基于深度学习的集群系统故障预测方法

姬莉霞[1,2]，　张庆开[1]，　周洪鑫[1]，　党依萍[1]，　张　晗[1]

(1. 郑州大学 网络空间安全学院　河南 郑州 450002；2. 四川大学 计算机学院　四川 成都 610065)

摘要：在面对集群系统故障预测时,长时间序列预测中存在因关键特征信息丢失而导致梯度消失或爆炸问题,从而影响了故障预测模型的准确性。基于此,提出一种新的基于深度学习的集群系统故障预测方法。该方法采用双向门控循环网络(bidirectional gate recurrent unit, BiGRU)来捕捉局部时序特征,同时采用 Transformer 来提高全局特征提取能力。通过 BiGRU 层中双向的信息传递获得集群系统日志上时序特征的动态变化,以获取集群事件中的潜在因果关系和局部时间特征,使用 Transformer 层并行处理 BiGRU 层输出的时间序列,得到全局的时间依赖性,继而由全连接神经网络层得到预测结果。通过由 Blue Gene/L 系统产生的真实日志所构建的公共数据集来验证方法的有效性,结果表明,所提方法优于对比方法,其最佳正确率和 $F1$ 值分别达到 91.69%和 92.74%。

关键词：故障预测；集群系统；特征提取；循环神经网络；Transformer；深度学习

中图分类号：TP391　　**文献标志码**：A　　**文章编号**：1671-6841(2024)05-0071-09

DOI：10.13705/j.issn.1671-6841.2023021

A Cluster System Failure Prediction Approach Based on Deep Learning

JI Lixia[1,2], ZHANG Qingkai[1], ZHOU Hongxin[1], DANG Yiping[1], ZHANG Han[1]

(1. *School of Cyber Science and Engineering*, *Zhengzhou University*, *Zhengzhou* 450002, *China*;
2. *College of Computer Science*, *Sichuan University*, *Chengdu* 610065, *China*)

Abstract: In the clustered system failure prediction, the long-time series prediction was accompanied by problem such as gradient disappearance or explosion, due to the loss of key feature information, which would affect the accuracy of the model for failure prediction. For this reason, a new model of cluster system fault prediction method based on deep learning was proposed. The method adopted bidirectional gate recurrent unit (BiGRU) to capture local timing features while employing Transformer to improve the global feature extraction capability. The dynamic changes of timing features on the cluster system logs were obtained through bidirectional information transfer in the BiGRU layer to capture the potential causality and local temporal features in the cluster events. The Transformer layer was used to process the time series output from the BiGRU layer in parallel to obtain the global temporal dependence, which followed by the fully connected neural network layer to obtain the prediction results. The effectiveness of the method was validated on a public dataset constructed from real logs generated by the Blue Gene/L system. The results showed that the proposed method outperformed the comparison methods with a best-correct rate and $F1$ value of 91.69% and 92.74%, respectively.

Key words: failure prediction; cluster system; feature extraction; recurrent neural network; Transformer; deep learning

收稿日期:2023-01-28
基金项目:国家自然科学基金项目(52179144);河南省重大科技专项(201300210500);郑州市重大科技创新专项(2020CXZX0053)。
第一作者:姬莉霞(1979—),女,副教授,主要从事多模态学习和数据智能研究,E-mail:jilixia@zzu.edu.cn。
通信作者:张晗(1985—),女,讲师,主要从事知识工程和信息安全研究,E-mail:zhang_han@zzu.edu.cn。

0 引言

目前大多数针对集群系统的故障预测引擎是基于系统日志来构建的[1],这是由于其包含集群系统实时状态的各种事件日志,可以较长时间且更准确地记录系统行为。同时,日志中的事件之间存在明显的相关性,并且系统的故障事件表现出明显的时间相关性[2]。

随着人工智能学科的兴起,深度学习被应用于故障预测领域[3-4],该类方法通过神经网络模型深度挖掘事件的时间关联性,对系统行为和历史状态进行建模分析,从而预测系统未来是否会发生故障及可能出现的故障类型,实现对系统故障的精准预测。这些方法更加关注故障特征与故障趋势的关系,虽然在特征提取能力方面有了进一步的提升,但随着集群规模的日益扩大,集群故障预测序列长度增加,这些方法往往伴随着梯度消失或爆炸问题[5],从而引起集群实时状态或一些时间点等关键信息的丢失。

循环神经网络(recurrent neural network,RNN)在提取时序中的时间相关性方面能力突出,但由于其自身顺序结构的局限性,只能实现局部因果时间相关性的特征提取[6],在面对长时间序列预测时会遗忘部分信息,从而导致梯度消失或爆炸问题。随着 Transformer 模型[7]在自然语言处理领域的深度应用,其展示出对长时序数据的强大建模能力,但其自注意力机制在处理局部特征时可能会因过于关注全局信息而忽视了局部细节,弱化了模型捕捉局部特征的能力[8]。

受此启发,采用 Transformer 来学习长时间序列的全局特征,与 RNN 学习长时间序列的局部特征相互补,以解决长时间序列中的重要特征丢失问题。同时,双向循环网络模型对文本序列上下文特征的提取有着优越的性能表现,能够进一步提升局部因果时间相关性的提取能力。因此,本文提出一种基于 Transformer 与 BiGRU 的集群故障预测方法,简称为 TBGRU。该方法首先通过 BiGRU 和 Transformer 相结合的方式来获取故障序列的局部时间特征和全局时间依赖性,然后通过全连接神经网络层输出预测结果。TBGRU 不仅可以捕获局部的时间依赖性和时序数据的因果关系,还可以捕获整体时间内事件的时序关系,并抓取长时的依赖信息,从而解决了目前研究中普遍存在的在预测长时间序列时会发生的梯度爆炸或消失问题。在 Blue Gene/L 系统的真实日志数据集中,相比其他基线模型,TBGRU 在集群故障预测中具有更好的性能表现。

