别是 x_n, y_{n-1}, c_n 的权重矩阵, b_o 是偏置向量。

LSTM 输出: 对 c_n 进行非线性激活, 然后和输出门逐点相乘, 相乘后的结果作为当前 LSTM 块输出。

$$y_n = g(c_n) \otimes o_n \quad (19)$$

一个时间点使用一个 LSTM 块预测 y_n , 最后得到的所有预测值 y 生成的预测集合 Y , 图 5 “+” 表示集合操作, Y 经过前馈神经网络得到最终的重构数据 O_2 。

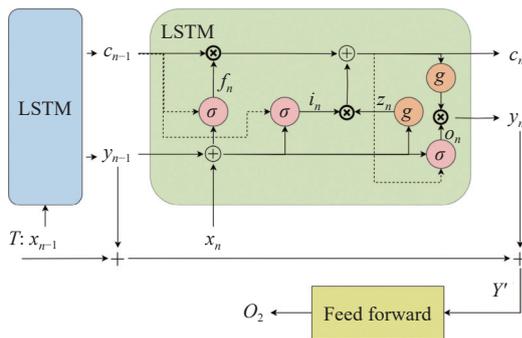


图 5 LSTM 模型结构图

$$O_2 = \text{FeedForward}(Y) \quad (20)$$

对基模型 LSTM 使用梯度下降算法优化模型权重, 误差函数使用均方误差, LSTM 经过迭代训练后得到重构数据 O_2' , 损失函数公式如下:

$$L_2 = \|O_2 - T\|_2 \quad (21)$$

2.4 PSO 生成最终重构数据

PSO 是一种启发式寻优算法, 它通过粒子群不断地迭代优化函数, 最终寻找到函数取得最小或最大值时的最优位置^[16]。

假设在 2 维空间有 N 个粒子, 每个粒子代表一个解。

$$W_i = \{w_{i1}, w_{i2}\} \quad (22)$$

其中, W_i 是第 i 个粒子的位置, 表示通过适应函数求得的一个解, 因为基模型有两个, 所以 W_i 维度为 2, W_i 的初始值随机设置。

$$f = \|(w_{i1} O_1' + w_{i2} O_2') - T\|_2 \quad (23)$$

其中, f 表示适应函数, w_{i1} 和 w_{i2} 分别作为输出 O_1' 和 O_2' 的权重进行加权求和, 加权求和的结果和原始数据 T 进行均方误差得到 f , f 越小说明预测输出与原始数据越接近。

$$V_i = \{v_{i1}, v_{i2}\} \quad (24)$$

其中, V_i 是第 i 个粒子的速度, 表示每个粒子移动的方向和大小, 粒子群算法通过对 V_i 的更新使得 W_i 向最优位置移动, V_i 的更新公式如下:

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_{il}^k - W_i^k) + c_2 r_2 (P_w^k - W_i^k) \quad (25)$$

其中, k 表示当前时刻, $k+1$ 表示下一时刻, ω 是惯性权重, c_1 和 c_2 是可以自定义的学习因子; r_1 和 r_2 是区间 $[0, 1]$ 内的随机数, 提供随机的方向。 P_{il} 表示第 i 个粒子的局部最优位置, 在每一次迭代中, 通过比较当前粒子的适应值和之前记录的局部最优适应值来更新 P_{il} 。 P_w 表示全局最优位置, 在每一次迭代中, 通过比较所有粒子的适应值和之前记录的全局最优适应值来更新 P_w 。

速度更新完成后再对粒子的位置进行更新, 更新公式如下:

$$W_i^{k+1} = W_i^k + V_i^{k+1} \quad (26)$$

更新后的 W_i^{k+1} 作为下一时刻的粒子位置继续迭代, 直到结束, PSO 算法整体流程如算法 2。

算法 2. PSO 算法

- 1) 初始化粒子群;
- 2) 计算每个粒子的适应值 f ;
- 3) 对于每个粒子, 将当前适应值和局部最优适应值进行比较, 如果当前适应值小于局部最优适应值, 那么更新局部最优适应值, 并且更新局部最优位置 P_{il} ;
- 4) 对于每个粒子, 将当前适应值和全局最优适应值进行比较, 如果当前适应值小于全局最优适应值, 那么更新全局最优适应值, 并且更新全局最优位置 P_w ;

- 5) 迭代更新粒子速度 V_i , 位置 W_i ;
6) 根据终止条件判断迭代次数是否达到最大, 或者粒子全局最优适应值是否小于给定阈值, 若是, 结束算法并输出全局最优适应值和对应最优位置 P_w ; 否则返回步骤 2).

