

一、文献信息：

作者：Chaoli Zhang, Zhiqiang Zhou, Yingying Zhang, Linxiao Yang, Kai He, Qingsong Wen, Liang Sun

论文题目：NETRCA: AN EFFECTIVE NETWORK FAULT CAUSE LOCALIZATION ALGORITHM

发表途径：ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

发表时间：23-27 May 2022

二、问题简介：

（一）研究问题：

在网络规模不断扩大，复杂性不断增加的情况下，如何自动并且智能地准确地定位网络中的故障，提升定位复杂 5G 网络中故障的准确性。对于本次挑战赛，具体来说，就是根据给定的数据和因果关系图，推断出每个不同的时间切片中，对特征 0 影响程度最大的原因。

我认为这在现实生活中有重大意义，利用相关数据定位故障，能更好的提升网络的运维成本，提升用户体验，在现阶段有重要意义。

（二）研究背景：

作者列举了文献[4-12]来简述了最新的科研成果，研究都是对网络中根本原因（root cause）的分析。其中包括时间序列分析和层次贝叶斯网络设计的分析系统，多种全局和局部可解释性的方法应用等。

（三）研究挑战：

主要是有三方面的挑战：1、随着网络深度的增加，根本原因节点的定位有难度；2、数据集的缺失，对于已有的故障数据缺少对应故障节点的标签；3、各个节点之间的相互关系比较复杂。

（四）研究方法：

此文将用 NetRCA 方法来解决上述问题。NetRCA 由特征工程、数据增强、模型集成三部分构成。

三、提出 NEetRCA 框架：

（一）框架简述：

该框架由三步构成：特征工程，数据增强，模型集成。

（二）特征工程：

每个样本中的时间戳数量不同，直接使用所有的时间戳来训练模型可能会导致偏差。作者基于从每个样本中提取的特征来训练模型。特征有四类，分别是时间特征、方向相关特征、归因特征和交互特征。

Date & Time	feature0	feature1	feature2	feature3	feature3	feature3	feature3	feature3
2020-09-23 23:56:28	756.3	24.03	3.92	19	0	33	1117	
2020-09-23 23:56:29	632.04	17.88	3.95	21	0	0	1226	
2020-09-23 23:56:30	701.49	16.66	3.99	5	0	0	1363	
2020-09-23 23:56:31	721.13	21.99	3.97	13	0	0	1120	
2020-09-23 23:56:32	863.06	22.22	3.97	13	0	0	1317	
2020-09-23 23:56:33	976.5	22.26	3.99	4	0	0	1432	
2020-09-23 23:56:34	563.78	20.38	3.93	22	0	0	939	
2020-09-23 23:56:35	774.3	19.66	3.97	13	0	0	1389	
2020-09-23 23:56:36	730.3	18.94	3.97	13	0	0	1330	
2020-09-23 23:56:37	786.06	19.09	3.99	4	0	0	1428	
2020-09-23 23:56:39	564.01	15.85	3.97	12	0	0	1381	
2020-09-23 23:56:40	550.76	15.51	3.97	13	0	0	1325	
2020-09-23 23:56:41	688.53	16.77	3.99	4	0	0	1425	
2020-09-23 23:56:42	510.77	17.81	3.93	23	0	0	952	
2020-09-23 23:56:43	762.26	19.73	3.97	12	0	0	1381	
2020-09-23 23:56:44	710.22	19.93	3.97	13	0	0	1317	
2020-09-23 23:56:45	806.73	19.01	3.99	4	0	0	1430	
2020-09-23 23:56:46	461.07	16.36	3.93	23	0	0	994	
2020-09-23 23:56:47	487.37	15.84	3.97	14	0	0	1250	
2020-09-23 23:56:48	516	13.44	3.98	10	0	0	1487	

图 1: 时间切片 1 中的训练数据

时间特征: 每个切片中的时间数据如图 1 所示, 时间特征可以是平均值、最小值、最大值、中位数、十分位数和偏度。时间序列的形状, 峰值的数量和变化的平均值等。

14	feature20_n, n=0,1,...,7	ID of 8 receiving directions	Continuous non-negative integer 0-31, arranged as 4*8 matrix: 24,25,26,27,28,29,30,31 16,17,18,19,20,21,22,23 8,9,10,11,12,13,14,15 0,1,2,3,4,5,6,7
----	--------------------------	------------------------------	---

