日志信息驱动的计算机网络节点故障预测研究

王雨晞，叶庆卫，周鹏，李冰，王晓东
（宁波大学信息科学与工程学院，浙江 宁波 315211）

摘　要：针对计算机网络中节点故障对正常业务运行的影响，提出了一种以日志信息为驱动的故障预测方法，通过构建高效的深度学习模型，并引入校正机制，对计算机网络中的节点故障进行预测和诊断，支持网络运维的需求。首先收集计算机网络中各节点产生的日志信息，获得各节点的状态向量和所有节点的状态矩阵，然后通过状态填补原则补充数据集，最后将故障预测问题转换成时间序列预测问题。在公开的小规模运维数据集GAIA中进行性能评估。实验结果表明，与其他算法相比，所提模型在局部网络场景下预测效果良好，预测有效性得到了验证，为计算机网络故障预测研究提供了一定的参考价值。

关键词：日志；计算机网络；节点故障；故障预测；深度学习；校正机制；时间序列
中图分类号：TN915
文献标志码：A
doi: 10.11959/j.issn.1000−0801.2024168

Research on fault prediction of computer network nodes driven by log information

WANG Yuxi, YE Qingwei, ZHOU Peng, LI Bing, WANG Xiaodong
Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China

Abstract: A fault prediction method driven by log information was proposed to address the impact of node failures on normal business operations in computer networks. By constructing an efficient deep learning model and introducing a correction mechanism, node failures in computer networks were predicted and diagnosed to meet the needs of network operation and maintenance. Firstly, the log information generated by each node in the computer network was collected, the state vectors of each node and the state matrices of all nodes were obtained, then the dataset through the state filling principle was supplemented, and finally the fault prediction problem into a time series prediction problem was transformed. The performance evaluation is conducted on the publicly available small-scale operation and maintenance dataset GAIA, and the experimental results show that compared with other algorithms, the proposed model has good predictive performance in local network scenarios, and its predictive effectiveness is verified, providing a certain reference value for computer network fault prediction research.

收稿日期：2024−02−21；**修回日期：**2024−05−09
基金项目：浙江省产学研合作项目（No.062400020）；大型横向项目网络运维平台研发项目（No.HK2022000189）
Foundation Items: Industry-University-Research Cooperation Project of Zhejiang Province (No.062400020), Large-Scale Horizontal Project Network Operation and Maintenance Platform Research and Development Project (No.HK2022000189)

Key words: log, computer network, node failure, failure prediction, deep learning, correction mechanism, time series

0 引言

计算机网络作为现代企业和组织的核心基础设施，其稳定性和可靠性对保障业务连续性至关重要。随着计算机网络技术的广泛普及应用，越来越多的设备开始接入网络，各种设备之间交错连接，互相影响，网络结构也变得越来越复杂，当其中一台设备出现故障时，极可能导致网络中其他设备受到影响，出现新故障。网络故障可能导致通信中断、服务不可用以及数据丢失等问题，严重影响组织的正常运作。因此，在执行关键任务前，对网络中设备的状态进行观察评估，并且预测在任务期间网络中所有设备可能出现的故障状态具有重要的研究意义和实用价值[1-4]。日志数据记录了网络中一些主机节点的活动、状态信息以及事件，日志反映了节点大部分的运行状态，网络各节点之间的故障传递在日志中也有一定的体现，日志提供了丰富的信息用于故障诊断和预测[5-12]。通过分析日志数据，可以提取关键特征[13-16]，进而构建预测模型。这些预测模型可以学习故障发生前若干时间内各节点运行日志的状态信息等关键特征，从而预测未来一定时间范围内网络中是否有主机节点会发生故障，以及可能发生的故障类型。这种基于日志信息驱动的计算机网络节点故障预测方法，可以在故障发生之前及时预警，对减轻当前网络运维负担、减少网络故障所造成的损失具有重要意义。

基于日志数据的故障预测面临两个关键性技术挑战。（1）网络设备故障前的日志数据往往不具备明显的异常特征，如何从日志语义信息和微弱的异常趋势中判断设备的异常状态是一个挑战。（2）网络设备故障前往往伴随持续性的异常变化，如何尽早预测出设备故障成为另一个挑战。文献[11]提出了一种基于告警日志的网络故障预测研究方法，以基于两级时间窗口的特征提取方法构建特征表征网络运行状态，并通过大量实验来选择构建特征所需的最佳参数组合，然后设计并实现了一种基于分类学习方法的自适应故障预测模型。虽然该分类预测模型得到了比基于威布尔分布的预测模型更好的预测效果，但是针对具体设备的预测结果较差，特别是故障出现较少的设备。文献[12]提出了一种基于网络日志的无线网络故障CNN-LSTM混合预测模型，首先对网络日志进行预处理，通过两级时间窗提取样本，然后通过CNN提取样本特征，最后将提取的特征输入LSTM中进行预测，结果表明CNN-LSTM的预测性能明显优于其他机器学习模型，但是该方法只针对网络整体的故障预测，并不能具体到网络中的设备信息。

本文提供了一种日志信息驱动的计算机网络节点故障预测研究方法，旨在基于深度学习技术，通过构建高效的深度学习模型，并且引入校正机制，对由少量节点组成并且节点之间跳数有限的局部网络中的故障进行预测和诊断。从网络各节点产生的原始日志中提取关键特征得到所有节点的故障状态矩阵，通过模型学习网络中各节点的历史状态类别、各节点的故障模式和趋势，以及各节点之间的异常传导联系，经过多层次的特征提取，识别出潜在的故障迹象，实现对局部网络中各节点未来时刻故障状态的分类预测。在具体的局部网络中，本文方法对网络中各节点未来时刻的故障状态预测取得了不错的效果，证实了在局部网络场景下的预测有效性。

1 计算机网络故障预测的日志信息时间序列模型

1.1 故障预测基本原理

本文基于日志信息和深度学习技术，通过构

建高效的深度学习网络模型，对计算机网络中的节点故障进行预测和诊断。首先，当计算机网络出现故障时，收集网络中各节点的历史运行日志；其次，将收集的历史日志通过正则表达式提取关键特征[17]，得到所有节点的故障状态矩阵；然后，将得到的故障状态矩阵进行数据预处理，其中包括节点前向状态填补、构造输入输出对、划分训练集和测试集；最后，将经过处理后的数据送入时间序列预测模型进行训练学习，不断调整模型的参数以达到更好的预测效果，实现对未来时刻计算机网络节点的故障预测。