PSO 算法得到的 P_w 作为 O'_1 和 O'_2 的权重系数, 经过加权求和后得到 O 作为 PSO 模块的输出.

$$P_w = \{p_1, p_2\} \quad (27)$$

$$O = p_1 O'_1 + p_2 O'_2 \quad (28)$$

经过 PSO 模块得到最优权重 P_w , 再通过加权求和的方式, 使得最终重构数据 O 同时包含了 O'_1 和 O'_2 的特征信息, 由于基模型 Transformer 和 LSTM 的结构特性, PSO 的输出 O 可以同时关注时序数据的全局特征和时间特征.

2.5 POT 阈值判断异常点

实验使用 POT 方法自动选择异常的阈值^[20], 对重构数据 O 使用均方误差得到损失 L , 损失 L 输入 POT 模块自动得到对应阈值, 一旦误差 L 大于阈值, 那么该时间点的测试数据被判定为异常, 对应 $y_i = 1$, 所有 y 构成的集合 Y 作为 PEAD 模型的输出. 本质上 POT 利用的是样本数据的极值, 极值采用超阈值的方法获得, 在先前的异常检测研究中, 许多实验使用 POT 来标记异常^[10,13,21].

$$L = \|O - T\|_2 \quad (29)$$

$$y_i = 1 \text{ (if } L \geq POT(L)) \quad (30)$$

$$Y = \{y_1, y_2, \dots, y_n\} \quad (31)$$

3 实验与分析

实验将模型 PEAD 与目前先进的多维时序数据异常检测模型作对比, 包括 TranAD^[13]、LSTM-NDT^[10]、OmniAnomaly^[11]和 MERLIN^[9]. 实验使用 PyTorch 1.7.1 版本.

所有模型统一使用 AdamW 优化器进行梯度下降, 初始学习率为 0.01, 学习率更新因子为 0.9 (每次下降 10%), 滑动窗口大小都设置为 10, 模型丢弃率设置为 0.1, 损失函数使用均方误差.

PEAD 的 Transformer 编码层和解码层的层数都为 1, 前馈神经网络输入维度为 2, 隐藏层维度为 64. LSTM 层数为 2, 隐藏层维度为 64. PSO 粒子群个数为 50, 迭代步数 3 000, 阈值 $1E-7$, 粒子位置边界 $[0, 1]$,

粒子速度边界 $[-0.3, 0.3]$, 惯性权重、学习因子都设置为 2. 对于 POT 参数, PEAD 和 TranAD 保持一致^[13], 系数为 $10E-4$, MSL 低分位数为 0.001, 多头注意力机制的头数和时序数据的维度保持一致.

TranAD 使用两个 Transformer 模型, 其中一个作为编码层, 层数为 1, 另外一个作为解码层, 层数为 1, 解码层的前馈神经网络单元为 2, 编码层中的隐藏单元为 64. TranAD 使用 GAN 方式将编码层作为生成器, 解码层作为鉴别器, 第 1 阶段生成器生成的预测数据作为焦点得分和原始数据结合, 然后将带有焦点得分的数据输入到第 2 阶段生成预测数据, 预测数据再通过均方误差进行梯度下降, 迭代优化.

LSTM-NDT 采用 2 层 LSTM 进行时序数据预测, 使用 linear 作为激活函数, 对预测数据构成的误差进行加权平均的平滑处理, 根据平滑后的数据计算阈值, 大于阈值的标记为异常, 和普通 LSTM 相比增加了非参数动态误差阈值.