1 相关工作

针对集群系统故障预测的研究主要分为两类:一类是基于传统的统计和规则基准的预测方法;另一类是基于人工智能技术的预测方法。其中第一类方法大多为基于系统日志的故障预测方法[9],通过跟踪和分析反映系统状态变化过程的系统日志来达到故障预测的目的。例如,王卫华等[10]提出一种基于频繁日志事件序列聚类的故障预测方法。Fu 等[11-12]使用 Apriori-LIS 和 Apriori-simiLIS 算法来挖掘日志事件之间的关系,并提出了事件关联图来表示事件之间的规则,进而预测可能发生的故障事件。但在数据特征挖掘阶段,这些方法大部分并未全面考虑事件之间的时间相关性,忽略了故障之间的因果关系对预测能力的影响,在预测细粒度的故障时,往往因粗糙的提取特征而影响预测性能。

基于统计的机器学习和基于神经网络的深度学习方法也被用于故障预测领域。Liang 等[13]针对 IBM 的 Blue Gene/L 集群系统日志进行研究,采用基于规则挖掘的分类算法 RIPPER、支持向量机、k-近邻和自定义最近邻方法分别构建了故障预测模型,进行二分类预测。王振华[14]在此基础上增强了日志特征提取能力,并且选择合适的分类器,使用贝叶斯网络、随机森林、AdaBoostSVM 自适应提升算法等构建分类预测模型。Mohammed 等[15]提出一种基于时间序列和机器学习的故障预测模型。

上述方法通过挖掘事件之间的时序特征,大大提高了模型对故障预测的精度。但在面对长时间序列时存在因部分关键信息丢失而导致梯度消失或爆炸等问题,降低了故障预测精度。Vaswani 等[16]放弃了 RNN 和 CNN,提出了完全基于全连接层和注意力机制的 Transformer。注意力机制在解决长序列信息丢失的问题中是有效的,并且在许多领域的基本问题上取得了最先进的性能[17-18]。但 Transformer 通过位置编码来实现序列特征的提取,这与 RNN 等自然序列特征提取器在特征提取能力上存在一定差距。因此,本文提出结合 Transformer 和 BiGRU 的 TBGRU 模型。在该模型中,Transformer 的多头自注意力机制和残差连接能更好地处理长时间序列信息特征丢失的问题,同时,BiGRU 的双向叠加设计使得模型能够更好地获得当前时间点的上下文信息,并学习其中的因果关系,进而解决深层次的特征挖

掘问题。

2　故障预测模型

2.1　问题定义

集群系统中的故障预测问题可以描述为:通过输入时长为 S 的历史时刻日志中事件的实时数据来预测接下来 T 时刻内的集群实时状态。选择一个长度为 L 的滑动窗口来定义原始向量序列 $\boldsymbol{X}$ 的特征序列,$\boldsymbol{X}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n)$。历史值或真实值由 $\boldsymbol{Y}$ 给出,$\boldsymbol{Y}=(\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_{n-1})$,其中 $\boldsymbol{y}_i\in\mathbf{R}^{d_L}$。通过将时间序列特征 $\boldsymbol{X}$ 经过 TBGRU 模型的训练来得到预测估计值 $\hat{\boldsymbol{Y}}=(\hat{\boldsymbol{y}}_1,\hat{\boldsymbol{y}}_2,\cdots,\hat{\boldsymbol{y}}_n)$。这里的集群系统状态包含故障、可恢复故障和非故障等一系列信息,使用系统日志中的实时状态作为集群系统是否故障的一种表示方式。

2.2　TBGRU 模型框架

模型由数据预处理层、BiGRU 层、Transformer 层和故障预测层组成。首先通过数据预处理将原始数据的关键信息进行向量化,然后将序列数据输入 BiGRU 中,捕获日志事件中局部时间依赖性和时序数据的因果关系。由 Transformer 对经过 BiGRU 层处理后的特征序列信息进行再处理,使得处理后的序列获得序列信息中的全局时序特征。最后,以 Transformer 层输出的最终状态作为分类的输入,输出到全连接神经网络层继而得到预测概率。TBGRU 模型框架如图 1 所示。

