



(12)发明专利申请

(10)申请公布号 CN 111679959 A
(43)申请公布日 2020.09.18

(21)申请号 202010328656.1

(22)申请日 2020.04.23

(71)申请人 平安科技(深圳)有限公司

地址 518000 广东省深圳市福田区福田街
道福安社区益田路5033号平安金融中
心23楼

(72)发明人 徐锐杰

(74)专利代理机构 深圳市赛恩倍吉知识产权代
理有限公司 44334

代理人 刘丽华 孙芬

(51)Int.Cl.

G06F 11/34(2006.01)

G06N 3/04(2006.01)

G06N 3/08(2006.01)

G06K 9/62(2006.01)

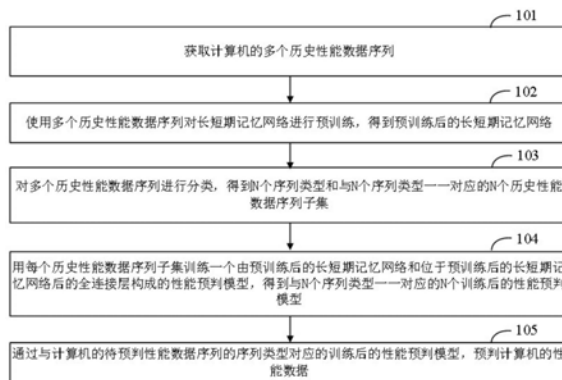
权利要求书2页 说明书12页 附图2页

(54)发明名称

计算机性能数据确定方法、装置、计算机设备
及存储介质

(57)摘要

本发明涉及区块链技术,提供一种计算机性能数据确定方法及相关设备,所述方法获取计算机的多个历史性能数据序列;使用多个历史性能数据序列对长短期记忆网络进行预训练;对多个历史性能数据序列进行分类,得到N个序列类型和与所述N个序列类型一一对应的N个历史性能数据序列子集;用每个历史性能数据序列子集训练一个由预训练后的长短期记忆网络和位于预训练后的长短期记忆网络后的全连接层构成的性能预判模型,得到与N个序列类型一一对应的N个训练后的性能预判模型;通过与计算机的待预判性能数据序列的序列类型对应的训练后的性能预判模型,预判计算机的性能数据。此外,本申请还涉及区块链技术,所述性能数据可存储于区块链中。



1. 一种计算机性能数据确定方法,其特征在于,所述方法包括:
获取计算机的多个历史性能数据序列;
使用所述多个历史性能数据序列对长短期记忆网络进行预训练,得到预训练后的长短期记忆网络;
对所述多个历史性能数据序列进行分类,得到N个序列类型和与所述N个序列类型一一对应的N个历史性能数据序列子集;
用每个历史性能数据序列子集训练一个由所述预训练后的长短期记忆网络和位于所述预训练后的长短期记忆网络后的全连接层构成的性能预判模型,得到与所述N个序列类型一一对应的N个训练后的性能预判模型;
通过与所述计算机的待预判性能数据序列的序列类型对应的训练后的性能预判模型,预判所述计算机的性能数据。
2. 如权利要求1所述的方法,其特征在于,所述获取计算机的多个历史性能数据序列包括:
获取多个计算机的历史性能数据主序列,历史性能数据主序列中的每个元素为计算机性能的状态向量;
以预设长度h为滑窗长度,以1为步长,从每个历史性能数据主序列中截取多个历史性能数据序列。
3. 如权利要求1所述的方法,其特征在于,所述对所述多个历史性能数据序列进行分类包括:
对所述多个历史性能数据序列进行聚类,根据接收的修改指令对聚类结果进行修改;
或
用训练好的预设卷积神经网络模型对所述多个历史性能数据序列进行分类。
4. 如权利要求1所述的方法,其特征在于,所述用每个历史性能数据序列子集训练一个由所述预训练后的长短期记忆网络和位于所述预训练后的长短期记忆网络后的全连接层构成的性能预判模型包括:
判断该历史性能数据序列的数量是否大于预设阈值;
若该历史性能数据序列的数量大于预设阈值,根据损失函数用该历史性能数据序列子集优化该历史性能数据序列子集对应的性能预判模型的参数;
若该历史性能数据序列的数量不大于预设阈值,根据损失函数用该历史性能数据序列子集优化该历史性能数据序列子集对应的性能预判模型的全连接层的参数。
5. 如权利要求1所述的方法,其特征在于,在所述通过与所述计算机的待预判性能数据序列的序列类型对应的训练后的性能预判模型,预判所述计算机的性能数据之前,所述方法还包括:
根据N个历史性能数据序列子集确定所述待预判性能数据序列的序列类型。
6. 如权利要求5所述的方法,其特征在于,所述根据N个历史性能数据序列子集确定所述待预判性能数据序列的序列类型包括:
计算N个历史性能数据序列子集的N个中心序列,将距离所述待预判性能数据序列最近的中心序列对应的序列类型确定为所述目标序列类型;或
用N个历史性能数据序列子集训练预设神经网络,其中每个历史性能数据序列子集中

