



(21) 申请号 202411025507.2

(22) 申请日 2024.07.30

(65) 同一申请的已公布的文献号

申请公布号 CN 118550757 A

(43) 申请公布日 2024.08.27

(73) 专利权人 安徽思高智能科技有限公司

地址 230088 安徽省合肥市高新区望江西

路900号中安创谷科技园A1栋408

(72) 发明人 张竞超 余鳌 张泽锟 张流潇潇
王健(74) 专利代理机构 武汉知产时代知识产权代理
有限公司 42238

专利代理师 朱信

(51) Int. Cl.

G06F 11/07 (2006.01)

H04L 41/0631 (2022.01)

G06F 18/25 (2023.01)

G06F 18/2135 (2023.01)

G06F 18/15 (2023.01)

G06N 3/0464 (2023.01)

G06N 3/0455 (2023.01)

G06N 3/0442 (2023.01)

G06N 3/084 (2023.01)

G06N 3/048 (2023.01)

G06F 123/02 (2023.01)

(56) 对比文件

CN 113568819 A, 2021.10.29

CN 116383096 A, 2023.07.04

审查员 史雪飞

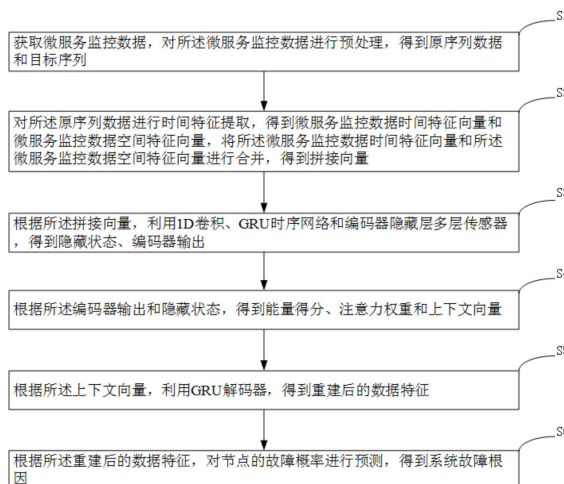
权利要求书3页 说明书12页 附图2页

(54) 发明名称

面向缺失数据的微服务系统根因定位方法、
介质及设备

(57) 摘要

本发明公开了一种面向缺失数据的微服务系统根因定位方法、介质及设备,涉及智能运维领域,面向缺失数据的微服务系统根因定位方法主要包括:基于卷积网络对输入数据构建时间和空间维度的向量特征并得到拼接向量,通过GRU获取拼接向量的隐藏层状态,将编码器输出进行注意力对齐,得到解码器的注意力上下文向量;最后利用GRU进行解码,得到重建后的数据特征,基于重建后的数据特征,对节点的故障概率进行预测,最终实现故障根因排序。实施本发明提供的面向缺失数据的微服务系统根因定位方法、介质及设备,能实现故障根因排序,提高维护微服务系统的稳定性和可观测性。



1. 一种面向缺失数据的微服务系统根因定位方法,其特征在于,包括以下步骤:

S1: 获取微服务监控数据,对所述微服务监控数据进行预处理,得到原序列数据和目标序列;

S2: 对所述原序列数据进行时间特征提取,得到微服务监控数据时间特征向量和微服务监控数据空间特征向量,将所述微服务监控数据时间特征向量和所述微服务监控数据空间特征向量进行合并,得到拼接向量;步骤S2具体包括以下步骤:

S21: 对所述原序列数据进行时间特征提取,得到微服务监控数据时间特征向量,如公式:

$$\mathbf{X}'_{time}(b, h, t, n) = \text{ReLU}\left(\sum_{f=0}^{F-1} \sum_{m=0}^{k_t-1} \mathbf{X}_{time}(b, f, t+m, n) \cdot W_t(h, f, m) + b_t(h)\right),$$

其中, $\mathbf{X}'_{time}(b, h, t, n)$ 为微服务监控数据时间特征向量,形状为 $[B, H_t, T, N]$, B 为数据批处理大小, H_t 表示时间维度卷积层输出的特征数量, T 为时间步数, N 为微服务系统节点数量, $\mathbf{X}_{time}()$ 是经过形状调整的输入数据,形状为 $[B, F, T, N]$, F 为每个节点监控的指标数量; b 为批次索引,表示当前处理的是哪一个批次中的样本; f 为特征通道索引,表示当前处理的是哪一个特征; h 为输出通道索引,对应于卷积层输出特征图中的一个特定“深度”或“通道”; t 为时间步索引,表示经过卷积操作后的时间维度上的位置; n 为节点索引,表示当前处理的是哪一个节点; $\text{ReLU}()$ 为激活函数; k_t 为时间维度上的卷积核大小; m 为卷积核的局部索引; $W_t()$ 为时间维度卷积核的权重参数; $b_t()$ 为时间维度卷积操作的偏置项;

S22: 对所述原序列数据进行空间特征提取,得到微服务监控数据空间特征向量,如公式:

$$\mathbf{X}'_{node}(b, h, f, t) = \text{ReLU}\left(\sum_{n=0}^{N-1} \sum_{m=0}^{k_n-1} \mathbf{X}_{node}(b, n, f+m, t) \cdot W_n(h, n, m) + b_n(h)\right),$$

其中, $\mathbf{X}'_{node}(b, h, f, t)$ 为微服务监控数据空间特征向量,形状为 $[B, H_n, T, N]$, B 为数据批处理大小, H_n 表示空间维度卷积层输出的特征数量, T 为时间步数, N 为微服务系统节点数量; b 为批次索引, h 为输出通道索引, f 为特征通道索引, t 为时间步索引; $\text{ReLU}()$ 为激活函数; k_n 为在空间维度上卷积核大小; $\mathbf{X}_{node}()$ 是经过形状调整的输入数据,形状为 $[B, N, F, T]$; m 为卷积核的局部索引; $W_n()$ 为空间维度卷积核的权重参数; $b_n()$ 为空间维度卷积操作的偏置项;

S23: 将所述微服务监控数据时间特征向量和所述微服务监控数据空间特征向量进行合并,得到拼接向量,如公式:

$$\mathbf{X}_{combined} = \text{concat}(\mathbf{X}'_{time}, \mathbf{X}'_{node}, \text{dim} = 1),$$

其中, $\mathbf{X}_{combined}$ 是拼接向量, $\text{concat}(\cdot, \text{dim}=1)$ 表示在第二维上拼接两个向量, $\mathbf{X}'_{time}$ 为微服务监控数据时间特征向量, $\mathbf{X}'_{node}$ 为微服务监控数据空间特征向量;

S3: 根据所述拼接向量,利用1D卷积、GRU时序网络和编码器隐藏层多层传感器,得到隐藏状态、编码器输出;

- S4:根据所述编码器输出和隐藏状态,得到能量得分、注意力权重和上下文向量;
 S5:根据所述上下文向量,利用GRU解码器,得到重建后的数据特征;
 S6:根据所述重建后的数据特征,对节点的故障概率进行预测,得到系统故障根因。

2.根据权利要求1所述的面向缺失数据的微服务系统根因定位方法,其特征在于,步骤S1包括以下步骤:

- S11:获取微服务监控数据,所述微服务监控数据包括CPU利用率、内存使用量、磁盘IO;
 S12:对所述微服务监控数据进行去异常值、标准化和插值填补,得到原序列数据和目标序列。

3.根据权利要求1所述的面向缺失数据的微服务系统根因定位方法,其特征在于,步骤S3包括以下步骤:

S31:利用1D卷积对所述拼接向量进行重塑操作,得到重塑序列向量,如公式:

$$\mathbf{X}_{seq}[b, h', t'] = \text{ReLU} \left(\sum_{h=0}^{H_t + H_n - 1} \sum_{m=0}^{k-1} \mathbf{X}_{combined}[b, h, s \cdot t' + m] \cdot W[h', h, m] + b[h'] \right),$$