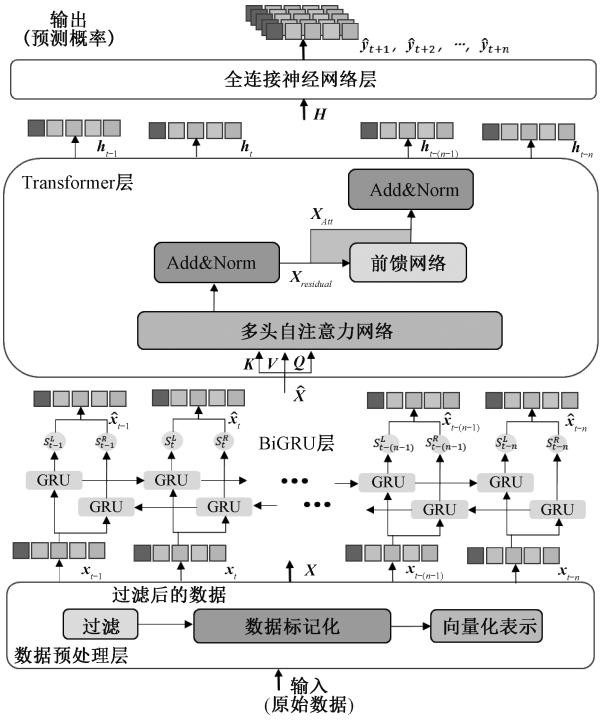

图 1　TBGRU 模型框架

Figure 1　TBGRU model framework

2.2.1　*数据预处理层*　在数据预处理阶段,将原始数据的多元特征映射到向量序列 $\boldsymbol{X}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n)$,其中:$\boldsymbol{x}_i\in\mathbf{R}^{d_R}$,$d_R$ 为特征在映射后的向量表示中的维数;n 为数据数量。原始数据到特征向量主要由过滤、数据标记化和向量化表示三部分组成。

原始数据通常包含大量冗余的记录以及与故障症状无关的正常系统记录,会影响故障预测的效率和准确性。因此,在数据预处理阶段主要完成以下任务。① 过滤冗余的数据信息。原始数据包含一些与故障预测无关的数据信息,如事件的描述、事件发生的地点等,只保留事件类型、故障级别和时间戳三个方面的信息。② 对原始数据进行标记化。由于复杂的集群环境和故障症状,将事件类型分为六类:APP(应用程序)、HARDWARE(硬件)、KERNEL(内核,一般与内存或网络相关)、LINKCARD(中间件通信)、DISCOVERY(资源更新和初始配置)、MONITOR(电源、温度等异常监控);将故障级别分为三类:无故障、可自愈的轻微故障、严重故障。并且,将事件类型和故障级别两个维度的信息进行融合。③ 对处理好的数据进行向量化表示。分别用不同的向量表示在不同时间内每种事件的发生。

为了消除各项指标之间的量纲影响,需要通过数据标准化来解决数据指标之间的可比性。采用最大最小标准化方法对原始数据进行归一化,使原始数据映射到[0,1],具体公式为

$$x_i=\frac{\tilde{x}_i-\tilde{x}_{\min}}{\tilde{x}_{\max}-\tilde{x}_{\min}},\tag{1}$$

其中:x_i 为归一化后的数据;$\tilde{x}_i$ 为原始数据;$\tilde{x}_{\max}$ 为原始数据中的最大值;$\tilde{x}_{\min}$ 为原始数据中的最小值。将每种故障类型作为输入,向量序列 $\boldsymbol{X}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n)$ 作为输出。经过上述处理后,将原始数据转换为矢量序列 $\boldsymbol{X}$,继而输出给 BiGRU 层提取时序特征信息。

2.2.2　*BiGRU 层*　BiGRU 是在传统 GRU 网络的基础上扩展了第二隐藏层,通过对序列进行正向和反向扫描来获取过去和未来的上下文信息。这种模型对输入数据的依赖性小,具有复杂度低、响应时间快等优点。对于 t 时刻的输入序列 $\boldsymbol{x}_t(\boldsymbol{x}_t\in\boldsymbol{X})$,经过 BiGRU 处理后可得到对应的输出 $\hat{\boldsymbol{x}}_t$,继而组成输出序列 $\hat{\boldsymbol{X}}=(\hat{\boldsymbol{x}}_1,\hat{\boldsymbol{x}}_2,\cdots,\hat{\boldsymbol{x}}_n)$,

$$\boldsymbol{u}_t = \text{sigmoid}(\boldsymbol{x}_t\boldsymbol{W}_z + \boldsymbol{h}_{t-1}\boldsymbol{U}_z), \quad (2)$$

$$\boldsymbol{r}_t = \text{sigmoid}(\boldsymbol{x}_t\boldsymbol{W}_r + \boldsymbol{h}_{t-1}\boldsymbol{U}_r), \quad (3)$$

$$\widetilde{\boldsymbol{h}}_t = \tanh(\boldsymbol{W}\cdot[\boldsymbol{r}_t \times \boldsymbol{h}_{t-1}], \boldsymbol{x}_t), \quad (4)$$

$$\boldsymbol{h}_t = (1-\boldsymbol{u}_t)\boldsymbol{h}_{t-1} + \boldsymbol{u}_t * \tanh(\boldsymbol{x}_t\boldsymbol{W}_h + (\boldsymbol{h}_{t-1}\boldsymbol{r}_t) * \boldsymbol{U}_h), \quad (5)$$

$$\hat{\boldsymbol{x}}_t = [\overrightarrow{\boldsymbol{h}_t} \oplus \overleftarrow{\boldsymbol{h}_t}], \quad (6)$$

其中：$\boldsymbol{r}_t$ 为复位门；$\boldsymbol{u}_t$ 为更新门；$\boldsymbol{x}_t$ 为 t 时刻的输入向量；$\boldsymbol{h}_{t-1}$ 表示 $t-1$ 时刻的状态信息；$\widetilde{\boldsymbol{h}}_t$ 表示候选隐藏状态；$\boldsymbol{h}_t$ 表示隐藏状态；$\boldsymbol{W}$ 和 $\boldsymbol{U}$ 为需要训练的权值矩阵。