图 2: 特征 20 的变量意义和变量属性

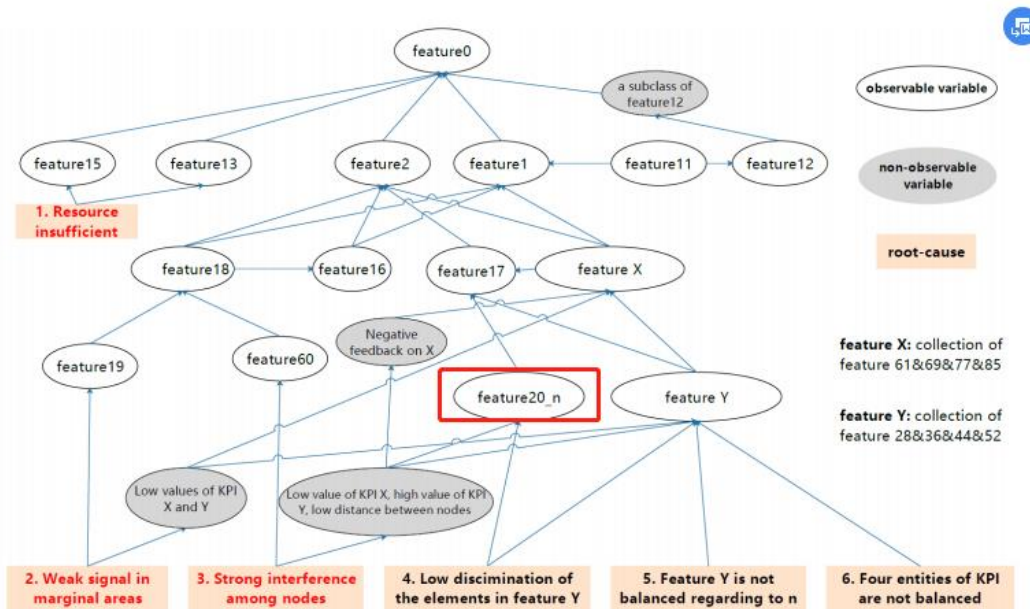


图 3: 因果关系图

方向相关特征: 特征 20 是检测原因是否为原因 2&3 的重要特征, 它是 8 个接收方向的 ID, 由图 2 和图 3 所知。通过他可以通过映射计算每对节点之间的欧几里得距离。得到 X

和 Y 的特征，即特征在 8 个方向的值。以及最终的统计特征。从每个时间切片样本的距离分布中总结为模型训练。

归因特征：根据因果图，可以推导出除特征 0 外的所有节点的归因特征。分析问题我们可知是根本原因最终导致特性 0 的值较低。因此是真正的根本原因及其相关特征对特性 0 影响大。因此，采用方法预测每个特征对我们的目标特征 0 的影响程度，具体使用改进的 sharply 方法，计算在后文给出。

15	feature28_n, n=0,1,...,7	Together as a set of feature Y, representing KPI Y; feature28/36/44/52 should be considered jointly regarding direction n.	feature28's strength in direction n, continuous negative value (8 directions in total)
16	feature36_n, n=0,1,...,7		feature36's strength in direction n, continuous negative value (8 directions in total)
17	feature44_n, n=0,1,...,7		feature44's strength in direction n, continuous negative value (8 directions in total)
18	feature52_n, n=0,1,...,7		Feature52's strength in direction n, continuous negative value (8 directions in total)
19	feature60		Continuous negative value
20	feature61_n, n=0,1,...,7	Together as a set of feature X, representing KPI X; Equal to the ratio of feature Y over some other factor; feature61/69/77/85 corresponds to feature 28/36/44/52 respectively.	feature61's strength ratio in direction n, continuous negative value (8 directions in total)
21	feature69_n, n=0,1,...,7		feature69's strength ratio in direction n, continuous negative value (8 directions in total)
22	feature77_n, n=0,1,...,7		feature77's strength ratio in direction n, continuous negative value (8 directions in total)
23	feature85_n, n=0,1,...,7		feature85's strength ratio in direction n, continuous negative value (8 directions in total)

图 4：交互特征 X 和 Y 的变量意义和变量属性

交互特征：X 和 Y 的二阶交互特征生成，如图 4 所示。将特征 X 赋值为特征 Y 与某些未知因素的比值。通过在特征 Y 的基础上生成特征 X 来衡量这些未知因素的影响。根据对应关系将 X 和 Y 中的特征分对。每一对，计算 X 与 Y 的比值，将这些统计数据作为时间特征。