1.2　计算机网络日志信息关键特征提取

日志记录着网络某节点在某个时间点执行了某些操作以及相应操作的结果，日志中记录的时间戳、节点IP地址、节点状态描述信息等关键特征可以帮助系统开发人员与运维人员监控系统，并剖析系统的异常和错误，从而快速定位异常、修复异常，以维护网络的稳定性。针对如何从历史日志信息文本串中得到在具体时间下具体节点对应的状态类型问题，本文设计了3种正则表达式，分别为regex1、regex2、regex3，用以从一般性的日志信息文本串中提取出3个关键特征：时间戳、节点IP地址、节点状态描述信息。

Regex regex1 = new Regex (“\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}”);

Regex regex2 = new Regex (“\b(?:[0-9]{1,3}\.){3}[0-9]{1,3}\b”);

Regex regex3 = new Regex (“\[\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}\] [A-Z]+: (.+)”);

对节点状态描述信息部分进行语义分析，做出状态评价，可以得到其对应的一种状态类别s，s的取值一般为离散值，设定如式（1）所示。

$$s=\begin{cases}0, & \text{正常}\\ 1, & \text{网络异常}\\ 2, & \text{系统异常}\\ 3, & \text{其他异常}\end{cases} \tag{1}$$

1.3　计算机网络节点故障状态矩阵和时间序列模型

对于局部子网来说，故障预测需要关心局部子网中所有的节点，把某一局部子网中的所有节点的故障状态类别组合成一个状态向量$\boldsymbol{x}_k$：

$$\boldsymbol{x}_k=\left(s_{k1},s_{k2},\cdots,s_{kM}\right) \tag{2}$$

其中，s_{k1}代表第k个时刻下第1个节点的故障状态类别，状态向量$\boldsymbol{x}_k$代表了在第k个时刻下从第1个节点到第M个节点的故障状态类别集合。

同样的，当考虑所有时刻的状态向量时，可以定义故障状态矩阵$\boldsymbol{x}$：

$$\boldsymbol{x}=\begin{pmatrix}s_{11} & \cdots & s_{1j} & \cdots & s_{1M}\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ s_{k1} & \cdots & s_{kj} & \cdots & s_{kM}\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ s_{n1} & \cdots & s_{nj} & \cdots & s_{nM}\end{pmatrix} \tag{3}$$

其中，M为网络中节点的个数，n为时间戳的个数，s_{kj}表示在第k个时刻下第j个网络节点的故障状态类别，其取值为故障类型离散值，见式（1），矩阵中的每一行代表每个时刻下所有节点的状态类别集合。

这样就可以把网络节点故障看作随时间变化的多变量时间序列模型，在具体网络中，某一个时刻的节点状态类型往往受前一个时刻或前几个时刻所有节点的影响，所以可以根据网络中前一个或前几个时刻所有节点的状态向量、状态矩阵来预测下一个时刻所有节点的状态类别，建立时间序列预测模型。网络故障预测的一阶时间序列模型为：

$$x_{k+1}=f(x_k)+\delta \tag{4}$$

其中，f代表预测算法或模型，δ是预测过程中出现的噪声误差。一阶时间序列模型只考虑用当前时刻的各节点故障状态来预测下一时刻的各节点故障状态，但是通常情况下网络中各节点的故障状态变化非常复杂，一阶时间序列模型显然比较粗糙，故障预测效果可能并不理想，因此同时考

虑多阶时间序列模型。网络故障预测的多阶时间序列模型如下。

$$\boldsymbol{x}_{k+1}=f(\boldsymbol{x}_k,\boldsymbol{x}_{k-1},\cdots,\boldsymbol{x}_{k-m+1})+\delta \tag{5}$$

其中，m 是模型的时间步阶数，即前 m 个时刻，$\boldsymbol{x}_k$ 代表状态矩阵 $\boldsymbol{x}$ 中第 k 时刻的行向量，$\boldsymbol{x}_{k+1}$ 代表预测出的下一时刻的所有节点状态值集合。具体地，即用网络中前 m 个时刻的故障状态矩阵，通过预测算法或模型来预测下一时刻所有节点的状态，属于典型的时间序列分类预测任务。

1.4 状态填补原则

在式（2）和式（3）中，要求局部子网中每个节点的每个时刻下均有状态值，而实际中，日志信息的时间是稀疏的，一个计算机网络中不同的节点可能来自不同的系统，它们的日志记录时间间隔是不一致的，即在大部分时刻下不一定有日志信息的记录，从而导致大部分时刻下无法从日志信息中提取具体节点在具体时刻下对应的故障状态，针对这种情况，本文采用前向填补原则来完善每个时刻下的节点故障状态，前向填补原则如式（6）所示。

$$s_n=\begin{cases} s_k, & \text{情况1} \\ s_{\text{new}}, & \text{情况2} \end{cases} \tag{6}$$

前向填补原则指的是一个网络节点的故障状态在新的状态值出现之前一直维持不变，式（6）中的情况 1 为第 k+1 时刻没有日志信息，此时 k+1 时刻的节点状态值与第 k 时刻的状态值保持一致，即 s_k，情况 2 为第 k+1 时刻有日志信息且提取故障类型为 s_{new}，此时 k+1 时刻的节点状态值即 s_{new}。

2 基于预测校正改进的深度学习模型

在时间序列预测领域，许多经典的模型已被广泛应用，随着大量算法的改进和算力的提升，以卷积神经网络（convolutional neural network，CNN）、循环神经网络（recurrent neural network，RNN）和 Transformer 为代表的深度学习网络在时间序列预测任务中取得了丰硕的成果[18-19]，被广泛应用于计算机视觉、自然语言处理和语音识别等场景中。