OmniAnomaly 采用 2 层 GRU 架构, 隐藏层单元为 32, 激活函数采用 ReLU 和 Sigmoid, 第 1 层 GRU 对时序数据进行编码后, 再对其使用 VAE, VAE 将编码后的数据映射到潜在空间 z , 然后再用第 2 层 GRU 生成预测数据, 损失函数除了均方误差外还有 KL 散度.

MERLIN 使用统计分布的办法进行异常检测, 算法整体分为两个阶段, 第 1 阶段目的是得到异常集合 C , 对时间序列使用 $\min L=60$ 进行数据滑动得到不同的子序列, 再使用均方误差得到每两个子序列之间的误差, 误差大于阈值 r 的时间点加入异常集合 C , $r = 2 \times \sqrt{\min L}$; 第 2 阶段为了去除异常集合 C 的假阳性结果, 使用集合 C 的均值 M 、标准差 S 计算阈值 r , $r = M - 2 \times S$, 利用 r 得到排除假阳性结果后的异常集合 D , 然后将异常集合 D 中的时间点标记到时序数据, 完成异常检测.

3.1 数据集

实验中使用的 6 个数据集如表 1 所示.

表 1 数据集

数据集	训练集	测试集	维度	异常率 (%)
UCR	1 600	5 900	1	1.88
MSL	58 317	73 729	55	10.72
MBA	100 000	100 000	2	0.14
MSDS	146 430	146 430	10	5.37
SWaT	496 800	449 919	51	11.98
SMD	708 405	708 420	38	4.16

表 1 是实验使用公开数据集的详细信息^[13], 包括训练集、测试集、维度和异常率, 异常率指的是测试集中数据点异常所占的百分比. 其中除了 UCR 是单维时序数据外其他数据集都是多维数据.

HexagonML 数据集 (UCR) 是 KDD 2021 杯使用的数据集^[22], 实验中仅使用 InternalBleeding 和 EGG 数据集. 火星科学实验室数据集 (Mars science laboratory, MSL) 来自火星车本身的传感器和制动器^[10]. MIT-BIH 室上性心律失常数据库 (MIT-BIH supraventricular arrhythmia database, MBA) 是 4 名患者的心电图集合, 包含了两种不同的心律失常的实例^[23,24]. 多源分布式系统数据集 (multi-source distributed system, MSDS) 包含了分布式跟踪日志、应用程序日志和度量等多个维度^[25]. 安全水处理数据集 (secure water treatment, SWaT) 来自真实水处理厂, 包括传感器值和制动器操作^[26]. 服务器数据集 (server machine dataset, SMD) 包含了 5 周数据, 数据来自 28 台机器资源调度情况^[11].

3.2 评估指标

异常检测实验使用 4 个指标进行评估, 包括精确率 (precision, P)、召回率 (recall, R)、接受操作特征曲线下面积 (area under the receiver operating characteristic curve, ROC/ AUC) 和 $F1$ 值, 实验使用这 4 个指标评估所有模型性能.

$$P = \frac{TP}{TP + FP} \quad (32)$$

其中, TP 表示正类预测为正类的个数, FP 表示负类预测为正类的个数, P 表示预测正确的个数在所有正类预测个数的比例.

$$R = \frac{TP}{TP + FN} \quad (33)$$

其中, FN 表示正类预测为负类的个数, R 表示正类被预测为正类的个数在所有标注数据上的占比.

$$AUC = \frac{\sum_{i \in \text{positive}} \text{rank}_i - \frac{M \times (M + 1)}{2}}{M \times N} \quad (34)$$

其中, rank_i 表示按照预估概率排序后正样本 i 的排序编号, M 是正样本个数, N 是负样本个数, AUC 表示 ROC 曲线下面积.

$$F1 = \frac{2 \times P \times R}{P + R} \quad (35)$$

其中, $F1$ 值表示精确率和召回率的调和平均数. P 、 R 、

AUC 和 $F1$ 值越大表示模型性能越好.

