



(12) 发明专利

(10) 授权公告号 CN 116662811 B

(45) 授权公告日 2024. 02. 06

(21) 申请号 202310700161.0

G06Q 10/20 (2023.01)

(22) 申请日 2023.06.13

(56) 对比文件

(65) 同一申请的已公布的文献号

申请公布号 CN 116662811 A

CN 105095515 A, 2015.11.25

CN 106980665 A, 2017.07.25

CN 115186764 A, 2022.10.14

(43) 申请公布日 2023.08.29

CN 116028618 A, 2023.04.28

CN 116192523 A, 2023.05.30

(73) 专利权人 无锡物联网创新中心有限公司

地址 214028 江苏省无锡市新加坡工业园

新集路1-1号及1-2号厂房

US 2021326728 A1, 2021.10.21

CN 114386521 A, 2022.04.22

CN 114065862 A, 2022.02.18

(72) 发明人 章军辉 付宗杰 顾煜帆 董接莲

陈明亮 庄宝森 陈大鹏

CN 114519469 A, 2022.05.20

CN 115018021 A, 2022.09.06

(74) 专利代理机构 北京集佳知识产权代理有限公司

11227

专利代理师 姚莹丽

CN 115078894 A, 2022.09.20

CN 115098620 A, 2022.09.23

CN 115222141 A, 2022.10.21

CN 115484102 A, 2022.12.16

(51) Int. Cl.

G06F 18/214 (2023.01)

G06F 18/22 (2023.01)

G06N 3/045 (2023.01)

G06N 3/08 (2023.01)

G06N 3/049 (2023.01)

G06Q 10/0639 (2023.01)

CN 115587335 A, 2023.01.10

CN 116151258 A, 2023.05.23

JP 2022066957 A, 2022.05.02

US 2019392302 A1, 2019.12.26

审查员 张皓静

权利要求书2页 说明书8页 附图2页

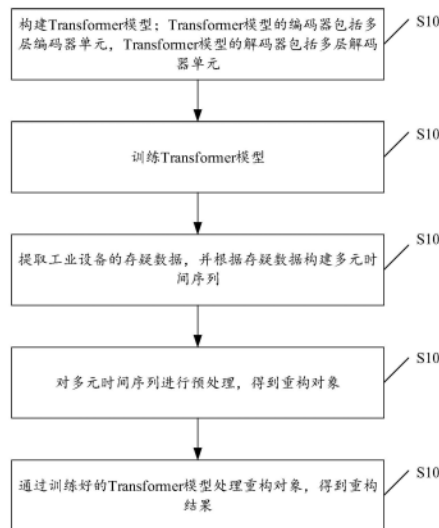
(54) 发明名称

一种工业设备的时序状态数据重构方法及  
相关装置

(57) 摘要

本申请公开了一种工业设备的时序状态数据重构方法及相关装置,涉及工业设备异常检测技术领域,包括:构建Transformer模型;Transformer模型的编码器包括多层编码器单元,Transformer模型的解码器包括多层解码器单元;训练Transformer模型;提取工业设备的存疑数据,并根据存疑数据构建多元时间序列;对多元时间序列进行预处理,得到重构对象;通过训练好的Transformer模型处理重构对象,得到重构结果。该方法能够快速可靠的进行数据重构,有利于进一步基于重构数据进行工业设备异

常检测,提升工业设备异常检测的效率。



CN 116662811 B

1. 一种工业设备的时序状态数据重构方法,其特征在于,包括:
  - 构建Transformer模型;所述Transformer模型的编码器包括多层编码器单元,所述Transformer模型的解码器包括多层解码器单元;
  - 训练所述Transformer模型;
  - 提取工业设备的存疑数据,并根据所述存疑数据构建多元时间序列;
  - 对所述多元时间序列进行预处理,得到重构对象;
  - 通过训练好的所述Transformer模型处理所述重构对象,得到重构结果;
  - 所述通过训练好的所述Transformer模型处理所述重构对象,得到重构结果包括:
    - 通过所述Transformer模型的所述编码器与所述解码器处理所述重构对象,并对所述解码器的中间输出进行全连接处理与归一化处理得到所述重构结果;其中,通过所述编码器与所述解码器处理所述重构对象包括:进行注意力计算;
    - 所述进行注意力计算包括:
      - 构建不同的哈希函数,并利用所述哈希函数对query与key进行编码,得到哈希值,根据所述哈希值对query与key进行分桶,得到最优分桶方案;
      - 根据所述最优分桶方案确定候选集;
      - 计算查询数据与所述候选集中各数据的相似度,并返回最近的多个近邻;
      - 在桶内对所述query与所述近邻进行点积计算,得到近似注意力。
2. 根据权利要求1所述的工业设备的时序状态数据重构方法,其特征在于,所述训练所述Transformer模型包括:
  - 采用元学习算法训练所述Transformer模型。
3. 根据权利要求2所述的工业设备的时序状态数据重构方法,其特征在于,所述采用元学习算法训练所述Transformer模型包括:
  - 随机初始化所述Transformer模型的权重参数;
  - 载入训练序列并对所述训练序列进行滑窗处理,得到多个窗口矩阵;
  - 处理所述窗口矩阵,更新各个所述权重参数。
4. 根据权利要求3所述的工业设备的时序状态数据重构方法,其特征在于,所述更新各个所述权重参数包括:
  - 根据 $\phi = \phi - \alpha \nabla_{\phi} \sum_{i=1}^T \mathcal{L}_i(f_{\phi})$ 更新所述权重参数; $\phi$ 表示所述权重参数, $\alpha$ 表示元学习的步长, $\nabla_{\phi}$ 表示梯度算子, $\mathcal{L}_i$ 表示代价函数, $f_{\phi}$ 表示网络模型,T表示所述窗口矩阵的个数。
5. 根据权利要求1所述的工业设备的时序状态数据重构方法,其特征在于,对所述多元时间序列进行预处理,得到重构对象包括:
  - 对所述多元时间序列进行滑窗处理,得到多个窗口矩阵;
  - 对所述窗口矩阵进行PE编码,得到所述重构对象。
6. 一种工业设备的时序状态数据重构装置,其特征在于,包括:
  - 构建模块,用于构建Transformer模型;所述Transformer模型的编码器包括多层编码器单元,所述Transformer模型的解码器包括多层解码器单元;
  - 训练模块,用于训练所述Transformer模型;