现有预测网络在各自的应用场景中表现出色，在局部网络节点故障预测的特定场景中，需要利用网络中前一个或前几个时刻所有节点的故障状态矩阵预测下一个时刻各节点的状态，这是一个典型的多变量输入多变量输出的时间序列预测问题，实际上，可以将多个变量的输出分解开来，看作多个任务，也就是多任务学习，其中每个任务就是对网络中每个节点下一时刻的预测。针对式（4）和式（5）的预测原理，本文设计了一种引入校正机制的多输入多输出时间序列预测网络，该预测网络主要包括初次预测模型和预测校正模型两个部分。

2.1 初次预测模型

初次预测模型旨在利用网络中前一个或前几个时刻的各节点故障状态矩阵，通过有效地建模时间序列数据中的复杂关系，实现对网络中各节点下一时刻状态的准确预测。该部分的设计思路是结合门控循环神经网络[20]（gated recurrent neural network，GRU）和全连接层，利用 GRU 来捕捉时间序列中的长期依赖关系，提高模型对历史信息的记忆能力，同时根据实际需求确定全连接层的个数，通过全连接层将特征映射到输出空间，确保网络能够充分学习和表达时间序列数据中的复杂关系，从而实现多变量输入多变量输出的时间序列预测。

2.2 预测校正模型

预测校正法[21-22]是一种常用的数值解决常微分方程的方法，它通常用于解决带有摩擦的非线性方程或者系统，能够提供比欧拉法精度更高的结果，并且这种方法具有较好的稳定性。预测校正法是一种多步算法，它通过在每一步使用多种不同的插值方法来提高精度，这些方法包括欧拉法及其改进方法，在预测校正法中，首先计算一

个预估值，然后使用精度更高的插值方法来计算校正值。

为了提升预测准确性，本文使用预测校正法，其过程如式（7）和式（8）所示，通过引入额外的子模型（预测校正模型），根据初次预测模型的输出进行校正，以提升整体预测的准确性。

$$\hat{y}=\mathrm{Net}_1(x,Q_1) \tag{7}$$

$$y=\mathrm{Net}_2(x,\hat{y},Q_2) \tag{8}$$

其中，$\hat{y}$为初次预测模型的输出，表示初次预测模型Net_1对输入$\boldsymbol{x}$（即节点故障状态矩阵）的直接预测，Q_1代表初次预测模型的网络权重系数集，y为预测校正模型Net_2的输出，表示对输入$\boldsymbol{x}$和初次预测模型的输出$\hat{y}$进行进一步的校正和预测，Q_2代表预测校正模型的网络权重系数集。

2.3　深度学习网络结构

初次预测模型Net_1和预测校正模型Net_2采用相同的网络结构，唯一不同的是，预测校正模型的输入是预测模块的输入与输出的合并拼接。网络结构如图1所示。

由图1和前文介绍可知，初次预测模型Net_1和预测校正模型Net_2主要由两层GRU和M个全连接层组成，全连接层个数M由具体的计算机网络节点个数决定，每个全连接层后都连接一个softmax分类器。具体地，初次预测模型Net_1的输入为计算机网络节点状态矩阵$\boldsymbol{x}$，$\boldsymbol{x}$是包含一个或多个时刻的所有节点状态集合，其输入Net_1中，首先依次经过两层GRU得到GRU的输出（output）；接着将output分别通过M个全连接层，每一层对应计算机网络中的一个节点，每层输出由ReLU激活函数激活，得到M个输出；然后将每一个输出都送入一个softmax分类器，得到一个包含各类别概率的向量，经过argmax操作找到最大概率对应的类别索引，即计算机网络中每个节点的分类结果$\mathrm{pred}_n(n\in[1,M])$；最后，将预测出来的$M$个分类结果进行合并拼接操作，得到初步预测结果$\hat{y}$。预测校正模型$\mathrm{Net}_2$的输入为$\mathrm{Net}_1$的原始输入$\boldsymbol{x}$与初步预测结果$\hat{y}$的合并值，该模块的处理过程与初次预测模型一致，最后得到更加精确的预测结果y，即下一时刻各节点的预测状态集合，如式（9）所示。

$$y=\boldsymbol{x}_{k+1}=(\mathrm{pred}_1,\mathrm{pred}_2,\cdots,\mathrm{pred}_M) \tag{9}$$