3.3 对比实验

选取 MERLIN、LSTM-NDT、OmniAnomaly 和 TranAD 作为比较基准, 使用测试集对训练好的模型进行测试, 测试主要以 AUC 和 $F1$ 的结果作为异常检测性能好坏的判断依据, 其中 MERLIN、LSTM-NDT 和 OmniAnomaly 的实验结果来自 Tuli 等人^[13].

表 2 提供了对比模型在所有数据集上的得分, 粗体表示在所有对比模型中获得最高得分. PEAD 模型在 P 、 R 、 AUC 和 $F1$ 上的平均得分分别为 0.9371、0.9275、0.9602、0.9259. 并且在 MAB、SWaT、SMD 和 MSL 这 4 个数据集上取得最高 $F1$ 得分. 由于时序数据具有多种特性, 包括数据之间的时间关联、空间关联, 因此, 通过 PSO 分配权重给基模型 Transformer 和 LSTM 的预测数据, 使得预测数据各自占有最佳比例的权重, 然后通过加权求和生成最终预测数据, 最终预测数据带有模型的结构特点, Transformer 由于多头注意力机制所以更注重全局特性, LSTM 因为前一刻的输出作为下一刻的输入所以更关注时间特性, 相比单一的深度模型, 具备丰富时序特性的 PEAD 能够在大部分数据集上取得更好的得分.

PEAD 模型在所有数据集的 $F1$ 和 AUC 得分, 全部超过基于统计分布的传统算法 MERLIN. 另外, 在多维数据集上, 所有深度模型 $F1$ 和 AUC 得分普遍优于 MERLIN. MERLIN 属于统计分布的方法, 通过发现子序列不一致的情况判断该时序区间是否异常, 这种方法更适用于序列长且周期规律的时序数据, 面对分布不均的多维时序数据, MERLIN 难以通过周期性进行异常检测, 而深度模型是通过预测的形式进行异常检测, 并不完全依赖周期规律, 所以深度模型在时序数据异常检测方面比传统方法更有优势.

PEAD 模型与 LSTM-NDT 模型相比, 在单维数据集 UCR 上, 对比指标 $F1$ 得分提高了 36.8%, AUC 得分提高了 1.96%, 在其他多维数据集上, $F1$ 得分平均提高了 13.8%, AUC 得分平均提高了 11.9%. PEAD 模型与 OminAnomaly 模型相比, 在单维数据集 UCR 上, 对比指标 $F1$ 得分下降了 3.55%, 在其他多维数据集上, $F1$ 得分平均提高了 5.35%, AUC 得分平均提高了 1.95%. PEAD 模型与 TranAD 模型相比, 在单维数据集 UCR 上, 对比指标 $F1$ 得分下降了 9.48%, AUC 得分下降了 0.21%, 在其他多维数据集上, $F1$ 得分平均提高了 1.11%,

AUC 得分平均提高了 0.833%。受限于基模型 Transformer 和 LSTM 的性能, PEAD 模型的集成效果在 UCR 和 MSD 数据集上表现不如 TranAD 模型, 但在其余数据集上的 *AUC* 和 *F1* 得分都超过了 TranAD 模型。LSTM-NDT 和 OminAnomaly 都是以 RNN 为主的深度模型, 模型的预测数据只与时序数据的时间特征相关联, TranAD 是以 Transformer 为主的深度模型, 模型的预

测数据只与时序数据的全局特征相关联, 这两类模型都只关注到单方面的时序特征, 预测数据的时序特征不够丰富, 包含信息量相对较少, 而 PEAD 通过 PSO 对预测数据分配权重, 使得最终预测数据同时关注到时序数据的时间特征 (来自基模型 LSTM) 和全局特征 (来自基模型 Transformer), 预测数据包含的信息量大于其他深度模型, 因此在大部分数据集上表现良好。

表 2 PEAD 模型和对比模型在所有数据集上的得分比较 (最佳 *F1* 得分和 *AUC* 得分用粗体表示)