提取模块,用于提取工业设备的存疑数据,并根据所述存疑数据构建多元时间序列;  
预处理模块,用于对所述多元时间序列进行预处理,得到重构对象;  
重构模块,用于通过训练好的所述Transformer模型处理所述重构对象,得到重构结果;

重构模块具体用于:

通过所述Transformer模型的所述编码器与所述解码器处理所述重构对象,并对所述解码器的中间输出进行全连接处理与归一化处理得到所述重构结果;其中,通过所述编码器与所述解码器处理所述重构对象包括:进行注意力计算;

所述进行注意力计算包括:

构建不同的哈希函数,并利用所述哈希函数对query与key进行编码,得到哈希值,根据所述哈希值对query与key进行分桶,得到最优分桶方案;

根据所述最优分桶方案确定候选集;

计算查询数据与所述候选集中各数据的相似度,并返回最近的多个近邻;

在桶内对所述query与所述近邻进行点积计算,得到近似注意力。

7.一种工业设备的时序状态数据重构设备,其特征在于,包括:

存储器,用于存储计算机程序;

处理器,用于执行所述计算机程序时实现如权利要求1至5任一项所述的工业设备的时序状态数据重构方法的步骤。

8.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有计算机程序,所述计算机程序被处理器执行时实现如权利要求1至5任一项所述的工业设备的时序状态数据重构方法的步骤。

## 一种工业设备的时序状态数据重构方法及相关装置

### 技术领域

[0001] 本申请涉及工业设备异常检测技术领域,特别涉及一种工业设备的时序状态数据重构方法;还涉及一种工业设备的时序状态数据重构装置、设备以及计算机可读存储介质。

### 背景技术

[0002] 为了及时发现工业设备的潜在异常,排查隐性故障,提高生产效率,降低运维成本,在工业设备异常检测时,可以首先进行数据重构,进而根据重构结果来判断工业设备是否异常。数据重构的速度无疑关乎设备整体的异常检测速度。同时,数据重构的可靠性直接影响设备整体的异常检测可靠性。因此,提供一种能够快速可靠的数据重构方案已成为本领域技术人员亟待解决的技术问题。

### 发明内容

[0003] 本申请的目的是提供一种工业设备的时序状态数据重构方法,能够快速可靠的重构工业时序数据,有利于工业设备异常检测。本申请的另一个目的是提供一种工业设备的时序状态数据重构装置、设备以及计算机可读存储介质,均具有上述技术效果。

[0004] 为解决上述技术问题,本申请提供了一种工业设备的时序状态数据重构方法,包括:

[0005] 构建Transformer模型;所述Transformer模型的编码器包括多层编码器单元,所述Transformer模型的解码器包括多层解码器单元;

[0006] 训练所述Transformer模型;

[0007] 提取工业设备的存疑数据,并根据所述存疑数据构建多元时间序列;

[0008] 对所述多元时间序列进行预处理,得到重构对象;

[0009] 通过训练好的所述Transformer模型处理所述重构对象,得到重构结果。

[0010] 可选的,所述训练所述Transformer模型包括:

[0011] 采用元学习算法训练所述Transformer模型。

[0012] 可选的,所述采用元学习算法训练所述Transformer模型包括:

[0013] 随机初始化所述Transformer模型的权重参数;

[0014] 载入训练序列并对所述训练序列进行滑窗处理,得到多个窗口矩阵;

[0015] 处理所述窗口矩阵,更新各个所述权重参数。

[0016] 可选的,所述更新各个所述权重参数包括:

[0017] 根据  $\phi = \phi - \alpha \nabla_{\phi} \sum_{i=1}^T \mathcal{L}_i(f_{\phi})$  更新所述权重参数;  $\phi$  表示所述权重参数,  $\alpha$  表示元学

习的步长,  $\nabla_{\phi}$  表示梯度算子,  $\mathcal{L}_i$  表示代价函数,  $f_{\phi}$  表示网络模型,  $T$  表示所述窗口矩阵的个数。

[0018] 可选的,对所述多元时间序列进行预处理,得到重构对象包括:

[0019] 对所述多元时间序列进行滑窗处理,得到多个窗口矩阵;