2.4　标签与损失函数设计

初次预测模型和预测校正模型的损失函数均使用交叉熵损失函数，两个模型的损失函数定义式分别如式（10）和式（11）所示。

$$\mathrm{loss}_1=\mathrm{CrossEntropy}(\hat{y},L) \tag{10}$$

$$\mathrm{loss}_2=\mathrm{CrossEntropy}(y,L) \tag{11}$$

其中，L代表两个模型的真实标签，即k+1时刻下各节点对应的真实故障状态集合。

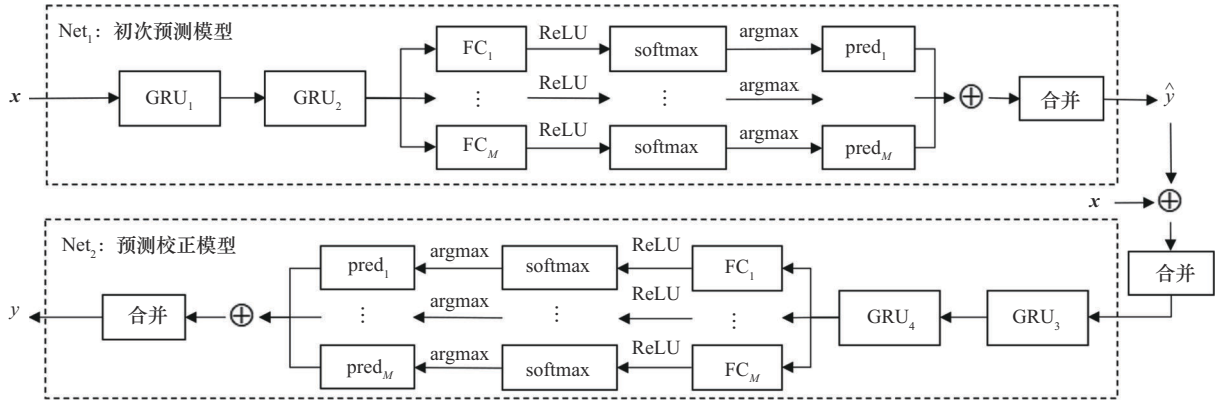

图1　网络结构

整个网络的损失函数Loss使用初次预测模型的损失函数loss_1和预测校正模型的损失函数loss_2的加权和，如式（12）所示。

$$\text{Loss}=0.5\text{loss}_1+0.5\text{loss}_2 \tag{12}$$

经过多次实验得出，将整体损失函数权重值位置为式（12）时网络性能较好。这是因为两个子模块采用了相同的网络结构，它们在任务中具有相同的重要性，所以在整体预测中以相等的比例参与，这也体现了本文对于初次预测模型和预测校正模型的信任程度是相等的。通过将两个子模块损失函数的加权和作为总损失，可以实现两个子模块参数共享，实现协同训练，有助于提升网络的性能和预测精度。

2.5 深度学习网络训练流程

深度学习网络训练流程如图2所示，主要步骤如下。

步骤1 计算机网络节点状态矩阵$\boldsymbol{x}$经过预处理后，输入初次预测模块Net_1，得到初步预测结果$\hat{y}$。

步骤2 将$\hat{y}$与真实标签L代入式（10），计算初次预测模型Net_1的损失函数loss_1。

步骤3 将初次预测模型的输出$\hat{y}$与原始输入$\boldsymbol{x}$进行合并，得到一个新的状态矩阵作为预测校正模型Net_2的输入，经过预测校正得到更加精确的预测结果y，将y与真实标签L代入式（11），计算预测校正模型Net_2的损失函数loss_2。

步骤4 将loss_1和loss_2代入式（12）得到深度学习网络整体损失函数Loss，随后进行梯度下降，更新Net_1和Net_2的网络参数。

3 实验方案与结果分析

3.1 数据集介绍及预处理

实验数据集采集自云智慧AIOps社区中的GAIA（generic aiops atlas）数据集，这是一个用于异常检测分析、日志分析、故障定位等运维问题的全量数据集。GAIA中的数据主要来自云智慧的业务模拟系统中的一个局部网络场景，收集局部网络中各节点产生的原始日志信息，并且已经人为标注好每条节点日志对应的状态类别。其中info级别的日志均为正常日志数据，本文使用数字标签0标注，error级别的日志中主要包含3种故障状态类别的错误日志，本文使用数字标签1～3标注，分别对应网络异常、数据库异常和其他异常，状态类别的标签对应于式（1）。

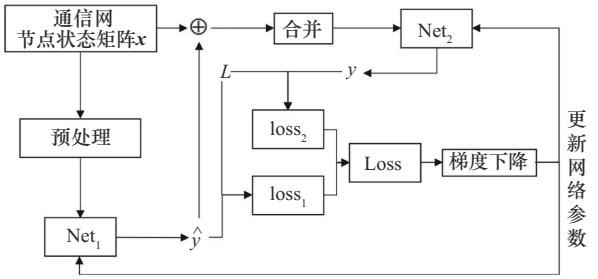

图2 深度学习网络训练流程

数据集一共收集了13个不同节点的日志信息及其对应的状态类别，本文将数据集中记录的所有节点日志发生的时间离散化表示，一共得到7 542个离散时间点。在原始数据集中，每个节点的样本个数很不均衡，在很多时刻下某些节点是没有日志信息记录的，使用状态填补原则获得这些时刻的节点状态，补充数据集。

使用状态填补原则得到完整的计算机网络节点状态数据集后，发现很多连续时间点下的所有节点状态完全相同，即在状态矩阵中有很多相同的行向量，有较多重复数据，并且这些连续时间点之间的时间间隔很小，各节点在短时间内连续产生了多条相同状态的日志描述信息。为了消除冗余造成的误差，针对这种情况，只保留其中一个时间点下各节点的状态值，这样可以避免重复数据对预测效果的影响。

经过上述处理后最终得到1 414个时间点的各节点状态数据，数据集表示为D：

$$\boldsymbol{D}=\begin{pmatrix} s_{11} & \cdots & s_{1j} & \cdots & s_{1,13} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ s_{k1} & \cdots & s_{kj} & \cdots & s_{k,13} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ s_{1414,1} & \cdots & s_{1414,j} & \cdots & s_{1414,13} \end{pmatrix} \tag{13}$$

由于计算机网络中各节点之间交错连接，互相影响，某一时刻某节点的故障状态由前一个时刻或前几个时刻所有节点的状态决定，所以需要将数据集切分成一系列的N阶时间步状态矩阵，每个状态矩阵包含了N个连续的时刻，获得的该状态矩阵作为网络的输入$\boldsymbol{X}_i$：

$$\boldsymbol{X}_i=\begin{pmatrix} s_{i,1} & \cdots & s_{i,j} & \cdots & s_{i,13} \\ s_{i+1,1} & \cdots & s_{i+1,j} & \cdots & s_{i+1,13} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ s_{i+N-1,1} & \cdots & s_{i+N-1,j} & \cdots & s_{i+N-1,13} \end{pmatrix} \tag{14}$$

该状态矩阵下一个时刻的状态向量作为网络的真实标签数据输出$\boldsymbol{Y}_i$：

$$\boldsymbol{Y}_i=\begin{pmatrix} s_{i+N,1} & \cdots & s_{i+N,j} & \cdots & s_{i+N,13} \end{pmatrix} \tag{15}$$

其中，$\boldsymbol{X}_i$为N×13的矩阵，$\boldsymbol{Y}_i$为1×13的矩阵，N为时间步阶数，$i\in[1,1\,414-N]$，$j\in[1,13]$，$s_{i,j}\in\{0,1,2,3\}$，获得$<X_i,Y_i>$数据对，然后以约8∶2的比例划分训练集和测试集。