其中, $\mathbf{X}_{seq}[b, h', t']$ 为重塑序列向量, b 为批次索引, h' 为1D卷积的输出通道索引; t' 为1D卷积的时间步索引, $\text{ReLU}()$ 为激活函数; H_t 为时间维度卷积层输出的特征数量, H_n 为空间维度卷积层输出的特征数量, k 为1D卷积的卷积核大小, s 为步长, h 为输出通道索引, m 为卷积核的局部索引, $W[h', h, m]$ 为1D卷积层的权重, $b[h']$ 为1D卷积操作的偏置项;

S32:根据重塑序列向量,利用GRU时序网络,得到隐藏状态,如公式:

$$\begin{aligned} h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t, \\ z_t &= \sigma(W_z[h_{t-1}, \mathbf{X}_{seq,t}] + b_z), \\ r_t &= \sigma(W_r[h_{t-1}, \mathbf{X}_{seq,t}] + b_r), \\ \tilde{h}_t &= \tanh(W_h[r_t \odot h_{t-1}, \mathbf{X}_{seq,t}] + b_h), \end{aligned}$$

其中, h_t 为隐藏状态, z_t 为更新门, h_{t-1} 表示前一时间步的隐藏状态, $\odot$ 表示元素乘法; $\tilde{h}_t$ 为候选隐藏状态, σ 表示sigmoid函数, 确保门控制信号的值在0和1之间; W_z, W_r, W_h 为权重矩阵, $\mathbf{X}_{seq,t}$ 表示在时间步 t 的输入特征向量; b_z, b_r, b_h 为偏置项; r_t 为重置门, $\tanh$ 提供非线性激活, 用于计算候选隐藏状态;

S33:根据所述隐藏状态,利用编码器隐藏层多层传感器,得到编码器输出。

4.根据权利要求1所述的面向缺失数据的微服务系统根因定位方法,其特征在于,步骤S4包括以下步骤:

S41:根据所述编码器输出和隐藏状态,得到能量得分,如公式:

$$\mathbf{E} = \tanh(W_{attn} \cdot [h_{t-1}'; O_{enc}] + b_{attn}),$$

其中, $\mathbf{E}$ 为能量得分, $\tanh$ 为非线性激活, W_{attn} 为权重矩阵, h_{t-1}' 为解码器前一时刻隐藏状态, O_{enc} 为编码器输出, b_{attn} 是偏置项;

S42:根据所述能量得分,得到注意力权重,如公式:

$$\mathbf{A} = \text{softmax}(v \cdot \mathbf{E}, \text{dim} = 1),$$

其中, $\mathbf{A}$ 为注意力权重, $\text{softmax}()$ 为归一化指数函数, v 是可学习参数, $\mathbf{E}$ 为能量得分, $\text{softmax}(\cdot, \text{dim}=1)$ 表示在第二维上对两个参数进行运算;

S43: 根据所述注意力权重和所述编码器输出, 得到上下文向量, 如公式:

$$\mathbf{C}_t = \sum_{i=1}^T A_{ti} \cdot O_{enc,i}$$

其中, $\mathbf{C}_t$ 为上下文向量, T 为时间步数, A_{ti} 是注意力权重中的第 i 个元素, 代表第 t 个解码步骤对第 i 个编码器输出的关注程度; $O_{enc,i}$ 是编码器输出中的第 i 个时间步的输出。

5. 根据权利要求1所述的面向缺失数据的微服务系统根因定位方法, 其特征在于, 步骤S5具体包括:

根据步骤S4中所述上下文向量, 利用GRU解码器, 得到重建后的数据特征, 如公式:

$$\mathbf{H}_M, h_t = \text{RNN}(\text{Concat}(\text{input}_t, \mathbf{C}_t), h_{t-1}'),$$

其中, $\mathbf{H}_M$ 为解码器输出, 表示对缺失数据重建后得到的序列特征; h_t 是当前解码器隐藏状态; $\text{RNN}()$ 表示循环神经网络; $\text{Concat}()$ 为连接函数, input_t 是当前时间步的输入, $\mathbf{C}_t$ 是上下文向量, h_{t-1}' 是前一时间步的解码器隐藏状态, 通过比较目标序列和重建序列 $\mathbf{H}_M$ 的差值计算损失, 使用均方误差MSE作为重建任务的损失函数, 基于损失函数的结果通过反向传播算法计算模型参数的梯度, 并使用Adam优化器更新模型的权重以减少预测和实际值之间的差异。

6. 根据权利要求1所述的面向缺失数据的微服务系统根因定位方法, 其特征在于, 步骤S6具体包括:

根据所述重建后的数据特征, 对节点的故障概率进行预测, 得到每个节点的故障概率, 根据每个节点的故障概率, 得到系统故障根因, 如公式:

$$\text{Faults} = \text{argsort}\left(\frac{\exp[(\mathbf{W}\mathbf{H}_G + b_f)_i]}{\sum_{j=1}^N \exp[(\mathbf{W}\mathbf{H}_G + b_f)_j]}\right)[-5:],$$

其中, Faults 为节点的故障概率, $\text{argsort}()$ 表示将元素从小到大排列, 并提取元素对应的索引号; $\exp$ 为自然指数函数, N 是微服务系统节点个数, i 是全连接输出向量的第 i 个值, $\mathbf{W}$ 是全连接层的权重矩阵, $\mathbf{H}_G$ 为 $\mathbf{H}_M$ 经过PCA降维后的一维向量, b_f 是偏置向量。

7. 一种计算机可读存储介质, 其上存储有计算机程序, 其特征在于, 该计算机程序被处理器执行时实现如权利要求1-6任一所述的面向缺失数据的微服务系统根因定位方法的步骤。

8. 一种计算机设备, 包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序, 其特征在于, 所述处理器执行所述计算机程序时实现如权利要求1-6任一所述的面向缺失数据的微服务系统根因定位方法的步骤。

面向缺失数据的微服务系统根因定位方法、介质及设备

技术领域

[0001] 本发明涉及智能运维领域,更具体地说,涉及一种面向缺失数据的微服务系统根因定位方法、介质及设备。

背景技术

[0002] 微服务系统指标对于维护系统健康状态和进行故障诊断至关重要。然而,受到云原生架构特性影响,部分监控指标数据呈现出缺失的状态。这种不完备性可能会严重影响后续的数据分析和根因定位的准确性。

[0003] 在现代微服务架构中,主流的指标收集方法采用的是边车(Sidecar)设计模式。具体来说,这涉及在业务进程旁边运行一个边车进程,该进程负责实时监控并收集业务进程的监控指标数据。收集到的数据随后被整合和上报,最终存储在特定的数据库中供后续分析使用。这种设计模式允许监控逻辑与业务逻辑分离,从而不干扰主业务的运行。然而,指标数据收集过程本身对计算、存储和网络资源的需求不可忽视。在资源有限的环境下,尤其是当机器资源紧张时,边车进程很可能会遭遇工作中断导致数据收集任务的不连续性,导致部分检测数据的缺失,这给传统的微服务根因定位任务带来了新的挑战。

发明内容

[0004] 本发明的目的在于,提供一种面向缺失数据的微服务系统根因定位方法、介质及设备,能实现故障根因排序,提高维护微服务系统的稳定性和可观测性。

[0005] 本发明提供一种面向缺失数据的微服务系统根因定位方法,包括以下步骤:

[0006] S1:获取微服务监控数据,对微服务监控数据进行预处理,得到原序列数据和目标序列;

[0007] S2:对原序列数据进行时间特征提取,得到微服务监控数据时间特征向量和微服务监控数据空间特征向量,将微服务监控数据时间特征向量和微服务监控数据空间特征向量进行合并,得到拼接向量;

[0008] S3:根据拼接向量,利用1D卷积、GRU时序网络和编码器隐藏层多层传感器,得到隐藏状态、编码器输出;