Method	UCR				MBA				MSL			
	<i>P</i>	<i>R</i>	<i>AUC</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>AUC</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>AUC</i>	<i>F1</i>
MERLIN	0.7542	0.8018	0.8984	0.7773	0.9846	0.4913	0.7828	0.6555	0.2613	0.4645	0.6281	0.3345
LSTM-NDT	0.5231	0.8294	0.9781	0.6416	0.9207	0.9718	0.9780	0.9456	0.6288	1.0000	0.9532	0.7721
OminAnomaly	0.8346	0.9999	0.9981	0.9098	0.8561	1.0000	0.9570	0.9225	0.7848	0.9924	0.9782	0.8765
TranAD	0.9407	1.0000	0.9994	0.9694	0.9569	1.0000	0.9885	0.9780	0.9038	0.9999	0.9916	0.9494
Ours (PEAD)	0.7817	0.9999	0.9973	0.8775	0.9698	1.0000	0.9920	0.9847	0.9064	0.9999	0.9918	0.9509
Method	SWaT				SMD				MSDS			
	<i>P</i>	<i>R</i>	<i>AUC</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>AUC</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>AUC</i>	<i>F1</i>
MERLIN	0.6560	0.2547	0.6175	0.3669	0.2871	0.5804	0.7158	0.3842	0.7254	0.3110	0.5022	0.4353
LSTM-NDT	0.7778	0.5109	0.7140	0.6167	0.9736	0.8440	0.9671	0.9042	0.9999	0.8012	0.8013	0.8896
OminAnomaly	0.9782	0.6957	0.8467	0.8131	0.8881	0.9985	0.9946	0.9401	1.0000	0.7964	0.8982	0.8867
TranAD	0.9760	0.6997	0.8491	0.8151	0.9262	0.9974	0.9974	0.9605	0.9999	0.8626	0.9013	0.9262
Ours (PEAD)	0.9646	0.7624	0.8791	0.8517	0.9999	0.9999	1.0000	0.9999	0.9999	0.8026	0.9012	0.8904

另外, 为了探究通过 FFT 生成的外部知识在 PEAD 模型中起的作用, 如图 6 所示, 上面的曲线是 PEAD 模型在 MSL 数据集第 0 维度上的预测曲线 (虚线) 和真实曲线 (实线), 下面的曲线是 OminAnomaly 模型在 MSL 数据集第 0 维度上的预测曲线和真实曲线。从图 6 可以看出, PEAD 模型的预测曲线在真实曲线突变点 (值突然下降的点) 可以较好地重构数据, 而没有外部知识引入的 OminAnomaly 模型, 它的预测曲线平缓, 对真实曲线突变点不敏感, 这是因为 PEAD 的基模型 Transformer 将高频预测数据和原始数据结合生成输入数据, 输入数据更加关注曲线陡峭的时间点, 从而有效地提高模型对时序数据局部趋势的建模能力, 最终提高模型的异常检测准确率。

3.4 消融实验

为了进一步验证模型的效果, 表 3 显示了消融实验的结果。包括 PEAD 模型、Transformer w/o FFT 模型、LSTM 模型和 Transformer FFT 模型在所有数据集上 *P*、*R*、*AUC* 和 *F1* 得分。Transformer w/o FFT 模型作为单一模型进行测试, 不引入 FFT 生成的外部知识, LSTM 模型和 Transformer FFT 模型也都作为单一模型进行测试, 但 Transformer FFT 模型引入 FFT 生成的外部知识。