2.2.3 Transformer 层　将经过 BiGRU 处理后得到的特征向量序列 $\hat{\boldsymbol{X}}$ 作为 Transformer 模型的输入，生成状态序列 $\boldsymbol{H} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n)$。如图 1 所示，变换编码器主要分为多头自注意力网络和前馈网络，计算公式为

$$multiHead(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = concat(Att_1, Att_2, \cdots, Att_n), \quad (7)$$

$$Att_1 = \text{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}})\boldsymbol{V}, \quad (8)$$

其中：$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$ 分别表示查询、键和值，均为输入矩阵；d_k 表示键的维数；在模型中 n 设置为 2。这里使用从 BiGRU 层得到的特征向量序列 $\hat{\boldsymbol{X}}$ 作为 $\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$，然后输出 $\hat{\boldsymbol{X}}_{Att}$，

$$\hat{\boldsymbol{X}}_{Att} = multiHead(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}), \quad (9)$$

$$\hat{\boldsymbol{X}}_{residual} = norm(\hat{\boldsymbol{X}}_{Att} + \hat{\boldsymbol{X}}), \quad (10)$$

$$\boldsymbol{H} = norm(\hat{\boldsymbol{X}}_{residual} + FFN(\hat{\boldsymbol{X}}_{residual}))。 \quad (11)$$

FFN 由两个线性变换和一个 ReLU 组成，

$$FFN(\hat{\boldsymbol{X}}_{residual}) = Linear(\max(0, Linear(\hat{\boldsymbol{X}}_{residual})))。 \quad (12)$$

Transformer 内部层的大小为 2 048，最后生成状态序列 $\boldsymbol{H} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n)$。使用最终状态 $\boldsymbol{h}_n$ 作为 Transformer 的输出，然后输入全连接神经网络，实现故障预测分类。

2.2.4 故障预测层　为了实现多分类故障预测，使用全连接神经网络对时间卷积层输出的结果 $\boldsymbol{H} \in \mathbf{R}^{N\times T}$ 进行线性变化处理，即将时间序列的维度转换成需要预测的时间长度，

$$\hat{\boldsymbol{Y}} = [\hat{\boldsymbol{y}}_{t+1}, \hat{\boldsymbol{y}}_{t+2}, \cdots, \hat{\boldsymbol{y}}_{t+T}] = \delta(\boldsymbol{W}_f\boldsymbol{H} + b_f), \quad (13)$$

其中：T 为预测的时间长度；$\hat{\boldsymbol{Y}} \in \mathbf{R}^{N\times T}$；$\delta(\cdot)$ 表示线性神经网络的激活函数；$\boldsymbol{W}_f \in \mathbf{R}^{2d\times T}$，为全连接神经网络的权重矩阵；$b_f$ 为偏置项。

为了进一步优化预测结果，采用软动态时间规整(Soft-DTW)[19]作为损失函数。简单地说，软动态时间规整算法可以根据两个时间序列的特征找到合适的匹配来计算两个序列的相似性，然后通过反向传播不断校正模型，最终达到最优的预测结果。

对于任意节点 $\boldsymbol{X}_i$ 的预测值 $\hat{\boldsymbol{Y}} \in \mathbf{R}^T$ 和真实标签值 $\boldsymbol{Y}_i \in \mathbf{R}^T$，损失值计算过程为

$$loss = dtw_\gamma(\hat{\boldsymbol{Y}}_i, \boldsymbol{Y}_i) = \min^\gamma\{(\boldsymbol{A}, \Delta(\hat{\boldsymbol{Y}}_i, \boldsymbol{Y}_i)), \boldsymbol{A} \in \boldsymbol{A}_{T,T}\} = -\gamma(\log(\sum_{A\in T,T} \mathrm{e}^{-(A,\Delta(\hat{\boldsymbol{Y}}_i,\boldsymbol{Y}_i)),A\in T,T)})), \quad (14)$$

其中：$\gamma \subset (0,1]$ 表示欧几里得损失值的取值范围；$\boldsymbol{A}_{T,T} \subset \{0,1\}^{T\times T}$ 表示长度均为 T 序列上的校准矩阵集合，$\boldsymbol{A} \in \boldsymbol{A}_{T,T}$ 代表一条路径。此外，这里的分类由一个 Linear 层和 logsoftmax 组成。将 Transformer 层输出的状态序列 $\boldsymbol{H} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n)$ 作为输入，最终输出模型预测的接下来一段时间集群系统的状态 $\hat{\boldsymbol{Y}} = (\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, \cdots, \hat{\boldsymbol{y}}_n)$。

2.3 模型训练

由于神经网络参数和超参数多种多样，为了减少模型训练时间，以便更好地进行验证和模型预测，执行了在算法 1 中定义的类似网格的搜索机制。算法 1 的具体步骤如下。

算法 1　TBGRU 模型超参数的调优算法

输入：原始故障数据时间序列 F，滑动窗口长度 L，隐藏层层数 H。

输出：优化的 TBGRU 模型 M_t，训练误差 ϵ_t，验证误差 ϵ_v。

1：划分 X 为训练集 X_t 和验证集 X_v；

2：随机排列 X_t 为 (X_t)；

3：随机排列 X_v 为 (X_v)；

4：设置 flag←false；

5：for 每个时间窗口步长 $L \in \{2, 3, \cdots, L_{\max}\}$；do

6：for 每个隐藏层的数量 $H \in \{H_1, H_2, \cdots, H_n\}$ 标记为 $iteration_H$；do

7：assert：$L \geqslant 2$ and $L_{\max} \ll length\ F$；

8：设置 $M_t \leftarrow \text{TBGRU}_{\text{net}}(X_t^L, X_v^L, seeds, \eta)$；

9：计算 $\epsilon_t^L \leftarrow trainingloss\ (M_t)$；

10：计算 $\epsilon_v^L \leftarrow validationloss\ (M_t)$；

11：if $\epsilon_t^L \leqslant \epsilon_{\min}$ and $\epsilon_v^L \leqslant \epsilon_{\min}$ then