[0009] S4:根据编码器输出和隐藏状态,得到能量得分、注意力权重和上下文向量;

[0010] S5:根据上下文向量,利用GRU解码器,得到重建后的数据特征;

[0011] S6:根据重建后的数据特征,对节点的故障概率进行预测,得到系统故障根因。

[0012] 进一步地,上述面向缺失数据的微服务系统根因定位方法的步骤S1包括以下步骤:

[0013] S11:获取微服务监控数据,微服务监控数据包括CPU利用率、内存使用量、磁盘IO;

[0014] S12:对微服务监控数据进行去异常值、标准化和插值填补,得到原序列数据和目标序列。

[0015] 进一步地,上述面向缺失数据的微服务系统根因定位方法的步骤S2包括以下步

骤:

[0016] S21:对原序列数据进行时间特征提取,得到微服务监控数据时间特征向量,如公式:

$$[0017] \quad \mathbf{X}'_{time}(b, h, t, n) =$$

$$[0018] \quad \text{ReLU}\left(\sum_{f=0}^{F-1} \sum_{m=0}^{k_t-1} \mathbf{X}_{time}(b, f, t+m, n) \cdot W_t(h, f, m) + b_t(h)\right),$$

[0019] 其中, $\mathbf{X}'_{time}(b, h, t, n)$ 为微服务监控数据时间特征向量, 形状为 $[B, H_t, T, N]$, B 为数据批处理大小, H_t 表示时间维度卷积层输出的特征数量, T 为时间步数, N 为微服务系统节点数量, $\mathbf{X}_{time}()$ 是经过形状调整的输入数据, 形状为 $[B, F, T, N]$, F 为每个节点监控的指标数量; b 为批次索引, 表示当前处理的是哪一个批次中的样本; f 为特征通道索引, 表示当前处理的是哪一个特征; h 为输出通道索引, 对应于卷积层输出特征图中的一个特定“深度”或“通道”; t 为时间步索引, 表示经过卷积操作后的时间维度上的位置; n 为节点索引, 表示当前处理的是哪一个节点; $\text{ReLU}()$ 为激活函数; k_t 为时间维度上的卷积核大小; m 为卷积核的局部索引; $W_t()$ 为时间维度卷积核的权重参数; $b_t()$ 为时间维度卷积操作的偏置项;

[0020] S22:对原序列数据进行空间特征提取,得到微服务监控数据空间特征向量,如公式:

$$[0021] \quad \mathbf{X}'_{node}(b, h, f, t) = \text{ReLU}\left(\sum_{n=0}^{N-1} \sum_{m=0}^{k_n-1} \mathbf{X}_{node}(b, n, f+m, t) \cdot W_n(h, n, m) + b_n(h)\right),$$

[0022] 其中, $\mathbf{X}'_{node}(b, h, f, t)$ 为微服务监控数据空间特征向量, 形状为 $[B, H_n, T, N]$, B 为数据批处理大小, H_n 表示空间维度卷积层输出的特征数量, T 为时间步数, N 为微服务系统节点数量; b 为批次索引, h 为输出通道索引, f 为特征通道索引, t 为时间步索引; $\text{ReLU}()$ 为激活函数; k_n 为在空间维度上卷积核大小; $\mathbf{X}_{node}()$ 是经过形状调整的输入数据, 形状为 $[B, N, F, T]$; m 为卷积核的局部索引; $W_n()$ 为空间维度卷积核的权重参数; $b_n()$ 为空间维度卷积操作的偏置项;

[0023] S23:将微服务监控数据时间特征向量和微服务监控数据空间特征向量进行合并,得到拼接向量,如公式:

$$[0024] \quad \mathbf{X}_{combined} = \text{concat}(\mathbf{X}'_{time}, \mathbf{X}'_{node}, \text{dim}=1),$$

[0025] 其中, $\mathbf{X}_{combined}$ 是拼接向量, $\text{concat}(\cdot, \text{dim}=1)$ 表示在第二维上拼接两个向量, $\mathbf{X}'_{time}$ 为微服务监控数据时间特征向量, $\mathbf{X}'_{node}$ 为微服务监控数据空间特征向量。

[0026] 进一步地,上述面向缺失数据的微服务系统根因定位方法的步骤S3包括以下步骤:

[0027] S31:利用1D卷积对拼接向量进行重塑操作,得到重塑序列向量,如公式:

$$[0028] \quad \mathbf{X}_{seq}[b, h', t'] = \text{ReLU}\left(\sum_{h=0}^{H_t+H_n-1} \sum_{m=0}^{k-1} \mathbf{X}_{combined}[b, h, s \cdot t' + m] \cdot W[h', h, m] + b[h']\right),$$

[0029] 其中, $\mathbf{X}_{seq}[b, h', t']$ 为重塑序列向量, b 为批次索引, h' 为1D卷积的输出通道索引; t' 为1D卷积的时间步索引, $\text{ReLU}()$ 为激活函数; H_t 为时间维度卷积层输出的特征数量, H_n 为

空间维度卷积层输出的特征数量, k 为1D卷积的卷积核大小, s 为步长, h 为输出通道索引, m 为卷积核的局部索引, $W[h', h, m]$ 为1D卷积层的权重, $b[h']$ 为1D卷积操作的偏置项;

[0030] S32:根据重塑序列向量,利用GRU时序网络,得到隐藏状态,如公式:

$$[0031] \quad h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t$$

$$[0032] \quad z_t = \sigma(W_z[h_{t-1}, \mathbf{X}_{seq,t}] + b_z)$$

$$[0033] \quad r_t = \sigma(W_r[h_{t-1}, \mathbf{X}_{seq,t}] + b_r)$$

$$[0034] \quad \tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, \mathbf{X}_{seq,t}] + b_h)$$

[0035] 其中, h_t 为隐藏状态, z_t 为更新门, h_{t-1} 表示前一时间步的隐藏状态, $\odot$ 表示元素乘法; $\tilde{h}_t$ 为候选隐藏状态, σ 表示sigmoid函数, 确保门控制信号的值在0和1之间; W_z, W_r, W_h 为权重矩阵, $\mathbf{X}_{seq,t}$ 表示在时间步 t 的输入特征向量; b_z, b_r, b_h 为偏置项; r_t 为重置门, $\tanh$ 提供非线性激活, 用于计算候选隐藏状态;

[0036] S33:根据隐藏状态,利用编码器隐藏层多层传感器,得到编码器输出。

[0037] 进一步地,上述面向缺失数据的微服务系统根因定位方法的步骤S4包括以下步骤:

[0038] S41:根据编码器输出和隐藏状态,得到能量得分,如公式:

$$[0039] \quad \mathbf{E} = \tanh(W_{attn} \cdot [h_{t-1}'; O_{enc}] + b_{attn})$$

[0040] 其中, E 为能量得分, $\tanh$ 为非线性激活, W_{attn} 为权重矩阵, h_{t-1}' 为解码器前一时间步隐藏状态, O_{enc} 为编码器输出, b_{attn} 是偏置项;

[0041] S42:根据能量得分,得到注意力权重,如公式:

$$[0042] \quad \mathbf{A} = \text{softmax}(v \cdot \mathbf{E}, \dim = 1)$$

[0043] 其中, A 为注意力权重, $\text{softmax}()$ 为归一化指数函数, v 是可学习参数, E 为能量得分, $\text{softmax}(\cdot, \dim=1)$ 表示在第二维上对两个参数进行运算;

[0044] S43:根据注意力权重和编码器输出,得到上下文向量,如公式:

$$[0045] \quad \mathbf{C}_t = \sum_{i=1}^T A_{ti} \cdot O_{enc,i}$$

[0046] 其中, C_t 为上下文向量, T 为时间步数, A_{ti} 是注意力权重中的第 i 个元素, 代表第 t 个解码步骤对第 i 个编码器输出的关注程度; $O_{enc,i}$ 是编码器输出中的第 i 个时间步的输出。