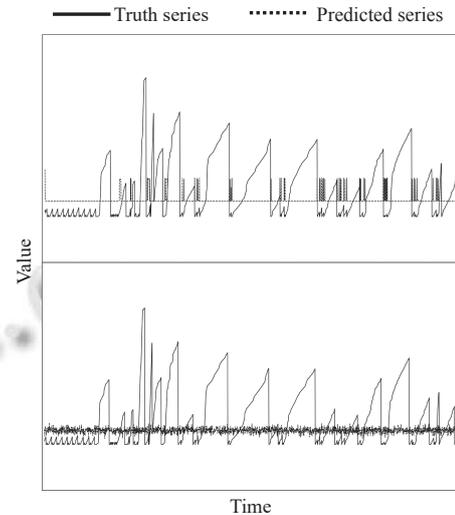


图 6 PEAD (上) 和 OminAnomaly (下) 在 MSL 数据集第 0 维度上的预测曲线和真实曲线

从所有模型在 UCR、MBA、MSL、SWaT、SMD 和 MSDS 数据集上的实验结果可以看出, PEAD 对比 Transformer w/o FFT 模型、LSTM 模型和 Transformer FFT 模型, *F1* 得分平均提高了 3%, *AUC* 平均得分提高了 1%。虽然在 MSDS 数据集上 PEAD 模型的 *F1* 得分下降了 0.112%, 但 PEAD 模型的 *F1* 得分十分接近 Transform FFT 模型。 *F1* 得分下降是因为 MSDS 数据集来自分布式系统, 包括指标、日志和跟踪数据,

这些维度的数据更加关注全局特征,但加入了低 $F1$ 得分的 LSTM 影响了 PEAD 的集成效果. 整体来看, PEAD 模型集成效果大部分优于单个基模型, 这是因为生成的预测数据包含更丰富的时序特征, 所以集成模型 PEAD 更有助于提高模型性能.

为了再次验证 FFT 生成的外部知识对模型异常检测能力的影响, 如图 7 所示, 上面的曲线表示 Transformer

FFT 模型在 SWaT 数据集第 0 维度上生成的预测曲线(虚线)和真实曲线(实线), 下面的曲线表示 Transformer w/o FFT 模型在 SWaT 数据集第 0 维度上生成的预测曲线和真实曲线, 从图 7 中可以看出, Transformer FFT 模型的预测曲线更接近真实曲线的局部趋势, 说明模型引入 FFT 生成的外部知识可以有效关注到时序数据的局部趋势, 从而提高模型的异常检测能力.

表 3 PEAD、Transformer FFT、LSTM、Transformer w/o FFT 在所有数据集上的得分(最佳 $F1$ 得分和 AUC 分数用粗体表示)

Method	UCR				MBA				MSL			
	P	R	AUC	$F1$	P	R	AUC	$F1$	P	R	AUC	$F1$
Ours (PEAD)	0.7817	0.9999	0.9973	0.8775	0.9698	1.0000	0.9920	0.9847	0.9064	0.9999	0.9918	0.9509
Transformer FFT	0.5495	0.9999	0.9921	0.7093	0.8672	0.9999	0.9608	0.9289	0.8757	0.9999	0.9887	0.9337
LSTM	0.6066	0.9999	0.9938	0.7551	0.8829	0.9999	0.9660	0.8829	0.8564	0.9999	0.9867	0.9226
Transformer w/o FFT	0.5311	0.9999	0.9915	0.6937	0.8664	0.9999	0.9605	0.9284	0.9038	0.9999	0.9916	0.9495
Method	SWaT				SMD				MSDS			
	P	R	AUC	$F1$	P	R	AUC	$F1$	P	R	AUC	$F1$
Ours (PEAD)	0.9646	0.7624	0.8791	0.8517	0.9999	0.9999	1.0000	0.9999	0.9999	0.8026	0.9012	0.8904
Transformer FFT	0.9977	0.6879	0.8438	0.8143	0.9989	0.9974	0.9986	0.9981	0.9999	0.8026	0.9013	0.8905
LSTM	0.9999	0.6770	0.8385	0.8074	0.9531	0.9974	0.9962	0.9747	0.9999	0.7964	0.8982	0.8867
Transformer w/o FFT	0.9718	0.6957	0.8463	0.8109	0.9989	0.9974	0.9986	0.9981	0.9999	0.8007	0.9003	0.8893