12：flag←true

13：break loops

14:　else

15:　　重复直至收敛,并且设置 flag←true;

16:　end if

17:　end for

18: end for

19: if flag←true then

20: 使用模型 M_t 进行 $\text{TBGRU}_{\text{net}}$ 预测任务;

21: else

22: 重复使用不同的条件进行实验,或者结束过程;

23: end if

算法 1 是一种实现双目标的算法,即调整 BiGRU 和 Transformer 的隐藏层层数和搜索滑动窗口时间步长,以便更好地验证和预测集群故障。其中第一层神经元数量 $N \in \{16,32,64,128,256\}$,隐藏层的层数 $H \in \{16,32,64,128,256\}$,学习率 $\eta \in \{0.1, 0.2, 0.5, 0.8, 1.0\}$。同时,TBGRU 模型采用的激活层函数为 ReLU,损失函数为 Soft-DTW,优化器函数为 Adam,最后一层激活层为 sigmoid,Batch size 为 64。

3　实验与分析

3.1　实验数据

实验采用 Blue Gene/L 集群系统产生的系统日志数据,Blue Gene/L 数据集是由从 Lawrence Livermore 国家实验室(LLNL)部署的 Blue Gene/LHPC 系统中收集到的事件日志组成,日志记录包括致命告警和非致命告警,是故障检测和预测研究中常用的数据集。该数据集可以从公共计算机故障仓库(computer failure data repository,CFDR)下载[20]。日志容量为 708.8 MB,共包括 4 399 503 条记录。

将数据集分为训练集与测试集,其中测试集占 20%,训练集占 80%,进行参数优化和模型选择,并评估模型的泛化能力,以提高模型的性能和预测效果。

3.2　评价指标和参数设置

3.2.1　评价指标　为了评估 TBGRU 方法的有效性,使用了 3 个性能指标:平均绝对误差(*MAE*)、均方根误差(*RMSE*)和平均绝对百分比误差(*MAPE*)。数学表达式为

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_T^2} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} (\hat{y} - y)^2}, \quad (15)$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |e_T| = \frac{1}{n}\sum_{t=1}^{n} (\hat{y} - y), \quad (16)$$

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n} \left(\frac{\hat{y} - y}{y}\right), \quad (17)$$

式中:$e_T = \hat{y} - y$。

同时,采用准确率(*Accuracy*)和 *F*1 值对故障预测结果进行综合评估。数学表达式为

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (18)$$

$$F1 = \frac{2 \cdot TP}{2 \times TP + FP + FN}, \quad (19)$$

其中:*TP* 是预测正确的故障事件个数;*TN* 是预测错误的故障事件个数;*FP* 是预测发生但实际未发生的故障事件个数;*FN* 是未预测出但实际发生的故障事件个数。

3.2.2　参数设置　实验环境配置如下:操作系统为 Windows 10;GPU 为 NVIDIA RTX3090 24 GB 显存;内存 64 GB;编程语言为 Python 3.9;深度学习框架为 Pytorch 1.3 稳定版。使用 Python 库加载数据集,设置模型的各种参数。通过算法 1 进行了参数设置的实验,实验结果如表 1~3 所示。

表 1　优化滑动窗口步长的实验结果

Table 1　Experimental results of optimizing the sliding window steps

序号	*L*	*MAE*	*RMSE*	*MAPE*/%
1	3	1.34	1.29	14.87
2	12	0.91	1.02	10.77
3	64	0.25	0.32	6.26
4	128	0.15	0.19	4.76
5	256	0.38	0.41	8.32

表 2　优化隐藏层层数的实验结果

Table 2　Experimental results of optimizing the number of hidden layers

序号	*H*	*MAE*	*RMSE*	*MAPE*/%
1	16	0.48	0.51	5.87
2	32	0.28	0.32	5.26
3	64	0.19	0.20	4.56
4	128	0.25	0.31	4.98
5	256	0.23	0.29	4.67

表 3　优化评估函数的实验结果

Table 3　Experimental results of optimizing evaluation functions

序号	优化器函数	*MAE*	*RMSE*	*MAPE*/%
1	Adam	0.16	0.19	4.32
2	Nadam	0.61	0.72	6.93

在搜索最优配置的实验中,设置 *L* 为每一行中向前滑动窗口的步长,$L \in \{3,12,64,128,256\}$,保留一个参数作为变量,其他参数保持不变。从表 1 可以看出,当 *L* 为 128 时,实验效果优于其他参数设定。

设置隐藏层层数 $H \in \{16,32,64,128,256\}$,从表2可以看出,当 H 为64时,结果较好。

从表3的优化评估函数的实验结果可以看出,优化器函数 Adam 比 Nadam 表现得更好。

3.2.3 基线模型 主要包括以下基线模型。

1) RF[21]:随机森林算法(random forest,RF)是基于决策树的集成学习算法模型。

2) LR[15]:Logistic 回归算法(Logistic regression,LR)是用于不平衡样本分类的回归算法模型。

3) SVM[22]:支持向量机(support vector machine,SVM),多分类支持向量机算法进行回归任务。

4) RNN[3]:循环神经网络(recurrent neural network,RNN)是处理序列数据的神经网络。

5) LSTM[23]:长短期记忆网络(long short-term memory, LSTM)是一种 RNN 的特殊类型,通过门控机制学习长期依赖信息。

6) GRU[24]:门控循环单元(gated recurrent unit,GRU)是 LSTM 的一个变体,GRU 在保持了 LSTM 效果的同时又使结构更加简单。