3.2　评价指标

在本文使用的数据集中，每个节点的状态类别一共有4个，对应前文式（1）中的4种状态，网络节点故障预测任务属于一个四分类预测问题，四分类问题的混淆矩阵见表1，引入预测准确率（accuracy，Acc）、虚警率（false alarm rate，FAR）、漏警率（missed alarm rate，MAR）等性能指标来对模型性能进行评估。表1中，b、c、d、e分别代表正常、网络异常、系统异常、其他异常这4个类别的实际样本数；h、l、q、r分别代表预测结果中这4个类别的样本数；w为所有样本的总数；a_{00}、a_{11}、a_{22}、a_{33}代表被正确分类的样本数，其他的代表被错误分类的样本数。

表1　四分类问题的混淆矩阵

对比项		预测类别				
		正常	网络异常	系统异常	其他异常	小计
实际类别	正常	a_{00}	a_{01}	a_{02}	a_{03}	b
	网络异常	a_{10}	a_{11}	a_{12}	a_{13}	c
	系统异常	a_{20}	a_{21}	a_{22}	a_{23}	d
	其他异常	a_{30}	a_{31}	a_{32}	a_{33}	e
	小计	h	l	q	r	w

预测准确率作为分类预测问题最原始的评价指标，定义为模型正确预测的样本数占总样本数的百分比，如式（16）所示；虚警率表示实际为正常状态的样本中，被模型错误预测为异常类别的概率，定义如式（17）所示；漏警率也称错过检测率，是评估模型在真实异常情况下未能正确检测到异常的指标，表示在实际为异常状态（类别1、2、3）的样本中，每种异常状态的样本被模型错误预测为其他3种状态的概率，考虑各个异常类别的样本数量不均匀，故采用加权平均的方法计算漏警率，定义如式（18）所示。

$$\mathrm{Acc}=(a_{00}+a_{11}+a_{22}+a_{33})/w \tag{16}$$

$$\mathrm{FAR}=(a_{01}+a_{02}+a_{03})/b \tag{17}$$

$$\mathrm{MAR}=\frac{(c-a_{11})+(d-a_{22})+(e-a_{33})}{c+d+e} \tag{18}$$

3.3　实验结果分析

3.3.1　时间序列阶数对预测结果的影响

在经过处理后的数据集上进行实验，深度学习网络训练完成并且参数收敛后，将测试集中的样本输入网络中进行分类预测，探讨时间序列阶数对预测结果的影响。本文考虑了1阶、2阶、3阶时间步序列，在本文设计的算法中，数据集的各节点样本集在不同时间步阶数N下的测试集预测准确率、虚警率、漏警率见表2。可以看出，随着时间步阶数N的增加，大部分节点的预测准确率都有一定的提升，并且虚警率和漏警率也大多随着N的增加而降低，其中节点10.0.2.208、

表2　不同时间步阶数*N*下的测试集预测准确率、虚警率、漏警率

节点IP地址	Acc			FAR			MAR		
	N=1	*N*=2	*N*=3	*N*=1	*N*=2	*N*=3	*N*=1	*N*=2	*N*=3
10.0.1.143	90%	90%	**92%**	4.6%	6.6%	6.1%	84.5%	**57.1%**	60%
10.0.1.144	90%	**91%**	90%	2.8%	4.1%	**1.6%**	62.7%	48.6%	**33.3%**
10.0.1.145	98%	98%	**98%**	6.3%	4.2%	**1.8%**	**1.4%**	2.6%	1.8%
10.0.1.153	93%	**95%**	92%	**0.8%**	3.1%	1.5%	80%	75%	**67.4%**
10.0.2.52	97%	97%	**98%**	**0**	1.8%	0.6%	50.4%	26.3%	**11.6%**
10.0.2.67	57%	63%	**66%**	29.2%	10.3%	**7.5%**	30.6%	**12.2%**	60.4%
10.0.2.191	98%	**98%**	97%	0.8%	**0.4%**	2.9%	**0**	5.1%	11.1%
10.0.2.208	98%	98%	**98%**	26.5%	16.8%	**14.7%**	69.6%	64.7%	**7.7%**
10.0.2.220	96%	96%	**97%**	6.9%	1.2%	**0.8%**	**5.3%**	12.1%	23.1%
10.0.7.42	90%	88%	**90%**	5.3%	**1%**	1.9%	45.8%	40%	**10.2%**
10.0.7.123	95%	**97%**	96%	1.6%	1.1%	**1.1%**	31.8%	40.9%	**30.4%**
10.0.9.207	90%	92%	**97%**	26.8%	20.7%	**14.9%**	46.1%	20.6%	**5.1%**
10.0.9.208	76%	80%	**85%**	3.6%	2.3%	**0.8%**	25%	**9.1%**	11.7%
平均	89.8%	91.0%	**92.0%**	8.9%	5.7%	**4.3%**	41.0%	31.9%	**25.7%**

10.0.9.208、10.0.2.67、10.0.7.42的预测效果提升较为明显，只有少部分节点在1阶和2阶时间步下取得了更好的预测结果，综合所有节点的平均预测结果来看，当时间步阶数*N*设置为3时，整个深度学习网络的预测效果更好，其中平均预测准确率达到了92%，平均虚警率为4.3%，平均漏警率为25.7%。

本文设计的网络模型在3阶时间步模型下表现更好，这是因为3阶时间步状态矩阵考虑了更多的时间步信息，现实情况下，计算机网络中各节点在某一时刻的状态有时不仅仅只取决于前一时刻的各节点状态信息，而是可能会受前几个时刻（一段时间）的各节点状态信息的影响，所以大部分情况下，适当增加时间步阶数能更好地提升网络模型的预测效果。

为了验证以上关于时间步阶数对预测结果影响的猜想，引入经典时间序列预测（LSTM）网络、GRU、*N*-BEATS[23]（neural basis expansion analysis for interpretable time series forecasting）和ResNet50（residual network-50）进行论证实验。

不同预测算法在不同时间步阶数下的平均预测准确率、平均虚警率、平均漏警率如图3所示。由图3所示，随着时间步阶数的增加，大部分网络对所有节点的平均预测准确率呈逐步上升的趋势，平均虚警率和平均漏警率呈现逐渐减少的趋势，1阶时间步到2阶时间步的变化较为明显，2阶到3阶的提升效果较小，只有部分模型在2阶时间步下达到更好的预测效果，其中ResNet-50在2阶时间步下取得了更好的平均虚警率，LSTM、N-BEAS在2阶时间步下取得了更好的平均漏警率。但综合3种评价指标的平均预测结果来看，图3中所有网络算法均在3阶时间序列模型下取得更好的预测效果，这也验证了上面的猜想，故选定3阶时间序列模型为本文的最优时间序列模型，并在此基础上进行后续实验。