- [0020] 对所述窗口矩阵进行PE编码,得到所述重构对象。
- [0021] 可选的,所述通过训练好的所述Transformer模型处理所述重构对象,得到重构结果包括:
- [0022] 通过所述Transformer模型的所述编码器与所述解码器处理所述重构对象,并对所述解码器的中间输出进行全连接处理与归一化处理得到所述重构结果;其中,通过所述编码器与所述解码器处理所述重构对象包括:进行注意力计算。
- [0023] 可选的,所述进行注意力计算包括:
- [0024] 构建不同的哈希函数,并利用所述哈希函数对query与key进行编码,得到哈希值,根据所述哈希值对query与key进行分桶,得到最优分桶方案;
- [0025] 根据所述最优分桶方案确定候选集;
- [0026] 计算查询数据与所述候选集中各数据的相似度,并返回最近的多个近邻;
- [0027] 在桶内对所述query与所述近邻进行点积计算,得到近似注意力。
- [0028] 为解决上述技术问题,本申请还提供了一种工业设备的时序状态数据重构装置,包括:
- [0029] 构建模块,用于构建Transformer模型;所述Transformer模型的编码器包括多层编码器单元,所述Transformer模型的解码器包括多层解码器单元;
- [0030] 训练模块,用于训练所述Transformer模型;
- [0031] 提取模块,用于提取工业设备的存疑数据,并根据所述存疑数据构建多元时间序列;
- [0032] 预处理模块,用于对所述多元时间序列进行预处理,得到重构对象;
- [0033] 重构模块,用于通过训练好的所述Transformer模型处理所述重构对象,得到重构结果。
- [0034] 为解决上述技术问题,本申请还提供了一种工业设备的时序状态数据重构设备,包括:
- [0035] 存储器,用于存储计算机程序;
- [0036] 处理器,用于执行所述计算机程序时实现如上所述的工业设备的时序状态数据重构方法的步骤。
- [0037] 为解决上述技术问题,本申请还提供了一种计算机可读存储介质,所述计算机可读存储介质上存储有计算机程序,所述计算机程序被处理器执行时实现如上所述的工业设备的时序状态数据重构方法的步骤。
- [0038] 本申请所提供的工业设备的时序状态数据重构方法,包括:构建Transformer模型;所述Transformer模型的编码器包括多层编码器单元,所述Transformer模型的解码器包括多层解码器单元;训练所述Transformer模型;提取工业设备的存疑数据,并根据所述存疑数据构建多元时间序列;对所述多元时间序列进行预处理,得到重构对象;通过训练好的所述Transformer模型处理所述重构对象,得到重构结果。
- [0039] 可见,本申请所提供的工业设备的时序状态数据重构方法,提取的是存疑数据不包含正常数据,并对基于存疑数据构建的多元时间序列进行数据重构,如此在数据重构过程中可以减少大量的不必要的计算,从而加快数据重构速度。另外,本申请构建并训练多层的Transformer模型,利用训练好的多层的Transformer模型进行数据重构,能够提高数据

重构的可靠性,有利于进一步基于重构数据进行工业设备异常检测,提升工业设备异常检测的效率。

[0040] 本申请所提供的工业设备的时序状态数据重构装置、设备以及计算机可读存储介质均具有上述技术效果。

### 附图说明

[0041] 为了更清楚地说明本申请实施例中的技术方案,下面将对现有技术和本申请实施例中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本申请的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0042] 图1为本申请实施例所提供的一种工业设备的时序状态数据重构方法的流程示意图;

[0043] 图2为本申请实施例所提供的一种工业设备的时序状态数据重构实现框图;

[0044] 图3为本申请实施例所提供的一种工业设备的时序状态数据重构装置的示意图;

[0045] 图4为本申请实施例所提供的一种工业设备的时序状态数据重构设备的示意图。

### 具体实施方式

[0046] 本申请的核心是提供一种工业设备的时序状态数据重构方法,能够快速可靠的重构工业时序数据,有利于工业设备异常检测。本申请的另一个核心是提供一种工业设备的时序状态数据重构装置、设备以及计算机可读存储介质,均具有上述技术效果。

[0047] 为使本申请实施例的目的、技术方案和优点更加清楚,下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范围。

[0048] 请参考图1,图1为本申请实施例所提供的一种工业设备的时序状态数据重构方法的流程示意图,参考图1所示,该方法包括:

[0049] S101:构建Transformer模型;所述Transformer模型的编码器包括多层编码器单元,所述Transformer模型的解码器包括多层解码器单元;

[0050] Transformer模型包括编码器与解码器。编码器包括多层级联的编码器单元,每层编码器单元的结构相同,参数不共享。前一层编码器单元的输出作为当前编码器单元的输入。解码器包括多层级联的解码器单元,每层解码器单元的结构相同,参数不共享。前一层解码器单元的输出作为当前解码器单元的输入。

[0051] S102:训练所述Transformer模型;

[0052] 本步骤旨在训练Transformer模型,以便以训练好的Transformer模型作为重构模型进行数据重构。

[0053] 其中,训练Transformer模型是利用正常数据构建的多元时间序列来训练Transformer模型,使Transformer模型学习到正常数据之间长短时依赖性与多特征相关性,从而通过数据重构,找出与正常数据偏离较大而无法被有效重构的异常事件。