7) Transformer[16]:基于自注意力机制的神经网络模型。

8) LogTrans[17]:Transformer 变种模型,在自注意力模型中引入了稀疏偏差,提出卷积自注意力。

9) Informer[25]:修改了 Transformer 的结构,隐式地引入稀疏偏差的一种长期预测模型。

上述基线模型的实验参数设置如下:将批处理大小和隐藏层层数分别设置为128和64,第一层神经元数量为256,序列滑动窗口步长为128,预测窗口步长为128,epoch 为100,Batch size 为64,学习率为0.8。对于结合双向循环模型和注意力机制的模型,单个方向的隐藏层层数设置为128,学习率设置为0.8。在 TBGRU 上使用相同的设置,并将 Transformer 的 dropout 设置为0.3。

3.3 实验结果和分析

3.3.1 模型性能评估 不同模型的评价指标结果如表4所示。可以看出,RF 的预测性能在 *MAE*、*RMSE* 和 *MAPE* 指标上的表现不如其他模型。使用 SVM 和 LR 等传统机器模型的结果虽然稍好于 RF,但与其他基于深度学习的方法相比仍有较大的差距。在深度学习方法中,因为 RNN、LSTM 和 GRU 只捕获了局部时间特征,在长期预测的过程中会造成部分信息遗忘,所以模型的预测性能不佳。Transformer 引入了注意力机制,更加关注对关键特征信息的记忆,在长时间序列预测的3项评价指标上优于 RNN、LSTM 和 GRU 模型,其 *MAE*、*RMSE* 和 *MAPE* 分别达到0.19、0.26 和6.15%。引入了卷积自注意力机制的 LogTrans,可以使局部上下文更好地联系关键特征,模型的预测性能得到进一步提升。Informer 通过修改 Transformer 内部结构和生成式解码器来直接产生长期预测,提升了预测性能,其 *MAE*、*RMSE* 和 *MAPE* 分别达到0.15、0.20 和5.03%。

表4 不同模型的评价指标结果

Table 4 Evaluation index results of different models

模型	*MAE*	*RMSE*	*MAPE*/%
RF	1.24	0.87	13.82
LR	1.09	0.79	11.75
SVM	0.64	0.78	9.82
RNN	0.32	0.30	8.52
LSTM	0.25	0.27	7.49
GRU	0.26	0.26	7.79
Transformer	0.19	0.26	6.15
LogTrans	0.17	0.22	5.73
Informer	0.15	0.20	5.03
TBGRU	0.13	0.17	4.46

本文的 TBGRU 是在 BiGRU 的基础上引入了 Transformer,不仅捕获了数据序列的局部时间特征,并且通过 Transformer 层来捕获全局时间依赖性,使得 TBGRU 堆叠了时间同步卷积层,可以很好地学习长程时间关系和异质性,其 *MAE*、*RMSE* 和 *MAPE* 分别达到0.13、0.17 和4.46%。与其他模型的最优性能指标相比,分别提升13.34%、15.00% 和11.33%。这证明了 TBGRU 模型可以准确地捕获集群日志数据中的局部时间依赖性和全局时间依赖性,取得了优异的预测效果。

不同模型的准确率和 $F1$ 值如表5所示。传统方法中 RF、LR 和 SVM 的准确率分别达到62.97%、66.02%和72.89%,远不如深度学习方法。深度学习基准模型 RNN、LSTM 和 GRU 中的最佳模型是 GRU,其准确率达到84.23%。引入了注意力机制的 Transformer 模型,其准确率和 $F1$ 值进一步提升。在先进的长时间序列预测模型中,Informer 表现较好,其准确率和 $F1$ 值分别达到91.31%和90.12%。TBGRU 模型表现最佳,其准确率和 $F1$ 值分别达到92.74%和91.69%。

3.3.2 模型有效性评估 RNN 的链式结构导致在处理长时间序列数据时,存在部分信息遗忘[11]。因此,使用 TBGRU 来获取时序的局部时间依赖性和整体时间依赖性,从而解决了这个问题。TBGRU 捕捉故障特征图如图2所示。可以看出,TBGRU 的预

表 5　不同模型的准确率和 *F*1 值

Table 5　Accuracy and *F*1 values for different models

单位:%

模型	*F*1 值	准确率
RF	61.12	62.97
LR	65.54	66.02
SVM	71.91	72.89
RNN	80.96	81.11
LSTM	83.96	84.11
GRU	84.13	84.23
Transformer	87.91	88.53
LogTrans	89.22	90.01
Informer	90.12	91.31
TBGRU	91.69	92.74