[0047] 进一步地,上述面向缺失数据的微服务系统根因定位方法的步骤S5具体包括:

[0048] 根据步骤S4中上下文向量,利用GRU解码器,得到重建后的数据特征,如公式:

$$[0049] \quad \mathbf{H}_M, h_t = \text{RNN}(\text{Concat}(input_t, C_t), h_{t-1}')$$

[0050] 其中, H_M 为解码器输出, 表示对缺失数据重建后得到的序列特征; h_t 是当前解码器隐藏状态; $\text{RNN}()$ 表示循环神经网络; $\text{Concat}()$ 为连接函数, $input_t$ 是当前时间步的输入, C_t 是上下文向量, h_{t-1}' 是前一时间步的解码器隐藏状态, 通过比较目标序列和重建序列 H_M 的

差值计算损失,使用均方误差(MSE)作为重建任务的损失函数,基于损失函数的结果通过反向传播算法计算模型参数的梯度,并使用Adam优化器更新模型的权重以减少预测和实际值之间的差异。

[0051] 进一步地,上述面向缺失数据的微服务系统根因定位方法的步骤S6具体包括:

[0052] 根据重建后的数据特征,对节点的故障概率进行预测,得到每个节点的故障概率,根据每个节点的故障概率,得到系统故障根因,如公式:

$$[0053] \quad Faults = \text{argsort}\left(\frac{\exp[(\mathbf{W}\mathbf{H}_G + \mathbf{b}_f)_i]}{\sum_{j=1}^N \exp[(\mathbf{W}\mathbf{H}_G + \mathbf{b}_f)_j]}\right)[-5:]$$

[0054] 其中,Faults为节点的故障概率,argsort()表示将元素从小到大排列,并提取元素对应的索引号;exp为自然指数函数,N是微服务系统节点个数,i是全连接输出向量的第i个值,W是全连接层的权重矩阵, $\mathbf{H}_G$ 为 $\mathbf{H}_M$ 经过PCA降维后的一维向量, $\mathbf{b}_f$ 是偏置向量。

[0055] 本发明还提供计算机可读存储介质,其上存储有计算机程序,该计算机程序被处理器执行时实现上述面向缺失数据的微服务系统根因定位方法的步骤。

[0056] 本发明还提供计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,该处理器执行该程序时实现上述的面向缺失数据的微服务系统根因定位方法的步骤。

[0057] 实施本发明提供的面向缺失数据的微服务系统根因定位方法、介质及设备,具有以下有益效果:

[0058] 本发明基于卷积网络对输入数据构建时间和空间维度的向量特征并得到拼接向量,通过门控循环单元(gated recurrent unit,GRU)时序网络编码器获取拼接向量的隐藏层状态,并输入到多层感知器(Multilayer Perceptron,MLP)得到编码器输出;将编码器输出进行注意力对齐,得到解码器的注意力上下文向量;从而将原始的低质量的微服务监控数据在时间和空间上在不损失原有特征的前提下进行补齐,最后利用门控循环单元(gated recurrent unit,GRU)解码器对注意力上下文向量进行解码,得到重建后的数据特征,基于重建后的数据特征,对节点的故障概率进行预测,最终实现故障根因排序,提高维护微服务系统的稳定性和可观测性。

附图说明

[0059] 图1是本发明提供的面向缺失数据的微服务系统根因定位方法的流程图;

[0060] 图2是本发明提供的面向缺失数据的微服务系统根因定位方法整体工作过程示意图;

[0061] 图3是本发明提供的计算机设备的结构框图。

具体实施方式

[0062] 为了对本发明的技术特征、目的和效果有更加清楚的理解,现对照附图详细说明本发明的具体实施方式。

[0063] 图1示出了本实施例的面向缺失数据的微服务系统根因定位方法的示意图。在本实施例中,面向缺失数据的微服务系统根因定位方法包括以下步骤:

[0064] S1:获取微服务监控数据,对微服务监控数据进行预处理,得到原序列数据和目标序列;

[0065] 具体地,上述面向缺失数据的微服务系统根因定位方法的步骤S1包括以下步骤:
S11:获取微服务监控数据,微服务监控数据包括CPU利用率、内存使用量、磁盘IO;S12:对微服务监控数据进行去异常值、标准化和插值填补,得到原序列数据和目标序列;

[0066] S2:对原序列数据进行时间特征提取,得到微服务监控数据时间特征向量和微服务监控数据空间特征向量,将微服务监控数据时间特征向量和微服务监控数据空间特征向量进行合并,得到拼接向量;

[0067] 具体地,上述面向缺失数据的微服务系统根因定位方法的步骤S2包括以下步骤:

[0068] S21:对原序列数据进行时间特征提取,得到微服务监控数据时间特征向量,如公式:

$$[0069] \quad \mathbf{X}'_{time}(b, h, t, n) =$$

$$[0070] \quad \text{ReLU}\left(\sum_{f=0}^{F-1} \sum_{m=0}^{k_t-1} \mathbf{X}_{time}(b, f, t+m, n) \cdot W_t(h, f, m) + b_t(h)\right),$$

[0071] 其中, $\mathbf{X}'_{time}(b, h, t, n)$ 为微服务监控数据时间特征向量,形状为 $[B, H_t, T, N]$, $[B, H_t, T, N]$, B 为数据批处理大小, H_t 表示时间维度卷积层输出的特征数量, T 为时间步数, N 为微服务系统节点数量, $\mathbf{X}_{time}()$ 是经过形状调整的输入数据,形状为 $[B, F, T, N]$, F 为每个节点监控的指标数量; b 为批次索引,表示当前处理的是哪一个批次中的样本; f 为特征通道索引,表示当前处理的是哪一个特征; h 为输出通道索引,对应于卷积层输出特征图中的一个特定“深度”或“通道”; t 为时间步索引,表示经过卷积操作后的时间维度上的位置; n 为节点索引,表示当前处理的是哪一个节点; $\text{ReLU}()$ 为激活函数; k_t 为时间维度上的卷积核大小; m 为卷积核的局部索引; $W_t()$ 为时间维度卷积核的权重参数; $b_t()$ 为时间维度卷积操作的偏置项;

[0072] S22:对原序列数据进行空间特征提取,得到微服务监控数据空间特征向量,如公式:

$$[0073] \quad \mathbf{X}'_{node}(b, h, f, t) = \text{ReLU}\left(\sum_{n=0}^{N-1} \sum_{m=0}^{k_n-1} \mathbf{X}_{node}(b, n, f+m, t) \cdot W_n(h, n, m) + b_n(h)\right)$$

[0074] 其中, $\mathbf{X}'_{node}(b, h, f, t)$ 为微服务监控数据空间特征向量,形状为 $[B, H_n, T, N]$, B 为数据批处理大小, H_n 表示空间维度卷积层输出的特征数量, T 为时间步数, N 为微服务系统节点数量; b 为批次索引, h 为输出通道索引, f 为特征通道索引, t 为时间步索引; $\text{ReLU}()$ 为激活函数; k_n 为在空间维度上卷积核大小; $\mathbf{X}_{node}()$ 是经过形状调整的输入数据,形状为 $[B, N, F, T]$; m 为卷积核的局部索引; $W_n()$ 为空间维度卷积核的权重参数; $b_n()$ 为空间维度卷积操作的偏置项;

[0075] S23:将微服务监控数据时间特征向量和微服务监控数据空间特征向量进行合并,得到拼接向量,如公式:

$$[0076] \quad \mathbf{X}_{combined} = \text{concat}(\mathbf{X}'_{time}, \mathbf{X}'_{node}, \text{dim} = 1)$$

[0077] 其中, $X_{combined}$ 是拼接向量, $\text{concat}(\cdot, \text{dim}=1)$ 表示在第二维上拼接两个向量, X'_{time} 为微服务监控数据时间特征向量, X'_{node} 为微服务监控数据空间特征向量;