(三) 数据增强：

1. 多元时间序列相似性

测量不同长度的多元时间序列之间的相似度，采用 Eros 算法来计算相似度。利用主成分和基于特征向量计算相似性，扩展 Eros。对 A 和 B 的协方差矩阵应用 SVD，得到两个右特征向量矩阵。那么 A 和 B 的相似性为公式 1，它是奇异值分解后的右向量，然后取它的特征值，代入公式进行计算。

$$Eros(A, B, \omega) = \sum_{i=1} \omega_i | \langle a_i, b_i \rangle | \quad (1)$$

2. 数据/标签增强

sample_index	root-cause(s)	
670	rootcause3	
673	rootcause3	
675	rootcause3	
684	rootcause3	
690	rootcause3	
694	rootcause3	
698	rootcause3	

图 5：训练集标记

比赛给的训练数据如图 5 所示。有标签的数据对训练模型很重要，但是标记的数据有限，训练数据中超过一半的训练数据没有标记，草率将其删除会错过有价值的信息。因此，作者先使用 Eros，测量任何两个训练样本之间的相似性（这两个样本都是多元时间序列）。在确定样本相似关系后，就可以选择与标记数据相似性较高的未标记数据进行标记，并将其标签进行标记。对每种类型的根本原因分别执行此过程，以提高计算效率。

此外，如果两个时间戳相近但是有一个没有被标记，那么这两个相似的时间切片对应的主要原因应该是相近的，就可以扩散，然后把没有标记的标记一下。这还能解决漏标记的问题。

最终，作者采取了以上两种方法结果的并集，对未标记的训练时间切片进行标记。

(三) 集成模型：

采用 XGBoost 获得初始结果，结合规则集学习、归因模型和图算法对结果进行细化。

1. XGBoost 作为基础模型，进行初步的分类。

因为是第一次接触这个框架，简单了解了一下，发现 python 中有直接可以调用的库，使用自己的简单数据进行测试，如图 6，学习 XGBoost 的算法结构。并查阅相关资料，发现这是一个常用的框架，在众多算法比赛中常用。

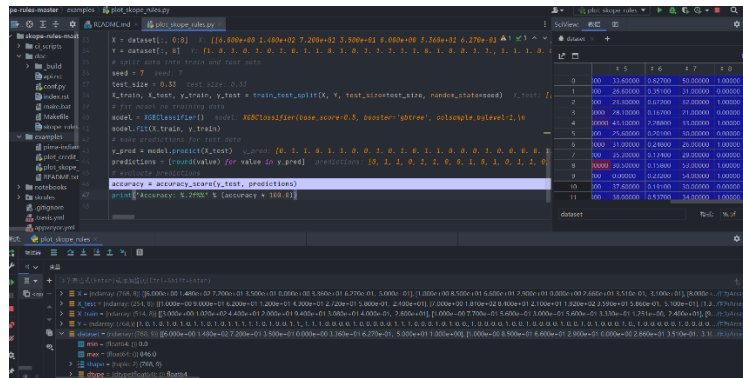
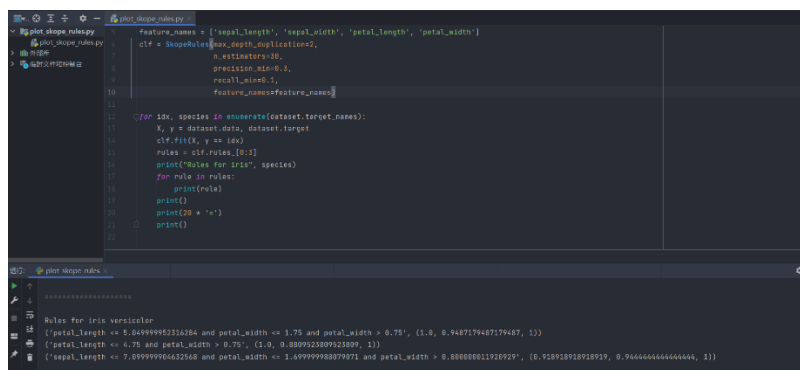


图 6：简单体验 XGBoost

2. 规则集学习：

作者分析构建一个强大的分类器的一个挑战是特征交互，当某些特征的值相互影响时，就会发生特征交互。反观这题，本质就是分类过程，是将已有的数据进行分类，将其划分到各个原因节点，在划分过程中，主要还是考虑到各个特征的相关性。此外，规则的可解释性使我们能够理解特征和目标之间的关系，并有助于检测导入特征。