[0054] 另外,工业时序数据有限,为了能够利用有限的工业时序数据训练Transformer模

型,在一些实施例中,所述训练所述Transformer模型包括:

[0055] 采用元学习算法训练所述Transformer模型。

[0056] 面对数据缺乏的场景,深度学习神经网络的学习能力与对新任务的快速泛化能力有效。而元学习(Model Agnostic Meta Learning, MAML)是利用模型在已有相关任务上的学习经验(元知识)来指导新任务的学习,从而使得模型能够在小样本条件下适应新任务。为了训练出合适的模型参数,使得模型在小样本情况下能够快速收敛,本实施例采用元学习算法对Transformer模型进行多任务训练,主要步骤包括:

[0057] Step1:随机初始化网络(编码器、解码器)的权重参数;

[0058] Step2:载入一段新的训练序列,并进行滑窗处理,得到T个窗口矩阵;

[0059] Step3:执行第i个任务,计算网络中各个参数的梯度  $\phi'_i = \phi - \gamma \nabla_{\phi} \mathcal{L}_i(f_{\phi})$ ,其中  $\nabla_{\phi}$  为梯度算子,  $f_{\phi}$  代表网络模型,  $\phi$  代表某个模型参数,  $\phi'_i$  代表执行任务i时参数  $\phi$  的梯度变化,  $\mathcal{L}_i$  代表执行任务i时的代价函数,  $\gamma$  为学习率;

[0060] Step4:按照step3完成T个任务;

[0061] Step5:一轮结束后,更新各个参数  $\phi = \phi - \alpha \nabla_{\phi} \sum_{i=1}^T \mathcal{L}_i(f_{\phi})$ ,其中  $\alpha$  为元学习的步

长;

[0062] Step6:当迭代截止条件满足时,结束Step2~Step5,完成训练,反之继续。

[0063] S103:提取工业设备的存疑数据,并根据所述存疑数据构建多元时间序列;

[0064] 为了减少不必要的冗余计算,提高效率,本实施例对采集到的工业时序数据(例如温度数据、电流数据、电压数据等等)中可能异常的数据(即存疑数据)进行标识与提取,并对存疑数据进行归一化处理。基于提取的存疑数据构建多元时间序列。多元时间序列可表示为:  $X = \{x_1, x_2, \dots, x_t, \dots, x_T\}^T$ ,其中, T为时间序列长度,  $x_t = [x_t^1, x_t^2, \dots, x_t^i, \dots, x_t^M]^T$  表示输入状态向量,包含M个特征或变量。

[0065] S104:对所述多元时间序列进行预处理,得到重构对象;

[0066] 在一些实施例中,对所述多元时间序列进行预处理,得到重构对象包括:

[0067] 对所述多元时间序列进行滑窗处理,得到多个窗口矩阵;

[0068] 对所述窗口矩阵进行PE编码,得到所述重构对象。

[0069] 为了提升Transformer模型的鲁棒能力,本实施例对多元时间序列进行滑窗处理,得到T个窗口矩阵:

[0070]  $\hat{X} = \{W_1, W_2, \dots, W_t, \dots, W_T\}$ ;

[0071] 式中,  $W_t = \{x_{t-K+1}, x_{t-K}, \dots, x_t\}^T$ ,  $t \in [1, T]$ ,其中,  $W_t$  为  $K \times M$  维的窗口矩阵, K为滑窗窗口长度。特殊地,对于  $W_t$  来说,当  $t < K$  时,用  $x_t$  来填充。

[0072] 由于滑窗处理会耗费大量的存储开销,且窗口长度越短所耗内存越大。对于实时流处理场景来说,滑窗处理会一定程度上影响在线实时检测效果,甚至会越来越滞后。另外,设备异常状态具有一定的持续性,因此可以根据异常检测的检测结果动态调整滑窗窗口长度K,以便能够较好地兼顾异常检测的实时性与Transformer模型的鲁棒性。

[0073] 由于Transformer模型没有类似RNN的循环结构,无法捕捉输入序列的位置信息,

为此本实施例采用PE (Positional Embedding, 位置编码) 编码对输入序列进行位置编码, 使得Transformer模型能够提取时序位置信息。PE编码格式为:

$$[0074] \quad P_{pos,i} = \begin{cases} \sin\left(pos / 10000^{2 \times \lfloor i/2 \rfloor / M}\right), & \text{if } i \text{ is even} \\ \cos\left(pos / 10000^{2 \times \lfloor i/2 \rfloor / M}\right), & \text{if } i \text{ is odd} \end{cases};$$

[0075] 式中, pos为时序位置,  $pos \in [1, K]$ ,  $i \in [0, M-1]$ ,  $\lfloor \cdot \rfloor$ 表示向下取整运算。经PE编码后, 得到位置矩阵 $P = \{P_{pos,i}\}_{K \times M}$ 即重构对象。

[0076] S105:通过训练好的所述Transformer模型处理所述重构对象, 得到重构结果。

[0077] 在一些实施例中, 所述通过训练好的所述Transformer模型处理所述重构对象, 得到重构结果包括:

[0078] 通过所述Transformer模型的所述编码器与所述解码器处理所述重构对象, 并对所述解码器的中间输出进行全连接处理与归一化处理得到所述重构结果; 其中, 通过所述编码器与所述解码器处理所述重构对象包括: 进行注意力计算。