[0078] S3: 根据拼接向量, 利用1D卷积、GRU时序网络和编码器隐藏层多层传感器, 得到隐藏状态、编码器输出;

[0079] 具体地, 上述面向缺失数据的微服务系统根因定位方法的步骤S3包括以下步骤:

[0080] S31: 利用1D卷积对拼接向量进行重塑操作, 得到重塑序列向量, 如公式:

$$[0081] \quad \mathbf{X}_{seq}[b, h', t'] = \text{ReLU}\left(\sum_{h=0}^{H_t+H_n-1} \sum_{m=0}^{k-1} \mathbf{X}_{combined}[b, h, s \cdot t' + m] \cdot W[h', h, m] + b[h']\right),$$

[0082] 其中, $X_{seq}[b, h', t']$ 为重塑序列向量, b 为批次索引, h' 为1D卷积的输出通道索引; t' 为1D卷积的时间步索引, $\text{ReLU}(\cdot)$ 为激活函数; H_t 为时间维度卷积层输出的特征数量, H_n 为空间维度卷积层输出的特征数量, k 为1D卷积的卷积核大小, s 为步长, h 为输出通道索引, m 为卷积核的局部索引, $W[h', h, m]$ 为1D卷积层的权重, $b[h']$ 为1D卷积操作的偏置项;

[0083] S32: 根据重塑序列向量, 利用GRU时序网络, 得到隐藏状态, 如公式:

$$[0084] \quad h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t$$

$$[0085] \quad z_t = \sigma(W_z[h_{t-1}, \mathbf{X}_{seq,t}] + b_z),$$

$$[0086] \quad r_t = \sigma(W_r[h_{t-1}, \mathbf{X}_{seq,t}] + b_r)$$

$$[0087] \quad \tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, \mathbf{X}_{seq,t}] + b_h)$$

[0088] 其中, h_t 为隐藏状态, z_t 为更新门, h_{t-1} 表示前一时间步的隐藏状态, $\odot$ 表示元素乘法; $\tilde{h}_t$ 为候选隐藏状态, σ 表示sigmoid函数, 确保门控制信号的值在0和1之间; W_z, W_r, W_h 为权重矩阵, $X_{seq,t}$ 表示在时间步 t 的输入特征向量; b_z, b_r, b_h 为偏置项; r_t 为重置门, $\tanh$ 提供非线性激活, 用于计算候选隐藏状态;

[0089] S33: 根据隐藏状态, 利用编码器隐藏层多层传感器, 得到编码器输出;

[0090] S4: 根据编码器输出和隐藏状态, 得到能量得分、注意力权重和上下文向量;

[0091] 具体地, 上述面向缺失数据的微服务系统根因定位方法的步骤S4包括以下步骤:

[0092] S41: 根据编码器输出和隐藏状态, 得到能量得分, 如公式:

$$[0093] \quad \mathbf{E} = \tanh(W_{attn} \cdot [h_{t-1}'; O_{enc}] + b_{attn})$$

[0094] 其中, E 为能量得分, $\tanh$ 为非线性激活, W_{attn} 为权重矩阵, h_{t-1}' 为解码器前一刻隐藏状态, O_{enc} 为编码器输出, b_{attn} 是偏置项;

[0095] S42: 根据能量得分, 得到注意力权重, 如公式:

$$[0096] \quad \mathbf{A} = \text{softmax}(v \cdot \mathbf{E}, \text{dim} = 1)$$

[0097] 其中, A 为注意力权重, $\text{softmax}(\cdot)$ 为归一化指数函数, v 是可学习参数, E 为能量得分, $\text{softmax}(\cdot, \text{dim}=1)$ 表示在第二维上对两个参数进行运算;

[0098] S43: 根据注意力权重和编码器输出, 得到上下文向量, 如公式:

$$[0099] \quad \mathbf{C}_t = \sum_{i=1}^T A_{ti} \cdot O_{enc,i}$$

[0100] 其中, $\mathbf{C}_t$ 为上下文向量, T 为时间步数, A_{ti} 是注意力权重中的第 i 个元素, 代表第 t 个解码步骤对第 i 个编码器输出的关注程度; $O_{enc,i}$ 是编码器输出中的第 i 个时间步的输出;

[0101] S5: 根据上下文向量, 利用GRU解码器, 得到重建后的数据特征;

[0102] 具体地, 上述面向缺失数据的微服务系统根因定位方法的步骤S5具体包括:

[0103] 根据步骤S4中上下文向量, 利用GRU解码器, 得到重建后的数据特征, 如公式:

$$[0104] \quad \mathbf{H}_M, h_t = \text{RNN}(\text{Concat}(\text{input}_t, \mathbf{C}_t), h_{t-1})$$

[0105] 其中, $\mathbf{H}_M$ 为解码器输出, 表示对缺失数据重建后得到的序列特征; h_t 是当前解码器隐藏状态; $\text{RNN}()$ 表示循环神经网络; $\text{Concat}()$ 为连接函数, input_t 是当前时间步的输入, $\mathbf{C}_t$ 是上下文向量, h_{t-1} 是前一时间步的解码器隐藏状态, 通过比较目标序列和重建序列 $\mathbf{H}_M$ 的差值计算损失, 使用均方误差 (MSE) 作为重建任务的损失函数, 基于损失函数的结果通过反向传播算法计算模型参数的梯度, 并使用Adam优化器更新模型的权重以减少预测和实际值之间的差异;

[0106] S6: 根据重建后的数据特征, 对节点的故障概率进行预测, 得到系统故障根因;

[0107] 具体地, 上述面向缺失数据的微服务系统根因定位方法的步骤S6具体包括:

[0108] 根据重建后的数据特征, 对节点的故障概率进行预测, 得到每个节点的故障概率, 根据每个节点的故障概率, 得到系统故障根因, 如公式:

$$[0109] \quad \text{Faults} = \text{argsort}\left(\frac{\exp[(\mathbf{W}\mathbf{H}_G + b_f)_i]}{\sum_{j=1}^N \exp[(\mathbf{W}\mathbf{H}_G + b_f)_j]}\right)[-5:]$$

[0110] 其中, Faults 为节点的故障概率, $\text{argsort}()$ 表示将元素从小到大排列, 并提取元素对应的索引号; $\exp$ 为自然指数函数, N 是微服务系统节点个数, i 是全连接输出向量的第 i 个值, $\mathbf{W}$ 是全连接层的权重矩阵, $\mathbf{H}_G$ 为 $\mathbf{H}_M$ 经过PCA降维后的一维向量, b_f 是偏置向量。

[0111] 在一些实施例, 上述的面向缺失数据的微服务系统根因定位方法也可以通过以下方式实施。

[0112] 在实施例, 如图2所示, 面向缺失数据的微服务系统根因定位方法包括以下步骤: S1: 对微服务系统监控数据进行去异常值、标准化和插值填补等预处理; S2: 基于卷积网络对输入数据构建时间和空间维度的向量特征并拼接为 $\mathbf{X}_{\text{combined}}$; S3: 通过GRU时序网络编码器获取S2中拼接向量 $\mathbf{X}_{\text{combined}}$ 的隐藏层状态, 并输入MLP得到编码器输出 O_{enc} ; S4: 将S3中的编码器输出 O_{enc} 进行注意力对齐, 得到解码器的注意力上下文向量 $\mathbf{C}$; S5: 通过GRU解码器对注意力上下文向量 $\mathbf{C}$ 进行解码, 得到重建后的数据特征; S6: 基于重建后的数据特征, 对节点的故障概率进行预测, 实现故障根因排序;

[0113] 具体地, 上述面向缺失数据的微服务系统根因定位方法的步骤S1具体为: 对收集到的微服务监控数据 (例如, CPU利用率、内存使用量、磁盘IO等指标) 使用3-sigma异常检测方法去除异常值; 使用Z-score方法进行标准化处理; 对于数据中的缺失部分, 通过平均插值进行初步填补, 以便模型能够在有监督的学习环境中训练; 设输入数据 $\mathbf{X}$ 的形状为 $[B, T]$,