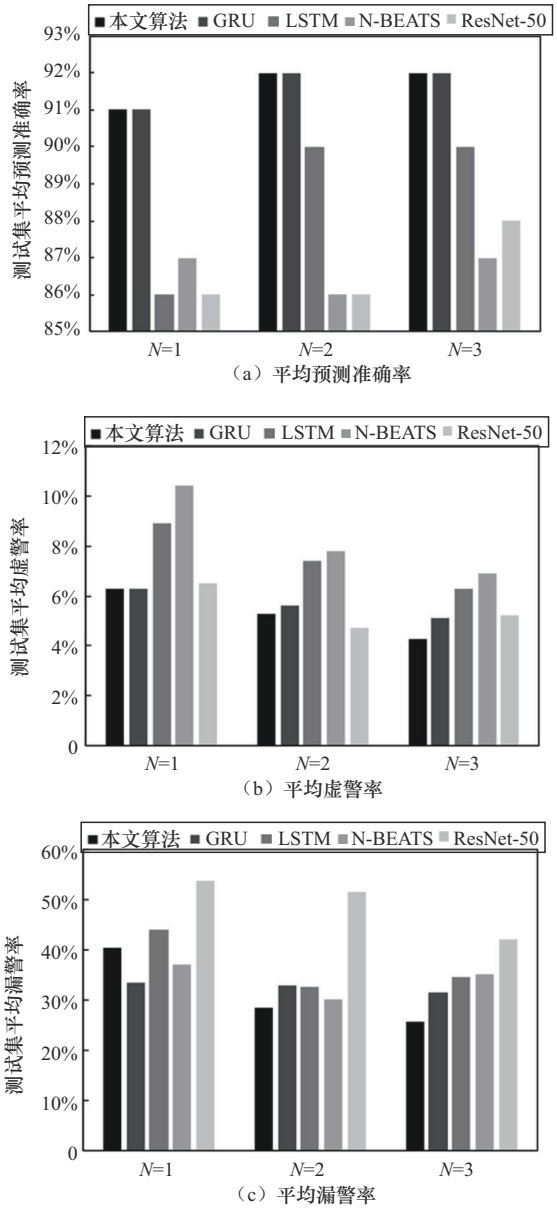

图3　不同预测算法在不同时间步阶数下的平均预测准确率、平均虚警率、平均漏警率

3.3.2　本文算法的有效性评估

为了验证本文算法中校正机制的作用，以及评估本文设计的预测校正模型在网络节点故障预测中的有效性和可行性，分别在初次预测模型、Transformer、N-BEATS、ResNet50 这些模型上进行对比实验。最优时间序列模型下测试集在不同网络上的预测准确率、虚警率、漏警率见表 3，部分节点在本文算法中的预测类别与真实类别分布情况如图 4 所示。

通过表 3 可以看出，N-BEATS 和 ResNet50 两个模型的预测效果均不是很理想，它们主要依赖于全连接和卷积操作，并不能很好地适用于小样本的多输入多输出预测任务，相较于其他网络模型，本文算法取得了更好的预测结果。本文算法在大部分节点上的预测准确率更高，但是较其他网络的预测准确率提升较小，这是因为数据集中的各节点大多处在正常状态，异常状态的记录较少，数据状态类别分布不均。对于虚警率和漏警率，本文算法在大部分节点上也取得了较好的预测效果，初次预测模型在节点 10.0.2.220 上取得了更优的虚警率和漏警率，在节点 10.0.1.143、10.0.9.208 上取得了更优的漏警率；Transformer 在节点 10.0.1.145、10.0.2.191、10.0.2.208 上取得了更优的虚警率，在节点 10.0.1.145、10.0.2.52 上取得了更优的漏警率；N-BEATS 在节点 10.0.1.153、10.0.9.207 上取得更优的漏警率；ResNet50 在节点 10.0.2.52、10.0.9.207 上取得了更优的虚警率。对于节点 10.0.2.67，所有模型的预测结果都不太理想，预测准确率最高仅为 66%，漏警率最低也高达 60.4%，结合图 4（b）来看，原因可能是该节点的异常状态变化较为频繁，网络模型无法较好地学习该节点的故障变化规律。

综合来看，通过引入校正机制，本文算法在计算机网络节点故障预测中取得了更优的平均预测准确率、平均虚警率和平均漏警率，相较于初次预测模型以及其他时间序列预测网络，本文算法有效提升了预测性能。

4　结束语

本文提出了一种日志信息驱动的计算机网络节点故障预测方法，设计了一种引入校正机制的深度学习预测模型，通过对历史节点日志信息的分析和处理，可以及时预警计算机网络中各节点未来时刻可能出现的故障状态。通过与初次预测