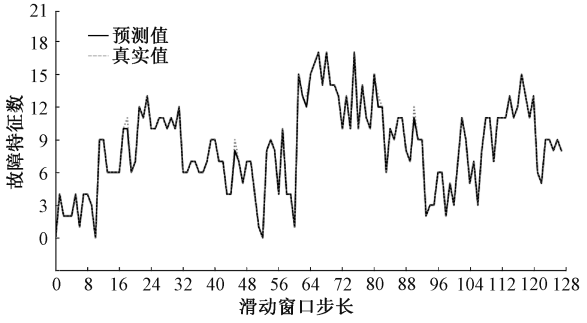

图 2　TBGRU 捕捉故障特征图

Figure 2　Capturing fault features with TBGRU

测结果与实际标签值吻合较好。

图 3 以热图的形式显示了不同模型在面对长时间序列时针对 18 个故障特征的抓取能力。颜色由浅到深代表模型抓取某一特征的能力由弱到强。可以看出,传统方法中 RF、LR 和 SVM 抓取特征的热力区域相对较“冷”,这是由于它们并没有挖掘数据序列之间的深层次特征,弱化了模型提取故障特征的能力。在深度学习方法中,RNN、LSTM 和 GRU 的表现不如引入了卷积自注意力机制的 LogTrans。但表现更为出色的是 Informer,这是由于该模型隐式地引入稀疏偏差,并设计一种生成式解码器,可以对序列信息直接进行长期预测,从而避免了长期预测时出现的累积误差,进而提升了预测性能,但在抓取某些特征时热力区域表现较“冷”。TBGRU 模型在这方面表现更好,热力区域普遍较“热”,这是由于该模型能够有效地从复杂集群系统故障中挖掘潜在的日志事件因果关系、局部时间相关性和整体时间相关性。实验证明,相比其他模型,TBGRU 捕获特征能力更为突出。

3.4　消融实验

为了进一步研究不同模块的影响,设计了 4 种模型变体,并与 TBGRU 模型进行了比较。4 种模型变体如下。

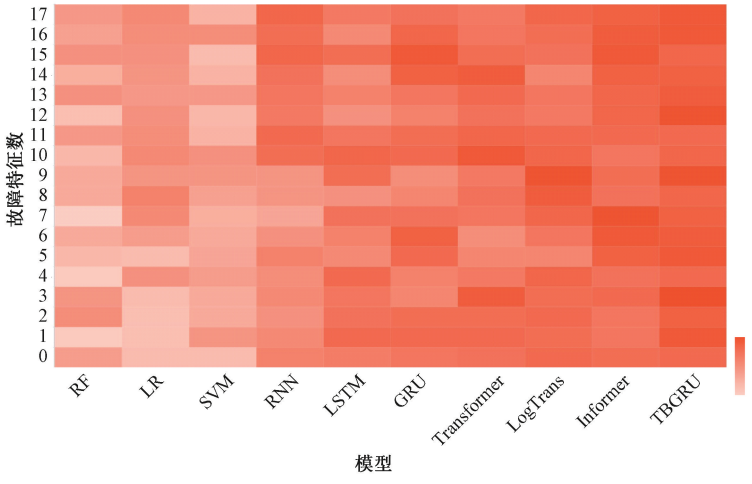

图 3　不同模型捕捉故障特征的热图

Figure 3　Heat maps for capturing fault features using different models

1) Baseline:该模型使用 GRU 捕获空间依赖性,使用 MSELoss 作为损失函数。

2) Replace_BiGRU:该模型在 Baseline 的基础上,使用 BiGRU 代替 GRU,捕获日志事件之间的时间依赖性。

3) Baseline(L-S):该模型使用 Soft-DTW 代替 MSELoss 作为损失函数。

4) Add_Transformer:该模型在 Baseline(L-S)的基础上,在 BiGRU 后增加一个 Transformer 层处理 BiGRU 输出的数据。

本文的 TBGRU 模型使用 BiGRU 获取集群系统日志之间的局部时间依赖性,利用 Transformer 捕获整体的时间依赖性,然后使用 Soft-DTW 作为模型的损失函数,最后通过全连接神经网络进行预测。消

融实验结果如表6所示。

表6 消融实验结果

Table 6 Results of ablation experiments

模型	*MAE*	*RMSE*	*MAPE*/%
Baseline	0.26	0.26	7.79
Replace_BiGRU	0.17	0.26	6.13
Baseline(L-S)	0.16	0.23	5.32
Add_Transformer	0.14	0.19	4.51
TBGRU	0.13	0.18	4.46

可以看出,使用BiGRU进行时序特征提取时要比单向GRU具有更好的性能,这是由于双向叠加的设计可以序列地进行正向和反向扫描,获取时间点过去和未来的上下文信息,提高了预测的准确度。与此同时,使用Soft-DTW作为损失函数比使用MSELoss要有明显的性能提升。对于添加了Transformer层的Add_Transformer,其具有更好的长时依赖性捕获能力,模型性能进一步提升。TBGRU模型综合了这些优点,使得其具有优秀的故障预测能力。

4 结语

针对集群系统故障预测方法在面对长时间序列预测时遇到的梯度爆炸或消失问题,提出一种新的基于深度学习的集群系统故障预测方法。该方法主要集合了Transformer的全局特征提取能力和双向循环模型BiGRU获取局部时序特征能力,同时捕获局部的时间依赖性和整体时间内事件的时序关系,并抓取长时的依赖信息,更适用于集群系统故障的长时间序列预测。使用Blue Gene/L集群系统日志数据对模型的有效性进行了验证,结果表明,与其他模型的最佳效果相比,TBGRU模型具有更好的故障预测效果。

参考文献:

[1] 郑维维. 集群系统失效预测与资源重配置方法[D]. 北京: 北京邮电大学, 2017.
ZHENG W W. Approaches for failure prediction and resource re-allocation in cluster systems[D]. Beijing: Beijing University of Posts and Telecommunications, 2017.