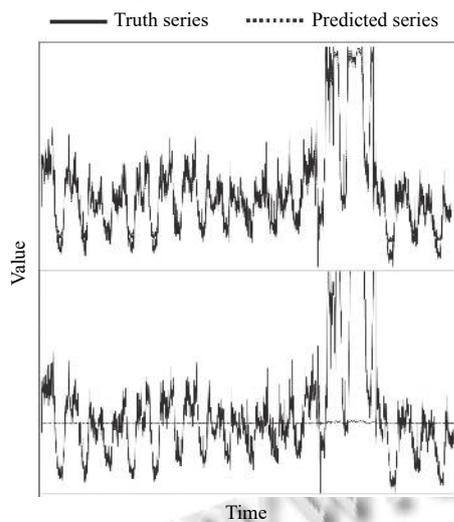


图 7 Transformer FFT (上) 和 Transformer w/o FFT (下) 在 SWaT 数据集第 0 维度上的预测曲线和真实曲线

4 结论

本文提出了一种基于 PSO 和外部知识的异常检测模型 PEAD, 并在 UCR、MBA、MSL、SWaT、SMD 和 MSDS 数据集上进行了异常检测实验, 实验结果表明, PEAD 模型在大部分数据集上比只关注单一时序特征的 MERLIN、LSTM-NDT、OmniAnomaly 和 TranAD 模型的异常检测效果要好. 本实验得出如下结论.

(1) 使用 PSO 集成学习的方式可以使模型共同关注时序数据的全局特征和时间特征, 增强模型特征提取能力, 丰富时序特征, 从而提高模型异常检测能力.

(2) 基模型引入 FFT 生成的外部知识, 增强模型对时序数据局部趋势的建模能力, 从而提高模型整体异常检测能力.

本文提出的模型具有一定的实际意义: 与传统模型比较, 本文提出的模型提高了异常检测的准确率; 与深度模型比较, 本文提出的模型在提高准确率的同时保持模型架构的简洁, 丰富了 PSO 算法在集成学习方面的研究; 在不同维度的数据集上, 本文提出的模型表现良好, 具有一定的泛化能力, 可以迁移到其他领域进行更深入的研究. 虽然 PEAD 模型在大部分数据集上表现良好, 但在个别数据集上表现不佳, 除了受限于基模型的性能外, 另一方面是因为 PSO 算法有一定的概率会落入局部最优解, 模型通过局部最优解无法分配基模型的最佳权重. 后续将继续深入研究 PSO 算法和基模型之间的联系, 尝试解决算法陷入局部最优解的问题.

参考文献

- 马浩轩, 杨笑千, 廖真, 等. 时序数据异常检测算法研究. 科学技术创新, 2022(6): 78–81. [doi: 10.3969/j.issn.1673-1328.2022.06.020]