表3　最优时间序列模型下测试集在不同网络上的预测准确率、虚警率、漏警率

节点IP	本文算法			初次预测模型			Transformer			N-BEATS			ResNet50		
	Acc	FAR	MAR	Acc	FAR	MAR	Acc	FAR	MAR	Acc	FAR	MAR	Acc	FAR	MAR
10.0.1.143	**92%**	6.1%	60%	89%	6.6%	**56%**	89%	5.8%	64%	88%	**5.2%**	68%	89%	5.4%	64%
10.0.1.144	90%	**1.6%**	**33.3%**	**94%**	2%	36.7%	91%	3.7%	47.4%	91%	2.8%	38.9%	94%	1.6%	33.3%
10.0.1.145	98%	1.8%	1.8%	98%	6.2%	2.3%	**99%**	**1.5%**	**1.4%**	96%	4.6%	3.2%	95%	4.6%	9.7%
10.0.1.153	**92%**	**1.5%**	67.4%	91%	4.6%	64.7%	91%	4.2%	73.4%	90%	3.6%	**61.9%**	91%	3.8%	81%
10.0.2.52	98%	0.6%	11.6%	**99%**	0.4%	11.6%	98%	0.4%	**9.3%**	93%	1.8%	39.5%	93%	**0**	44.2%
10.0.2.67	**66%**	**7.5%**	**60.4%**	65%	21%	63.3%	66%	17.1%	64.4%	41%	29.8%	84.2%	60%	21.5%	73.2%
10.0.2.191	**97%**	2.9%	**11.1%**	96%	1.6%	25%	96%	**1.2%**	28.2%	93%	1.8%	25%	95%	2.4%	19.4%
10.0.2.208	**98%**	14.7%	**7.7%**	98%	1.2%	8.3%	98%	**0**	12.8%	97%	1.6%	12.8%	98%	0.8%	10.3%
10.0.2.220	**97%**	0.8%	23.1%	93%	**0**	**21.7%**	96%	1.2%	34.7%	93%	0.76%	34.8%	97%	0.8%	30.4%
10.0.7.42	90%	**1.9%**	**10.2%**	**92%**	7.5%	10.2%	88%	11.3%	13.9%	79%	1.9%	25%	87%	8.5%	17.6%
10.0.7.123	**96%**	1.1%	**30.4%**	96%	**0**	100%	94%	3.0%	50%	94%	0.77%	50%	95%	1.5%	50%
10.0.9.207	97%	14.9%	5.1%	**98%**	0.5%	5.1%	98%	1.0%	3.8%	98%	0.49%	**2.5%**	71%	**0**	100%
10.0.9.208	85%	**0.8%**	11.7%	87%	15%	**10%**	**88%**	16.7%	13.8%	82%	33.3%	12.6%	85%	16.7%	15.3%
平均	**92.0%**	**4.3%**	**25.7%**	92.0%	5.1%	31.9%	92.0%	5.2%	32.1%	87.3%	6.8%	35.3%	88.5%	5.2%	42.2%

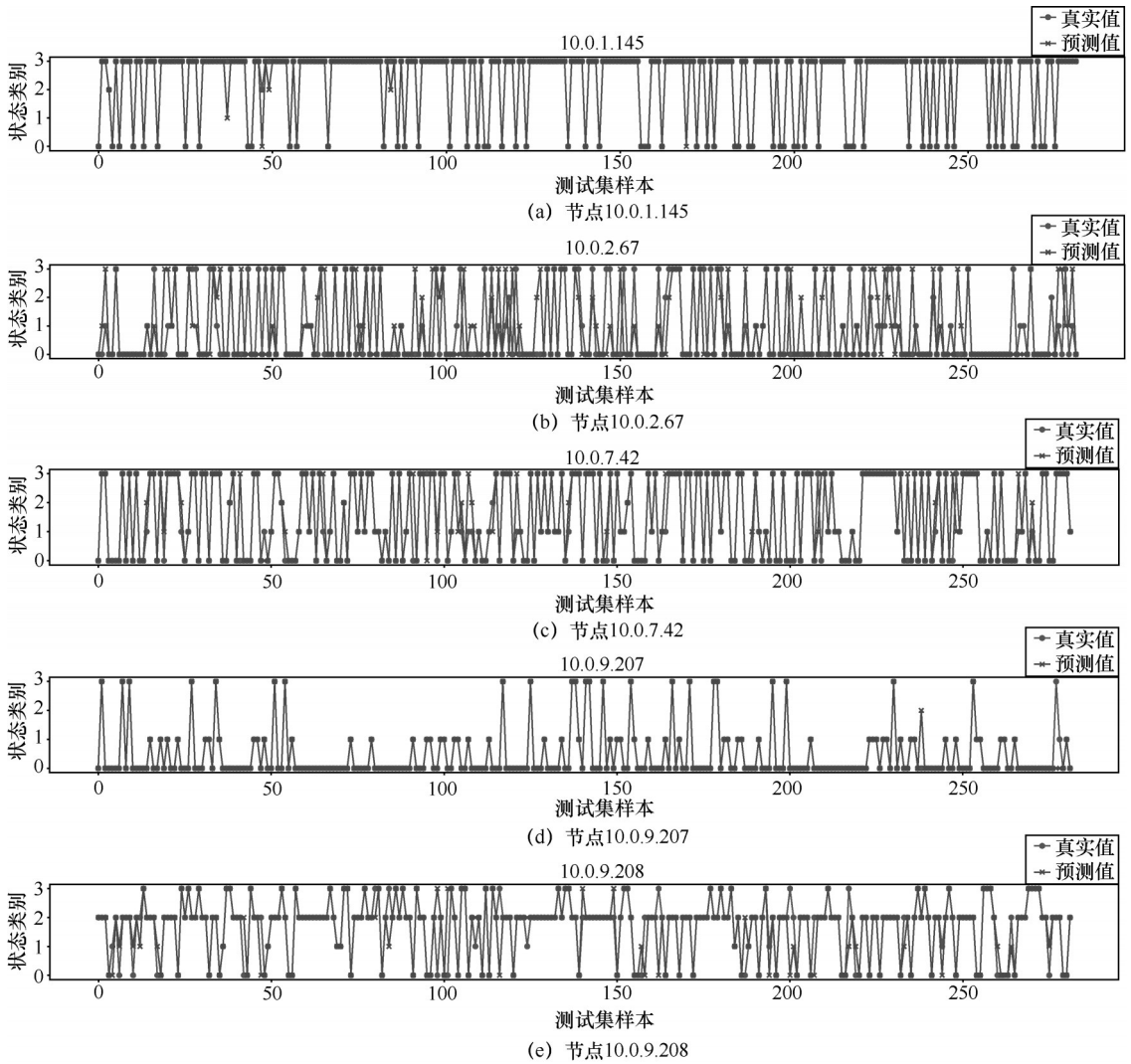

图4　部分节点在本文算法中的预测类别与真实类别分布情况

模型和其他经典的时间序列预测网络对比，本文设计的引入校正机制的预测模型在局部网络场景下具有更优的预测效果，其中平均预测准确率高达92%，虚警率仅为4.3%，漏警率为25.7%。总体来看，本文模型有不俗的预测效果，有助于减轻网络运维负担，减少网络故障所造成的损失。

未来的研究方向包括优化预测模型的算法和结构，增加数据集样本数，提高预测模型的准确性和效率，探索更多的日志数据处理和分析方法，以及将预测模型应用于更广泛的网络场景中。

参考文献：

[1] 樊志强, 刘日昇. 网络设备智能化管理的研究与应用[J]. 物联网技术, 2023, 13(8): 51-55.
FAN Z Q, LIU R S. Research and application of intelligent management of network equipment[J]. Internet of Things Technologies, 2023, 13(8): 51-55.