[2] 董婧. 基于时空关联分析的集群系统故障预测方法[D]. 北京: 北京邮电大学,2020.
DONG J. Failure prediction method of cluster system based on spatio-temporal correlation analysis[D]. Beijing: Beijing University of Posts and Telecommunications,2020.

[3] YANG Y, DONG J, FANG C, et al. FP-STE: a novel node failure prediction method based on spatio-temporal feature extraction in data centers[J]. Computer modeling in engineering and sciences, 2020, 123(3): 1015-1031.

[4] MA Y, WU S, GONG S, et al. Artificial intelligence-based cloud data center fault detection method[C]//IEEE 9th Joint International Information Technology and Artificial Intelligence Conference. Piscataway: IEEE Press, 2021: 762-765.

[5] BENGIO Y, SIMARD P, FRASCONI P. Learning long-term dependencies with gradient descent is difficult[J]. IEEE transactions on neural networks, 1994, 5(2): 157-166.

[6] WANG Z G, GAO L X, GU Y, et al. A fault-tolerant framework for asynchronous iterative computations in cloud environments[C]//IEEE Transactions on Parallel and Distributed Systems. Piscataway: IEEE Press, 2018: 1678-1692.

[7] KHAN S, NASEER M, HAYAT M, et al. Transformers in vision: a survey[J]. ACM computing surveys, 2022, 54(10): 1-41.

[8] ZHOU T, MA Z Q, WEN Q S, et al. FEDformer: frequency enhanced decomposed transformer for long-term series forecasting[C]//Proceedings of International Conference on Machine Learning. New York: ACM Press, 2022: 27268-27286.

[9] REN R, LI J H, YIN Y, et al. Failure prediction for large-scale clusters logs via mining frequent patterns[M]// Communications in Computer and Information Science. Berlin: Springer Press, 2021: 147-165.

[10] 王卫华, 应时, 贾向阳, 等. 一种基于日志聚类的多类型故障预测方法[J]. 计算机工程, 2018, 44(7): 67-73.
WANG W H, YING S, JIA X Y, et al. A multi-type failure prediction method based on log clustering[J]. Computer engineering, 2018, 44(7): 67-73.

[11] FU X Y, REN R, ZHAN J F, et al. LogMaster: mining event correlations in logs of large-scale cluster systems[C]//IEEE 31st Symposium on Reliable Distributed Systems. Piscataway: IEEE Press, 2013: 71-80.

[12] FU X Y, REN R, MCKEE S A, et al. Digging deeper into cluster system logs for failure prediction and root cause diagnosis[C]//IEEE International Conference on Cluster Computing. Piscataway: IEEE Press, 2014: 103-112.

[13] LIANG Y, ZHANG Y Y, XIONG H, et al. Failure prediction in IBM Blue Gene/L event logs[C]//Proceedings of the 7th IEEE International Conference on Data Mining. Piscataway: IEEE Press, 2008: 583-588.

[14] 王振华. 基于日志分析的网络设备故障预测研究[D]. 重庆: 重庆大学, 2015.
WANG Z H. Study on failure prediction for network equipment based on log analysis[D]. Chongqing: Chongqing University, 2015.

[15] MOHAMMED B, AWAN I, UGAIL H, et al. Failure prediction using machine learning in a virtualised HPC system and application[J]. Cluster computing, 2019, 22(2): 471-485.

[16] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all You need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: ACM Press,2017: 6000-6010.

[17] LI S Y, JIN X Y, XUAN Y, et al. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting[EB/OL]. (2019-06-29)[2022-12-21]. https://doi.org/10.48550/arXiv.1907.00235.

[18] WU H X, XU J H, WANG J M, et al. Autoformer: decomposition transformers with auto-correlation for long-term series forecasting[EB/OL]. (2022-01-01)[2022-12-21]. https://doi.org/10.48550/arXiv.2106.13008.

[19] CUTURI M, BLONDEL M. Soft-DTW: a differentiable loss function for time-series[C]//Proceedings of the 34th International Conference on Machine Learning. New York: ACM Press, 2017: 894-903.

[20] Ultrascale Systems Research Center. CFDR data[EB/OL]. (2022-02-01)[2022-11-21]. https://www.usenix.org/cfdr-data.

[21] BOTEZATU M M, GIURGIU I, BOGOJESKA J, et al. Predicting disk replacement towards reliable data centers[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2016: 39-48.

[22] XU C, WANG G, LIU X G, et al. Health status assessment and failure prediction for hard drives with recurrent neural networks[J]. IEEE transactions on computers, 2016, 65(11): 3502-3508.

[23] 王鑫, 吴际, 刘超, 等. 基于 LSTM 循环神经网络的故障时间序列预测[J]. 北京航空航天大学学报, 2018, 44(4): 772-784.
WANG X, WU J, LIU C, et al. Exploring LSTM based recurrent neural network for failure time series prediction[J]. Journal of Beijing university of aeronautics and astronautics, 2018, 44(4): 772-784.

[24] HAI Q D, ZHANG S W, LIU C, et al. Hard disk drive failure prediction based on GRU neural network[C]//IEEE/CIC International Conference on Communications in China. Piscataway:IEEE Press, 2022: 696-701.

[25] ZHOU H Y, ZHANG S H, PENG J Q, et al. Informer: beyond efficient transformer for long sequence time-series forecasting[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2021: 11106-11115.