- 2 Choi K, Yi JH, Park C, *et al.* Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines. *IEEE Access*, 2021, 9: 120043–120065. [doi: [10.1109/ACCESS.2021.3107975](https://doi.org/10.1109/ACCESS.2021.3107975)]
- 3 徐天慧, 郭强, 张彩明. 基于全变分比分隔距离的时序数据异常检测. *计算机科学*, 2022, 49(9): 101–110.
- 4 吴星辰. 基于KNN算法的城市轨道车辆时序数据异常检测. *智能城市*, 2021, 7(22): 20–21. [doi: [10.19301/j.cnki.znzs.2021.22.007](https://doi.org/10.19301/j.cnki.znzs.2021.22.007)]
- 5 闫妮锦, 夏元清, 张宏伟, 等. 一种非规则采样航空时序数据异常检测方法. *航空学报*, 2021, 42(4): 525019.
- 6 宋文婧, 周海波, 吴宗培, 等. 图记忆诱导的大气排污时序数据异常检测算法. *华东理工大学学报(自然科学版)*, 2023, 49(3): 341–350. [doi: [10.14135/j.cnki.1006-3080.20221118001](https://doi.org/10.14135/j.cnki.1006-3080.20221118001)]
- 7 刘明群, 何鑫, 覃日升, 等. 基于改进K-means聚类k值选择算法的配网电压数据异常检测. *电力科学与技术学报*, 2022, 37(6): 91–99. [doi: [10.19781/j.issn.1673-9140.2022.06.010](https://doi.org/10.19781/j.issn.1673-9140.2022.06.010)]
- 8 Yaacob AH, Tan IKT, Chien SF, *et al.* ARIMA based network anomaly detection. *Proceedings of the 2nd International Conference on Communication Software and Networks*. Singapore: IEEE, 2010. 205–209.
- 9 Nakamura T, Imamura M, Mercer R, *et al.* MERLIN: Parameter-free discovery of arbitrary length anomalies in massive time series archives. *Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM)*. Sorrento: IEEE, 2020. 1190–1195.
- 10 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.
- 11 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.
- 12 Smagulova K, James AP. A survey on LSTM memristive neural network architectures and applications. *The European Physical Journal Special Topics*, 2019, 228(10): 2313–2324. [doi: [10.1140/epjst/e2019-900046-x](https://doi.org/10.1140/epjst/e2019-900046-x)]
- 13 Tuli S, Casale G, Jennings NR. 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)]
- 14 徐继伟, 杨云. 集成学习方法: 研究综述. *云南大学学报(自然科学版)*, 2018, 40(6): 1082–1092. [doi: [10.7540/j.ynu.20180455](https://doi.org/10.7540/j.ynu.20180455)]
- 15 陈仲磊, 伊鹏, 陈祥, 等. 基于集成学习的系统调用实时异常检测框架. *计算机工程*, 2023, 49(6): 162–169, 179. [doi: [10.19678/j.issn.1000-3428.0065643](https://doi.org/10.19678/j.issn.1000-3428.0065643)]
- 16 Li JB, Izakian H, Pedrycz W, *et al.* Clustering-based anomaly detection in multivariate time series data. *Applied Soft Computing*, 2021, 100: 106919. [doi: [10.1016/j.asoc.2020.106919](https://doi.org/10.1016/j.asoc.2020.106919)]
- 17 Kennedy J, Eberhart R. Particle swarm optimization. *Proceedings of the 1995 International Conference on Neural Networks*. Perth: IEEE, 1995. 1942–1948.
- 18 吴颖超, 胡靖. 基于快速傅里叶变换和选择性卷积核网络的图像补全. *计算机系统应用*, 2023, 32(11): 149–158. [doi: [10.15888/j.cnki.csa.009298](https://doi.org/10.15888/j.cnki.csa.009298)]
- 19 Vaswani A, Shazeer N, Parmar N, *et al.* Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach: Curran Associates Inc., 2017. 6000–6010.
- 20 Siffer A, Fouque PA, Termier A, *et al.* Anomaly detection in streams with extreme value theory. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Halifax: ACM, 2017. 1067–1075.
- 21 Boniol P, Palpanas T, Meftah M, *et al.* GraphAn: Graph-based subsequence anomaly detection. *Proceedings of the VLDB Endowment*, 2020, 13(12): 2941–2944. [doi: [10.14778/3415478.3415514](https://doi.org/10.14778/3415478.3415514)]
- 22 Dau HA, Bagnall A, Kamgar K, *et al.* The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 2019, 6(6): 1293–1305. [doi: [10.1109/JAS.2019.1911747](https://doi.org/10.1109/JAS.2019.1911747)]
- 23 Goldberger AL, Amaral LAN, Glass L, *et al.* PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 2000, 101(23): e215–e220.
- 24 Moody GB, Mark RG. The impact of the MIT-BIH arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 2001, 20(3): 45–50. [doi: [10.1109/51.932724](https://doi.org/10.1109/51.932724)]
- 25 Nedelkoski S, Bogatinovski J, Mandapati AK, *et al.* Multi-source distributed system data for AI-powered analytics. *Proceedings of the 8th Service-oriented and Cloud Computing: IFIP WG 2.14 European Conference*. Heraklion: Springer, 2020. 161–176.
- 26 Mathur AP, Tippenhauer NO. SWaT: A water treatment testbed for research and training on ICS security. *Proceedings of the 2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*. Vienna: IEEE, 2016. 31–36.

(校对责编: 牛欣悦)