[2] SHUAN L H, FEI T Y, KING S W, et al. Network equipment failure prediction with big data analytics[J]. International Journal of Advances in Soft Computing & Its Applications, 2016, 8(3).

[3] LEE H. Framework and development of fault detection classification using IoT device and cloud environment[J]. Journal of Manufacturing Systems, 2017, 43: 257-270.

[4] CHOE H O, LEE M H. Artificial intelligence-based fault diagnosis and prediction for smart farm information and communication technology equipment[J]. Agriculture, 2023, 13(11): 2124.

[5] HE S L, ZHU J M, HE P J, et al. Experience report: system log analysis for anomaly detection[C]//Proceedings of the 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE). Piscataway: IEEE Press, 2016: 207-218.

[6] NAM S, HONG J, YOO J H, et al. Virtual machine failure prediction using log analysis[C]//Proceedings of the 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS). Piscataway: IEEE Press, 2021: 279-284.

[7] DU M, LI F F, ZHENG G N, et al. DeepLog: anomaly detection and diagnosis from system logs through deep learning[C]//Proceedings of the Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2017: 1285-1298.

[8] CHEN Z B, LIU J Y, GU W W, et al. Experience report: deep learning-based system log analysis for anomaly detection[EB]. 2021: 2107.05908.

[9] 贾统, 李影, 吴中海. 基于日志数据的分布式软件系统故障诊断综述[J]. 软件学报, 2020, 31(7): 1997-2018.
JIA T, LI Y, WU Z H. Survey of state-of-the-art log-based failure diagnosis[J]. Journal of Software, 2020, 31(7): 1997-2018.

[10] MASSARO A, KOSTADINOV D, SILVA A, et al. Predicting network hardware faults through layered treatment of alarms logs[J]. Entropy, 2023, 25(6): 917.

[11] 钟将, 时待吾, 王振华. 基于告警日志的网络故障预测[J]. 计算机应用, 2016, 36(S1): 49-53.
ZHONG J, SHI D W, WANG Z H. Network failure prediction based on alarm log[J]. Journal of Computer Applications, 2016, 36(S1): 49-53.

[12] TAN Z L, PAN P S. Network fault prediction based on CNN-LSTM hybrid neural network[C]//Proceedings of the 2019 International Conference on Communications, Information System and Computer Engineering (CISCE). Piscataway: IEEE Press, 2019: 486-490.

[13] 杨锋英, 刘会超. 基于Hadoop的在线网络日志分析系统研究[J]. 计算机应用与软件, 2014, 31(8): 311-316.
YANG F Y, LIU H C. Research on hadoop-based online network log analysis system[J]. Computer Applications and Software, 2014, 31(8): 311-316.

[14] MA J C, LIU Y, WAN H J, et al. Automatic parsing and utilization of system log features in log analysis: a survey[J]. Applied Sciences, 2023, 13(8): 4930.

[15] HAN S B, WU Q H, ZHANG H, et al. Log-based anomaly detection with robust feature extraction and online learning[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 2300-2311.

[16] LIM M H, LOU J G, ZHANG H Y, et al. Identifying recurrent and unknown performance issues[C]//Proceedings of the 2014 IEEE International Conference on Data Mining. Piscataway: IEEE Press, 2014: 320-329.

[17] 胡军伟, 秦奕青, 张伟. 正则表达式在Web信息抽取中的应用[J]. 北京信息科技大学学报(自然科学版), 2011, 26(6): 86-89.
HU J W, QIN Y Q, ZHANG W. Regular expression and its applications to web information extraction[J]. Journal of Beijing Information Science & Technology University, 2011, 26(6): 86-89.

[18] 毛远宏, 孙琛琛, 徐鲁豫, 等. 基于深度学习的时间序列预测

方法综述[J]. 微电子学与计算机, 2023, 40(4): 8-17.
MAO Y H, SUN C C, XU L Y, et al. A survey of time series forecasting methods based on deep learning[J]. Microelectronics & Computer, 2023, 40(4): 8-17.

[19] ZHOU H Y, ZHANG S H, PENG J Q, et al. Informer: beyond efficient transformer for long sequence time-series forecasting [J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(12): 11106-11115.

[20] CHE Z P, PURUSHOTHAM S, CHO K, et al. Recurrent neural networks for multivariate time series with missing values[J]. Scientific Reports, 2018, 8(1): 6085.

[21] XIE K J, LIU M B, LU W T, et al. Measurement-based prediction-correction online distributed optimal power flow algorithm for multi-phase active distribution networks[J]. Journal of Cleaner Production, 2022(362): 131935.

[22] ZHANG R T, MA X L, DING W P, et al. MAP-FCRNN: Multi-step ahead prediction model using forecasting correction and RNN model with memory functions[J]. Information Sciences, 2023(646): 119382.

[23] ORESHKIN B N, CARPOV D, CHAPADOS N, et al. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting[EB]. 2019: 1905.10437.

[作者简介]

王雨晞（1999– ），男，宁波大学信息科学与工程学院硕士生，主要研究方向为通信网智能运维。

叶庆卫（1970– ），男，宁波大学信息科学与工程学院教授、硕士生导师，主要研究方向为通信信号处理、信号传输与检测、分类器与机器学习等。

周鹏（1999– ），男，宁波大学信息科学与工程学院硕士生，主要研究方向为脑电信号处理。

李冰（2001– ），女，宁波大学信息科学与工程学院硕士生，主要研究方向为通信网智能运维。

王晓东（1970– ），男，宁波大学信息科学与工程学院教授、硕士生导师，主要研究方向为多媒体信号处理、图像处理等。