作者在学习规则时，使用了 python 的包 Skope-rules package。因为是第一次了解，为了方便理解，我用一个简单实例进行实验。给定花的三个特征以及花的种类，该方法将返回花的判断标准，以及精确程度。而应用到该挑战赛中，就是返回判断故障节点的标准，但考虑到节点数多，原因数少，如果不用其他模型，结果误差将较大。



3. 预测归因模型:

当节点之间的相互依赖关系可用时,就可以估计这些特性的重要性。可以帮助我们确定根本原因,文中特征重要性估计是基于 shapley 值,公式如(2)所示。

$$\varphi_i(v) = \sum_{S \in N} \frac{[(|S| - 1)!(n - |S|)!]}{n} \times [v(S) - v(S \setminus \{i\})] \quad (2)$$

为方便理解,我找了网上的一道例题:

<https://blog.csdn.net/luzaijiaoxia0618/article/details/104527561>

但是,作者指出如果要用 shapley 算,有两个要求,1. 必须要求所有特征都算好 2. 太复杂,浪费时间,然后就用式子(3)近似等于(2),其中内部节点和特征 0 节点之间的关系函数 f (对应关系值)是通过训练 XGboost 模型来估计,最终发现其效果不错。

$$\varphi(i) \approx |f(\mathbf{x}_S) - f([\mathbf{x}_{S \setminus \{i\}}, \bar{x}_i])| \quad (3)$$

在确定了特征的重要性之后,他们通过简单地将其与预定义的阈值进行比较来确定根本原因。

4. 图算法:

为了进一步利用所提供的因果图,作者设计了一种基于单变量时间的专门图算法。先用皮尔逊相关系数评估特征 i 和特征 0 的相似度。计算相似度前要对缺失的时间对应数据进行插值。该相似度表示两个特征如何随着时间共同变化。体现两个特征基于时间的相关性。但相关性不代表因果关系,单用会造成假阳性。因此,将相似关系和因果关系图结合起来使用,基于相似性得分对因果图进行随机游走。从特征 0 开始,随机选取邻居。游走概率与通过归一化相似度评分计算出的边缘权值成正比。并且如果对根本原因旁边的特征的访问越多,根本原因就越有可能是特征 0 的真正故障节点。

四、实验和讨论:

在该部分总结并讨论了 NetRCA 在 ICASSP2022AI0ps 挑战赛数据集上的效果。

(一) 简介数据集和评估方法:

特征数据集包含 2984 个样本,23 个可观测变量。在 2984 个样本中,只有约 45% 的样本被标记为根本原因故障,而其他样本仍未被标记,说明标签稀缺且不全面。对于评价指标,采用挑战提供的归一化最终分数,每个真阳性增加 1 分,每个假阳性扣除 1 分。最终的分数由测试样本的数量归一化,因此最终的最高分数为 1。

(二) 实验和配置:

分别对根本原因 1、根本原因 2 和根本原因 3 训练了三种二元分类模型。基于从特征工程和数据增强中导出的数据。对于根本原因 1,我们主要使用来自特征 0、13、15 的信息以及它们之间的交互作用。因为实验发现除去特征 X、Y 啥的效果更好。对于根本原因 2,主要用 19 和 20 对于根本原因 3,主要用 X, Y。

(三) 模型可解释性:

通过应用 NetRCA 提出的可解释模型,对模型预测正确答案的有更深入的理解

(四) 性能比较:

Models	Root1 acc	Root2 acc	Root3 acc	Final Score
XGB	0.9828	0.97849	0.9957	0.78139
XGB+FE	0.9957	0.97849	0.9914	0.86611
XGB+FE+Graph	0.9957	0.97849	0.9914	0.87917
Proposed NetRCA	0.9957	0.98495	0.9914	0.91778

图 7: 性能比较

具体性能比较如上表所示，可以发现 NetRCA 模型的优越性。

五、结论：

本文提出的一种新的 NetRCA 算法的确能较好的定位网络故障的根本原因。

六、启发思考：

本篇论文虽然页数不是很多，但值得我学习的模型还是很多。因为它是比赛的论文，通过对赛题的阅读，以及对作者思路的分析，能进一步是我对论文有深入了解。其中提到的各个模型和算法，有的也是我第一次遇到，进一步拓宽了我的思路，有所启发。