[0079] 具体而言, 结合图2所示, 编码器包括L层级联的编码器单元, 编码器单元的处理过程如下:

$$[0080] \quad \mathbf{I}_1^1 = \text{LayerNorm}\left(\mathbf{I}_1 + \text{LSHAttention}(\mathbf{I}_1, \mathbf{I}_1, \mathbf{I}_1)\right);$$

$$[0081] \quad \mathbf{I}_1^2 = \text{LayerNorm}\left(\mathbf{I}_1^1 + \text{FFN}(\mathbf{I}_1^1)\right);$$

[0082] 式中,  $\text{LayerNorm}(\cdot)$ 表示残差连接与层归一化操作,  $\text{LSHAttention}(\cdot)$ 表示增强型局部敏感哈希注意力,  $\text{FFN}(\cdot)$ 表示前馈神经网络,  $\mathbf{I}_1^2$ 为编码器单元的中间输出,  $\mathbf{I}_1$ 为 $K \times M$ 维的编码器单元的输入矩阵, 也是前一层的编码器单元的中间输出。特殊地, 对于第一层编码器单元来说,  $\mathbf{I}_1 = \mathbf{I}_1 \oplus \mathbf{P}$ , 其中,  $\oplus$ 表示两个矩阵相同位置的元素相加。残差连接与层归一化操作旨在解决梯度消失与权重矩阵退化的问题。

[0083] 解码器包括L层级联的解码器单元, 解码器单元的处理过程如下:

$$[0084] \quad \mathbf{I}_2^1 = \text{LayerNorm}\left(\mathbf{I}_2 + \text{Mask}\left(\text{LSHAttention}(\mathbf{I}_2, \mathbf{I}_2, \mathbf{I}_2)\right)\right);$$

$$[0085] \quad \mathbf{I}_2^2 = \text{LayerNorm}\left(\mathbf{I}_2^1 + \text{LSHAttention}(\mathbf{I}_1^2, \mathbf{I}_1^2, \mathbf{I}_2^1)\right);$$

$$[0086] \quad \mathbf{d} = \text{LayerNorm}\left\{\text{FFN}(\mathbf{I}_2^2)\right\};$$

[0087] 式中,  $\text{Mask}(\cdot)$ 表示掩膜机制,  $\mathbf{d}$ 为解码器单元的中间输出,  $\mathbf{I}_2$ 为 $K \times M$ 维的解码器单元的输入矩阵, 也是前一层的解码器单元的中间输出。特殊地, 对于第一层解码器单元来说,  $\mathbf{I}_2 = (\mathbf{O}_{t-1} \odot \mathbf{O}_{t-1}) \oplus \mathbf{P}$ , 其中,  $\mathbf{O}_{t-1}$ 表示在第 $t-1$ 个时间步解码器的重构输出矩阵。 $\text{Mask}$ 掩膜的作用主要是屏蔽掉无效的padding区域。

[0088] 经过解码器处理后解码器的输出, 送入全连接层、Softmax层, 得到重构矩阵:

$$[0089] \quad \mathbf{O}_t = \text{Softmax}\{\text{Linear}(\mathbf{d})\};$$

[0090] 式中,  $\text{Linear}\{\cdot\}$ 表示全连接处理, 经过 $\text{Softmax}\{\cdot\}$ 归一化处理后概率最高的会被输出,  $\mathbf{O}_t = \{o_{t-K+1}, o_{t-K}, \dots, o_t\}^T$ 表示在第 $t$ 个时间步解码器的重构输出, 其中

$$\mathbf{o}_i = [O_i^1, O_i^2, \dots, O_i^i, \dots, O_i^M]^T。$$

[0091] 在一些实施例中,进行注意力计算包括:

[0092] 构建不同的哈希函数,并利用所述哈希函数对query与key进行编码,得到哈希值,根据所述哈希值对query与key进行分桶,得到最优分桶方案;

[0093] 根据所述最优分桶方案确定候选集;

[0094] 计算查询数据与所述候选集中各数据的相似度,并返回最近的多个近邻;

[0095] 在桶内对所述query与所述近邻进行点积计算,得到近似注意力。

[0096] 具体而言,传统注意力机制需要计算Q与K之间的所有相关性,所得到的注意力分配矩阵非常稀疏。在注意力计算过程中,由于经过点积计算 $QK^T$ 后,会使得不相关的特征向量相乘后的值很小,再经过Softmax函数处理,结果近似为0,即当前时间步与这些数据之间的时间依赖性完全可以忽略掉。也就是说,对于每个query,仅需关注离它“比较近”的几个key即可。为此,本实施例提出了一种增强型局部敏感哈希注意力机制,使相关度较高的数据落入同一个桶中去,从而在进行最近邻搜索时,仅需在一个桶或临近桶内进行搜索,可以有效降低多头自注意力的计算开销,极大地提升了计算效率。主要步骤包括:

[0097] 步骤1:确定每张哈希表内哈希函数的个数 $N_{\text{hash-func}}$ 、每个哈希函数分桶的个数 $N_{\text{hash-bucket}}$ ,以及哈希函数中的设计参数 $(d_1, d_2, p_1, p_2)$ 。其中,哈希函数需要满足以下条件:1、若 $d(x_1, x_2) \leq d_1$ ,则 $h(x_1) = h(x_2)$ 的概率至少为 $p_1$ ;2、若 $d(x_1, x_2) \geq d_2$ ,则 $h(x_1) = h(x_2)$ 的概率至多为 $p_2$ 。