$F, N]$, 其中 B 是数据批处理大小, T 是时间步数, N 是微服务系统的节点数量, F 是每个节点监控的指标数量;

[0114] 具体地, 上述面向缺失数据的微服务系统根因定位方法的步骤S2具体为:S21: 基于时间序列的动态特征以及节点间的相互关系对输入的预处理数据进行时间层面特征学习; 设 X_{time} 是 X 经过形状调整的输入数据, 形状为 $[B, F, TN]$, 则时间维度的卷积可以表达为:

$$[0115] \quad \mathbf{X}'_{time}(b, h, t, n) = \text{ReLU}\left(\sum_{f=0}^{F-1} \sum_{m=0}^{k_t-1} \mathbf{X}_{time}(b, f, t+m, n) \cdot W_t(h, f, m) + b_t(h)\right)$$

[0116] 其中: b 为批次索引, 表示当前处理的是哪一个批次中的样本; h 为输出通道索引, 对应于卷积层输出特征图中的一个特定“深度”或“通道”; t 为时间步索引, 表示经过卷积操作后的时间维度上的位置; n 为节点索引, 表示当前处理的是哪一个节点; m 为卷积核的局部索引; W_t 为时间维度卷积核的权重参数; b_t 为时间维度卷积操作的偏置项。 X'_{time} 的形状为 $[B, H_t, T, N]$, 其中 H_t 表示时间维度卷积层输出的特征数量;S22: 在空间维度向量化指标数据;对于空间维度的卷积, 设卷积核尺寸为 (k_n, N) , 其中 k_n 是在空间维度上卷积核大小;设 X'_{node} 是经过形状调整的输入数据, 形状为 $[B, N, F, T]$, 则指标数据节点空间维度的卷积可以表达为:

$$[0117] \quad \mathbf{X}'_{node}(b, h, f, t) = \text{ReLU}\left(\sum_{n=0}^{N-1} \sum_{m=0}^{k_n-1} \mathbf{X}_{node}(b, n, f+m, t) \cdot W_n(h, n, m) + b_n(h)\right)$$

[0118] 其中, W_n 为空间维度卷积核的权重参数; b_n 为空间维度卷积操作的偏置项;S23: 合并指标数据在时间维度和空间维度的特征向量;将时间维度的卷积输出 X'_{time} 和节点空间维度的卷积输出 X'_{node} 在特征维度上(假定为第二维, 即通道维度)进行合并;之前步骤得到的 X'_{time} 和 X'_{node} 的形状分别为 $[B, H_t, T, N]$ 和 $[B, H_n, T, N]$, 合并操作可以表示为:

$$[0119] \quad \mathbf{X}_{combined} = \text{concat}(\mathbf{X}'_{time}, \mathbf{X}'_{node}, \text{dim} = 1)$$

[0120] 其中 $\text{concat}(\cdot, \text{dim}=1)$ 表示在第二维上拼接两个向量, $X_{combined}$ 的形状变为 $[B, (H_t + H_n), T, N]$;

[0121] 具体地, 上述面向缺失数据的微服务系统根因定位方法的步骤S3具体为:S31: 通过1D卷积对 $X_{combined}$ 进行重塑操作, 以匹配1D卷积期望的输入形状, 加速序列数据处理;将时间步 T 和节点数 N 合并为一个新的维度, 因此 $X_{combined}$ 被重塑为 $[B, H_t + H_n, T \times N]$;假设1D卷积的卷积核大小为 k , 步长为 s , 1D卷积沿着合并的 $T \times N$ 维度滑动, 每次滑动覆盖 k 个连续元素并产生一个新的特征值;这一过程可以表示为:

$$[0122] \quad \mathbf{X}_{seq}[b, h', t'] = \text{ReLU}\left(\sum_{h=0}^{H_t+H_n-1} \sum_{m=0}^{k-1} \mathbf{X}_{combined}[b, h, s \cdot t' + m] \cdot W[h', h, m] + b[h']\right)$$

[0123] 其中, h' 为1D卷积的输出通道索引; t' 为1D卷积的时间步索引;经过1D卷积层后, 输出 X_{seq} 的形状为 $[B, H_{out}, T \times N]$, 其中 B 是批次大小, H_{out} 是卷积层输出的特征数量, $T \times N$ 表示时间步和节点数的组合维度;S32: 通过GRU时序网络获取隐层特征;首先将 X_{seq} 的形状从 $[B, H_{out}, T \times N]$ 重塑为 $[B, H_{out}, T, N]$, 将时间步 T 和节点数 N 明确分离;然后通过Permute操作将 X_{seq} 的维度进行转置, 使其形状变为 $[B, T, H_{out}, N]$, 此时每个时间步 T 都包含了 $H_{out} \times N$ 个特征;最后, 通过Reshape操作将最后两维合并, 得到最终形状 $[B, T, H_{out} \times N]$, 这样每个时间步

对应的特征维度就是节点数乘以卷积输出特征数量;对于每个时间步 t ,GRU的隐藏状态更新公式为:

$$[0124] \quad h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t$$

[0125] 其中,

$$[0126] \quad z_t = \sigma(W_z[h_{t-1}, \mathbf{X}_{seq,t}] + b_z)$$

$$[0127] \quad r_t = \sigma(W_r[h_{t-1}, \mathbf{X}_{seq,t}] + b_r)$$

$$[0128] \quad \tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, \mathbf{X}_{seq,t}] + b_h)$$

[0129] $\mathbf{X}_{seq,t}$ 表示在时间步 t 的输入特征向量; h_{t-1} 表示前一时间步的隐藏状态; z_t 为更新门, r_t 为重置门, $\tilde{h}_t$ 为候选隐藏状态; $\odot$ 表示元素乘法; W_z, W_r, W_h 为权重矩阵, b_z, b_r, b_h 为偏置项; σ 表示sigmoid函数,确保门控制信号的值在0和1之间; $\tanh$ 提供非线性激活,用于计算候选隐藏状态;S33:将隐藏状态 h_t 送入 H_{enc} 层的MLP,最后得到编码器输出 O_{enc} ,其形状为 $[B, T, H_{enc}]$,其中 B 是批次大小, T 是时间步数, H_{enc} 是编码器隐藏层的大小;

[0130] 具体地,上述面向缺失数据的微服务系统根因定位方法的步骤S4具体为:S41:计算能量得分;将GRU作为解码器主干网络,基于解码器前一时刻隐藏状态 h_{t-1}' 和编码器输出 O_{enc} 计算能量得分,表示为 E ,计算过程如下:

$$[0131] \quad \mathbf{E} = \tanh(W_{attn} \cdot [h_{t-1}', O_{enc}] + b_{attn})$$

[0132] 其中 W_{attn} 是一个形状为 $[H_{dec} + H_{enc}, H_{dec}]$ 的权重矩阵, H_{dec} 表示解码器的隐藏层大小, b_{attn} 是偏置项; E 的形状为 $[B, T, H_{dec}]$,表示每个时间步对当前解码步骤的贡献;S42:计算注意力权重;接下来通过对能量得分 E 应用softmax函数,计算注意力权重 A :

$$[0133] \quad \mathbf{A} = \text{softmax}(v \cdot \mathbf{E}, \dim = 1)$$

[0134] 其中 v 是可学习参数;通过计算注意力权重将能量得分转换成一个概率分布,表示各个输入序列元素对当前解码步骤的重要性;S43:最后,使用注意力权重 A 对编码器的输出 O_{enc} 进行加权求和,得到上下文矩阵 C :

$$[0135] \quad \mathbf{C}_t = \sum_{i=1}^T A_{ti} \cdot O_{enc,i}$$

[0136] 其中, A_{ti} 是注意力权重中的第 i 个元素,代表第 t 个解码步骤对第 i 个编码器输出的关注程度; $O_{enc,i}$ 是编码器输出中的第 i 个时间步的输出;上下文向量 C_t 通过对所有编码器输出进行加权平均得到。综合以上步骤,得到的上下文向量 C 可以表达为:

$$[0137] \quad \mathbf{C} = \sum_{t=1}^T \text{softmax}(\tanh(W_{attn} \cdot [h_{dec}, O_{enc}] + b_{attn}))_t \odot O_{enc,t}$$