[0098] 步骤2:构造多个不同的哈希函数,以实现多维度分桶操作,即构造 $N_{\text{hash-func}}$ 个哈希函数,分别对某query与key进行编码,得到 $N_{\text{hash-func}}$ 对哈希值。若其中超过一定比例的哈希值是相同的,则将该query与key分到同一个桶中去;

[0099] 步骤3:重复步骤2,完成多轮分桶任务,筛选出最优分桶方案;

[0100] 步骤4:对最优分桶方案进行排序,使得注意力权重集中在对角线上,再进行块分割,划分出不同的批量,并完成并行处理;

[0101] 步骤5:将待查询数据经过 $N_{\text{hash-func}}$ 个哈希函数得到各个相应的桶号,再根据“与”、“或”逻辑对桶进行合并操作,获得最终的候选集;

[0102] 步骤6:采用距离方法计算查询数据与候选集中各数据之间的相似度,并返回最近的 $K$ 个近邻;

[0103] 步骤7:最后在桶内执行点积计算,获得近似注意力。

[0104] 重构得到重构结果后,可以计算重构误差,即计算原始输入与重构输出之间的偏差,并可以根据重构误差以及预先定义的计算方式,计算得到异常分数。通常异常分数与重构误差呈正相关。或者还可以直接以重构误差作为异常分数。异常分数越大,表明工业设备异常的可能性越大。当异常分数超出异常阈值时,可以判定工业设备异常,由此快速可靠的检测出工业设备的潜在异常。

[0105] 综上所述,本申请所提供的工业设备的时序状态数据重构方法,提取的是存疑数据不包含正常数据,并对基于存疑数据构建的多元时间序列进行数据重构,如此在数据重构过程中可以减少大量的不必要的计算,从而加快数据重构速度。另外,本申请构建并训练多层的Transformer模型,利用训练好的多层的Transformer模型进行数据重构,能够提高

数据重构的可靠性,有利于进一步基于重构数据进行工业设备异常检测,提升工业设备异常检测的效率。

[0106] 本申请还提供了一种工业设备的时序状态数据重构装置,下文描述的该装置可以与上文描述的方法相互对应参照。请参考图3,图3为本申请实施例所提供的一种工业设备的时序状态数据重构装置的示意图,结合图3所示,该装置包括:

[0107] 构建模块10,用于构建Transformer模型;所述Transformer模型的编码器包括多层编码器单元,所述Transformer模型的解码器包括多层解码器单元;

[0108] 训练模块20,用于训练所述Transformer模型;

[0109] 提取模块30,用于提取工业设备的存疑数据,并根据所述存疑数据构建多元时间序列;

[0110] 预处理模块40,用于对所述多元时间序列进行预处理,得到重构对象;

[0111] 重构模块50,用于通过训练好的所述Transformer模型处理所述重构对象,得到重构结果。

[0112] 在上述实施例的基础上,作为一种具体的实施方式,训练模块20具体用于:

[0113] 采用元学习算法训练所述Transformer模型。

[0114] 在上述实施例的基础上,作为一种具体的实施方式,训练模块20包括:

[0115] 初始化单元,用于随机初始化所述Transformer模型的权重参数;

[0116] 第一滑窗处理单元,用于载入训练序列并对所述训练序列进行滑窗处理,得到多个窗口矩阵;

[0117] 更新单元,用于处理所述窗口矩阵,更新各个所述权重参数。

[0118] 在上述实施例的基础上,作为一种具体的实施方式,更新单元具体用于:

[0119] 根据  $\phi = \phi - \alpha \nabla_{\phi} \sum_{i=1}^T \mathcal{L}_i(f_{\phi})$  更新所述权重参数;  $\phi$  表示所述权重参数,  $\alpha$  表示元学

习的步长,  $\nabla_{\phi}$  表示梯度算子,  $\mathcal{L}_i$  表示代价函数,  $f_{\phi}$  表示网络模型,  $T$  表示所述窗口矩阵的个数。

[0120] 在上述实施例的基础上,作为一种具体的实施方式,预处理模块40包括:

[0121] 第二滑窗处理单元,用于对所述多元时间序列进行滑窗处理,得到多个窗口矩阵;

[0122] 编码单元,用于对所述窗口矩阵进行PE编码,得到所述重构对象。

[0123] 在上述实施例的基础上,作为一种具体的实施方式,重构模块50具体用于:

[0124] 通过所述Transformer模型的所述编码器与所述解码器处理所述重构对象,并对所述解码器的中间输出进行全连接处理与归一化处理得到所述重构结果;其中,通过所述编码器与所述解码器处理所述重构对象包括:进行注意力计算。

[0125] 在上述实施例的基础上,作为一种具体的实施方式,重构模块50具体用于:

[0126] 构建不同的哈希函数,并利用所述哈希函数对query与key进行编码,得到哈希值,根据所述哈希值对query与key进行分桶,得到最优分桶方案;

[0127] 根据所述最优分桶方案确定候选集;