[0138] 具体地,上述面向缺失数据的微服务系统根因定位方法的步骤S5具体为:基于编码器的信息和之前的输出来更新解码器的状态,将上下文向量 C_t 和当前时间步的输入 $input_t$ 一起送入解码器GRU单元进行处理:

[0139] $\mathbf{H}_M, h_t = \text{RNN}(\text{Concat}(input_t, C_t), h_{t-1}')$

[0140] 其中 $input_t$ 是当前时间步的输入, C_t 是上下文向量, h_{t-1}' 是前一时间步的解码器隐藏状态。 H_M 为解码器输出,表示对缺失数据重建后得到的序列特征;

[0141] 通过比较目标序列和重建序列 H_M 的差值计算损失,使用均方误差(MSE)作为重建任务的损失函数,基于损失函数的结果通过反向传播算法计算模型参数的梯度,并使用Adam优化器更新模型的权重以减少预测和实际值之间的差异;

[0142] 具体地,上述面向缺失数据的微服务系统根因定位方法的步骤S6具体为:将 H_M 降维成一维向量,再输入到全连接层进行维度映射,然后经过softmax层进行概率预测,得到每个节点的故障概率Faults:

$$[0143] \quad Faults = \text{argsort}\left(\frac{\exp[(\mathbf{W}\mathbf{H}_G + b_f)_i]}{\sum_{j=1}^N \exp[(\mathbf{W}\mathbf{H}_G + b_f)_j]}\right)[-5:]$$

[0144] 其中,N是微服务系统节点个数, i 是全连接输出向量的第 i 个值, $\mathbf{W}$ 和 b_f 是全连接层的权重矩阵以及偏置向量,argsort()表示将元素从小到大排列,并提取元素对应的索引号;后对Faults中的节点根据故障概率进行排序,概率越大的越有可能是故障根因。

[0145] 在本实施例中,以Sockshop微服务系统作为数据集来源;SockShop是一个真实运行的软件系统,完全采用了云原生技术和微服务架构进行容器化;这个应用是一个在线销售袜子的商店,专注于为用户提供前台交互界面;它被设计用来展示和测试微服务架构以及云原生技术的实用性;SockShop包含了多个微服务实例,如前端服务、订单服务、支付服务、用户服务、用户数据库服务、购物车服务、分类服务、购物服务和消息队列服务等,这些服务共同协作支撑微服务系统的运行;实施例对所选用的数据集进行了特定的掩码处理,以生成固定类型的非完整指标数,旨在为不同的对比方法提供一个统一且公正的非完备指标数据下的实验环境;

[0146] 根因定位任务常用命中率(Hit Rate)来评估预测根因节点的准确性;在本实施例中采用HR@1, HR@3, HR@5准确率和评价Top5 (Avg5) 准确率作为主要的评估标准;HR@k提供了一个清晰的度量方式,可以量化在前 k 个预测实例中包含真实根因实例的概率,是领域内根因定位工作中常用衡量系统定位能力的指标;HR@k在数学上可以用以下的公式计算得出:

$$[0147] \quad HR@k = \frac{1}{|F|} \sum_{f \in F} \begin{cases} 1, & \text{if } RC_i \in RC_{pk} \\ 0, & \text{otherwise} \end{cases}$$

[0148] 其中, F 为测试集中出现错误的样本集合, RC_i 为对于第 i 个样本,其真实异常根因实例; C_{pk} 代表模型预测的Top- k 异常实例集合结果; k 代表具体对应指标HR@值;

[0149] 除基础评价指标外,本实施例引入Avg5作为本场景下额外的指标,以明确对比各个方法的性能,具体计算方式如下:

$$[0150] \quad Avg5 = \frac{\sum_{i=1}^5 HR@i}{5}$$

[0151] 最终通过和利用指标数据进行根因分析的先进方法PC_based、GES_based、LiNGAM_based和CausalRCA比较,本发明所提出的根因定位方法取得了较好的结果,如表1所示。

[0152] 表1:在Sockshop数据集上的根因定位性能

对比方法	Dataset_Sockshop			
	HR@1	HR@3	HR@5	Avg5
PC_Based	0.2349	0.4432	0.6179	0.4320
[0153] GES_Based	0.2491	0.4789	0.7729	0.5003
LiNGAM_Based	0.1984	0.6763	0.8632	0.5793
CausalRCA	0.1632	0.5323	0.7695	0.4883
本发明	0.2842	0.7942	0.8823	0.6536

[0154] 本实施例提供计算机可读存储介质,其上存储有计算机程序,该计算机程序被处理器执行时实现上述的面向缺失数据的微服务系统根因定位方法的步骤。其中,该存储介质可为磁碟、光盘、只读存储记忆体(Read-Only Memory,ROM)、随机存储记忆体(Random Access Memory,RAM)、快闪存储器(Flash Memory)、硬盘(Hard Disk Drive,缩写:HDD)或固态硬盘(Solid-State Drive,SSD)等;该存储介质还可以包括上述种类的存储器的组合。

[0155] 本实施例提供计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,该处理器执行该程序时实现上述的面向缺失数据的微服务系统根因定位方法的步骤。

[0156] 如图3所示,该计算机设备可以包括:至少一个处理器121,例如CPU(Central Processing Unit,中央处理器),至少一个通信接口123,存储器124,至少一个通信总线122。其中,通信总线122用于实现这些组件之间的连接通信。其中,通信接口123可以包括显示屏(Display)、键盘(Keyboard),可选通信接口123还可以包括标准的有线接口、无线接口。存储器124可以是高速RAM存储器(Random Access Memory,易挥发性随机存取存储器),也可以是非不稳定的存储器(non-volatile memory),例如至少一个磁盘存储器。存储器124可选的还可以是至少一个位于远离前述处理器121的存储装置。其中,存储器124中存储应用程序,且处理器121调用存储器124中存储的程序代码,以用于执行上述任一方法步骤。其中,通信总线122可以是外设部件互连标准(peripheral component interconnect,简称PCI)总线或扩展工业标准结构(extended industry standard architecture,简称EISA)总线等。通信总线122可以分为地址总线、数据总线、控制总线等。为便于表示,图3中仅用一条线表示,但并不表示仅有一根总线或一种类型的总线。其中,存储器124可以包括易失性存储器(英文:volatile memory),例如随机存取存储器(英文:random-access memory,缩写:RAM);存储器也可以包括非易失性存储器(英文:non-volatile memory),例如快闪存储器(英文:flash memory),硬盘(英文:hard disk drive,缩写:HDD)或固态硬盘(英文:solid-state drive,缩写:SSD);存储器124还可以包括上述种类的存储器的组合。其中,处

理器121可以是中央处理器(英文:central processing unit,缩写:CPU),网络处理器(英文:network processor,缩写:NP)或者CPU和NP的组合。其中,处理器121还可以进一步包括硬件芯片。上述硬件芯片可以是专用集成电路(英文:application-specific integrated circuit,缩写:ASIC),可编程逻辑器件(英文:programmable logic device,缩写:PLD)或其组合。上述PLD可以是复杂可编程逻辑器件(英文:complex programmable logic device,缩写:CPLD),现场可编程逻辑门阵列(英文:field-programmable gate array,缩写:FPGA),通用阵列逻辑(英文:generic array logic,缩写:GAL)或其任意组合。可选地,存储器124还用于存储程序指令。处理器121可以调用程序指令,实现如本实施例的面向缺失数据的微服务系统根因定位方法。