[0128] 计算查询数据与所述候选集中各数据的相似度,并返回最近的多个近邻;

[0129] 在桶内对所述query与所述近邻进行点积计算,得到近似注意力。

[0130] 本申请还提供了一种工业设备的时序状态数据重构设备,参考图4所示,该设备包括存储器1和处理器2。

[0131] 存储器1,用于存储计算机程序;

[0132] 处理器2,用于执行计算机程序实现如下的步骤:

[0133] 构建Transformer模型;所述Transformer模型的编码器包括多层编码器单元,所述Transformer模型的解码器包括多层解码器单元;训练所述Transformer模型;提取工业设备的存疑数据,并根据所述存疑数据构建多元时间序列;对所述多元时间序列进行预处理,得到重构对象;通过训练好的所述Transformer模型处理所述重构对象,得到重构结果。

[0134] 对于本申请所提供的设备的介绍请参照上述方法实施例,本申请在此不做赘述。

[0135] 本申请还提供了一种计算机可读存储介质,该计算机可读存储介质上存储有计算机程序,计算机程序被处理器执行时可实现如下的步骤:

[0136] 构建Transformer模型;所述Transformer模型的编码器包括多层编码器单元,所述Transformer模型的解码器包括多层解码器单元;训练所述Transformer模型;提取工业设备的存疑数据,并根据所述存疑数据构建多元时间序列;对所述多元时间序列进行预处理,得到重构对象;通过训练好的所述Transformer模型处理所述重构对象,得到重构结果。

[0137] 该计算机可读存储介质可以包括:U盘、移动硬盘、只读存储器(Read-Only Memory,ROM)、随机存取存储器(Random Access Memory, RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0138] 对于本申请所提供的计算机可读存储介质的介绍请参照上述方法实施例,本申请在此不做赘述。

[0139] 说明书中各个实施例采用递进的方式描述,每个实施例重点说明的都是与其他实施例的不同之处,各个实施例之间相同相似部分互相参见即可。对于实施例公开的装置、设备以及计算机可读存储介质而言,由于其与实施例公开的方法相对应,所以描述的比较简单,相关之处参见方法部分说明即可。

[0140] 专业人员还可以进一步意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、计算机软件或者二者的结合来实现,为了清楚地说明硬件和软件的可互换性,在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本申请的范围。

[0141] 结合本文中所公开的实施例描述的方法或算法的步骤可以直接用硬件、处理器执行的软件模块,或者二者的结合来实施。软件模块可以置于随机存储器(RAM)、内存、只读存储器(ROM)、电可编程ROM、电可擦除可编程ROM、寄存器、硬盘、可移动磁盘、CD-ROM、或技术领域内所公知的任意其它形式的存储介质中。

[0142] 以上对本申请所提供的工业设备的时序状态数据重构方法、装置、设备以及计算机可读存储介质进行了详细介绍。本文中应用了具体个例对本申请的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本申请的方法及其核心思想。应当指出,对于本技术领域的普通技术人员来说,在不脱离本申请原理的前提下,还可以对本申请进行若干改进和修饰,这些改进和修饰也落入本申请权利要求的保护范围。