[0157] 上面结合附图对本发明的实施例进行了描述,但是本发明并不局限于上述的具体实施方式,上述的具体实施方式仅仅是示意性的,而不是限制性的,本领域的普通技术人员在本发明的启示下,在不脱离本发明宗旨和权利要求所保护的范围情况下,还可做出很多形式,这些均属于本发明的保护之内。

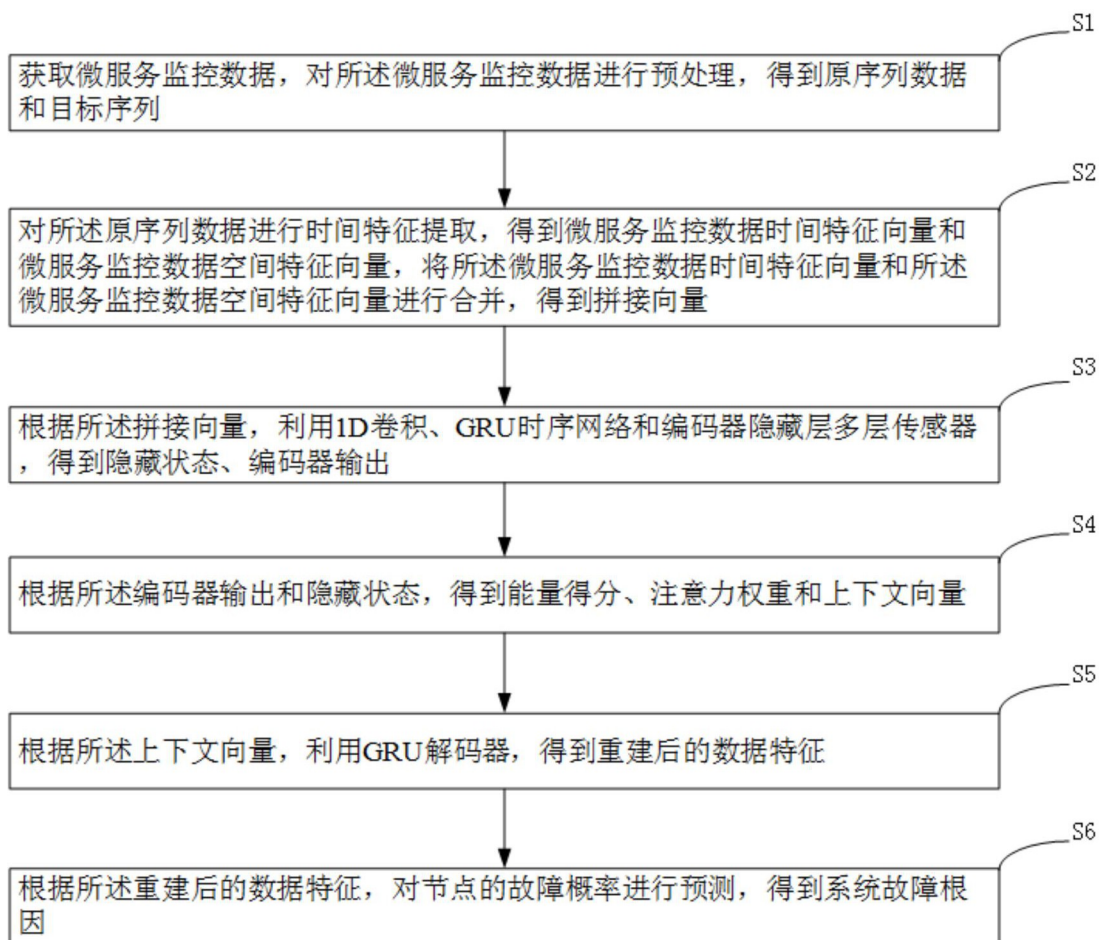


图1

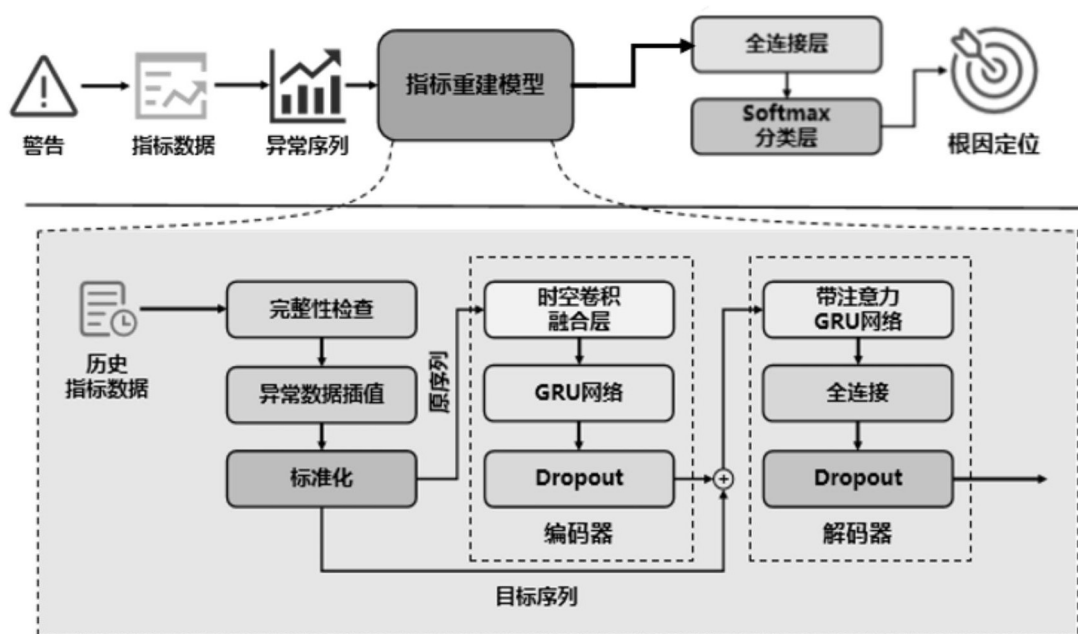


图2

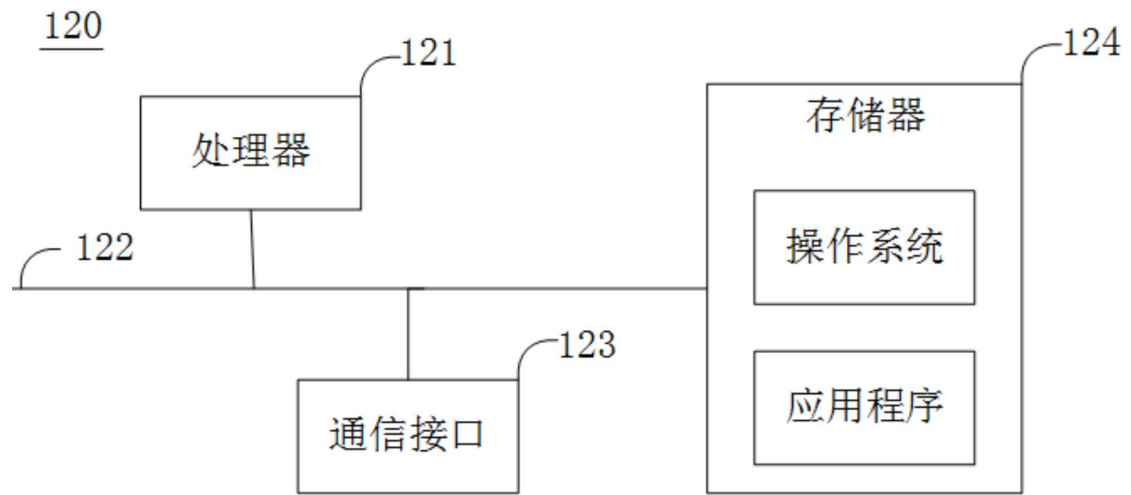


图3