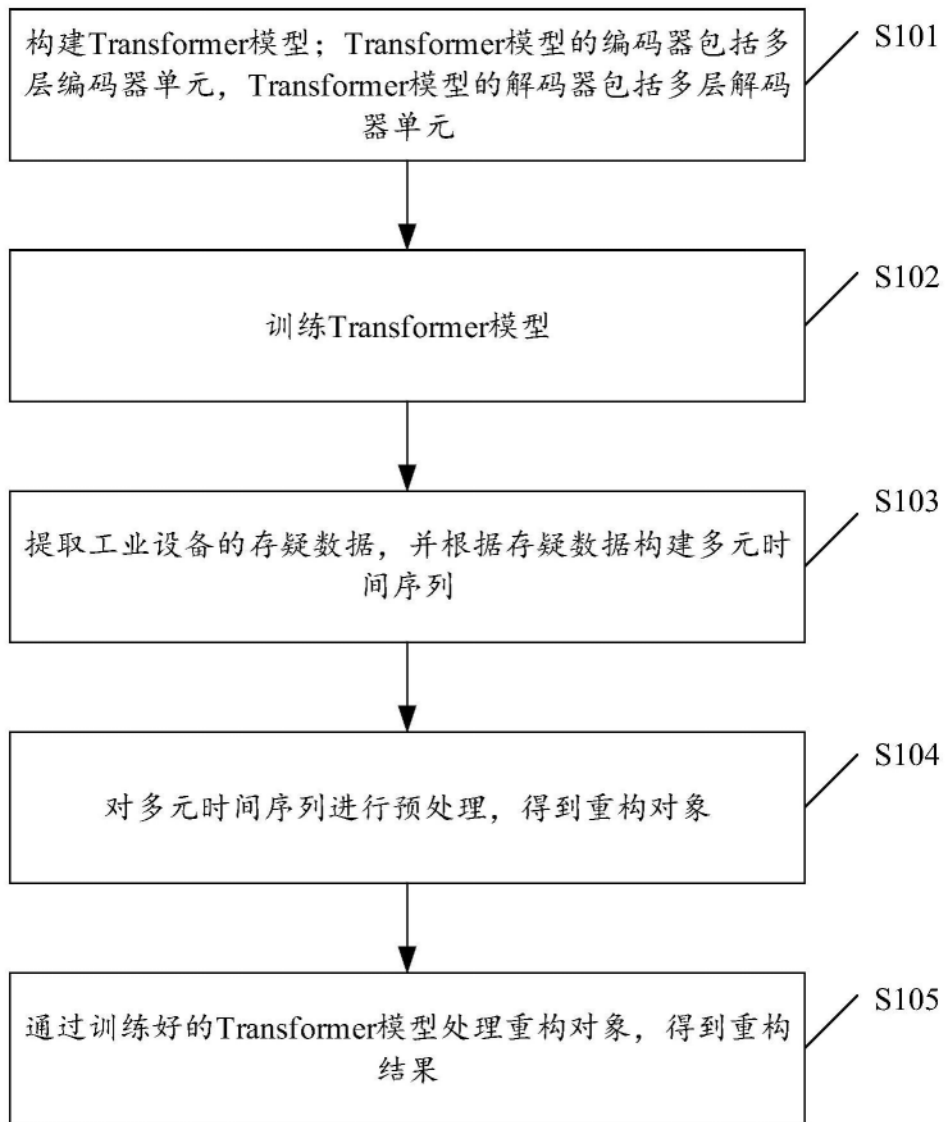


图1

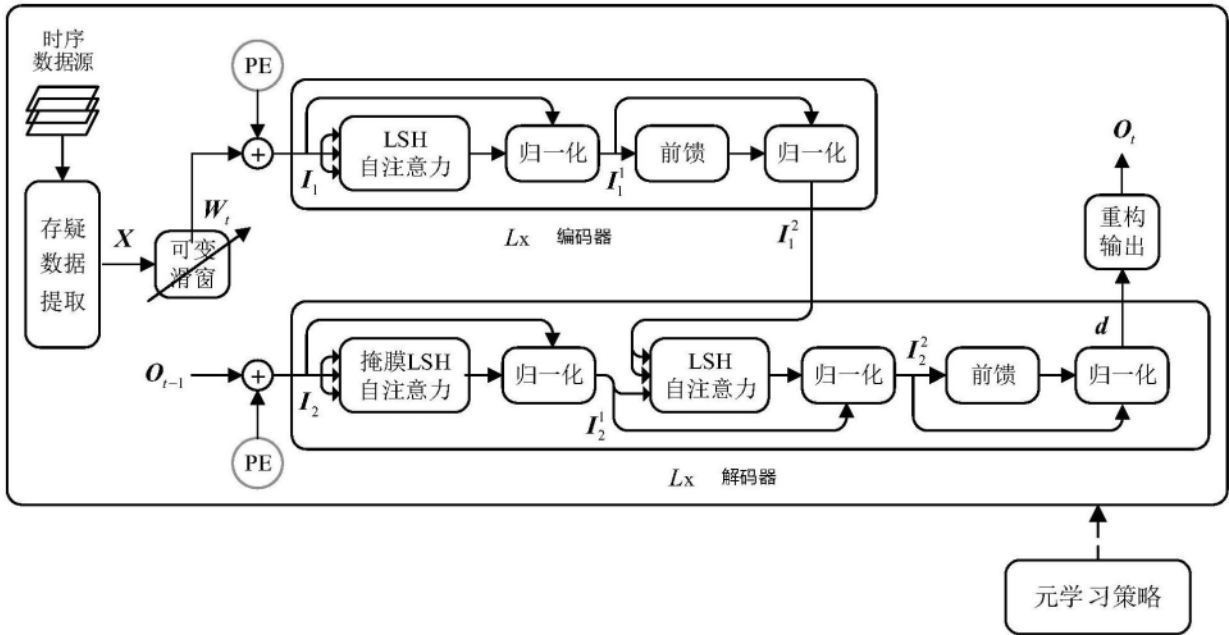


图2

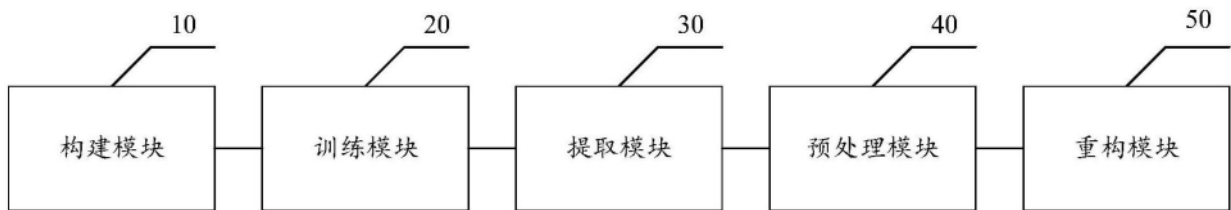


图3

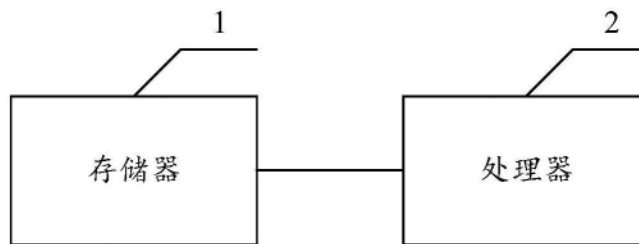


图4