的每个历史性能数据序列的标签为该历史性能数据序列的序列类型,将所述待预判性能数据序列输入训练后的所述预设神经网络,根据训练后的所述预设神经网络的输出确定所述待预判性能数据序列的序列类型。

7. 如权利要求1-5中任一项所述的方法,其特征在于,所述方法还包括:

若所述计算机的性能数据预测结果不在预设正常状态范围,返回计算机性能异常提醒。

8. 一种计算机性能数据确定装置,其特征在于,所述装置包括:

获取模块,用于获取计算机的多个历史性能数据序列;

第一训练模块,用于使用所述多个历史性能数据序列对长短期记忆网络进行预训练,得到预训练后的长短期记忆网络;

分类模块,用于对所述多个历史性能数据序列进行分类,得到N个序列类型和与所述N个序列类型一一对应的N个历史性能数据序列子集;

第二训练模块,用于用每个历史性能数据序列子集训练一个由所述预训练后的长短期记忆网络和位于所述预训练后的长短期记忆网络后的全连接层构成的性能预判模型,得到与所述N个序列类型一一对应的N个训练后的性能预判模型;

预判模块,用于通过与所述计算机的待预判性能数据序列的序列类型对应的训练后的性能预判模型,预判所述计算机的性能数据。

9. 一种计算机设备,其特征在于,所述计算机设备包括处理器,所述处理器用于执行存储器中存储的计算机程序以实现如权利要求1-7中任一项所述计算机性能数据确定方法。

10. 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,其特征在于,所述计算机程序被处理器执行时实现如权利要求1-7中任一项所述计算机性能数据确定方法。

计算机性能数据确定方法、装置、计算机设备及存储介质

技术领域

[0001] 本发明涉及区块链技术,具体涉及一种计算机性能数据确定方法、装置、计算机设备及计算机可读存储介质。

背景技术

[0002] 在云计算平台服务厂商,性能(如容量)预测是一个重要的AIOPS(基于人工智能的IT运营)应用场景。现有的性能预测方法对云计算平台服务厂商进行性能预测的精准度不高,对长期的性能预测效果不好。当存在大规模性能指标需要进行预测时,如果对每个性能指标使用一个模型进行预测,同时不同的性能指标需要调节模型的参数大不相同,导致需要的人力投入成本巨大。另一方面,当某一个性能指标的训练数据较少时,对性能指标的预测可能出现过拟合。

发明内容

[0003] 鉴于以上内容,有必要提出一种计算机性能数据确定方法、装置、计算机设备及计算机可读存储介质,其可以根据计算机的历史性能数据预判计算机性能。

[0004] 本申请的第一方面提供一种计算机性能数据确定方法,所述方法包括:

[0005] 获取计算机的多个历史性能数据序列;

[0006] 使用所述多个历史性能数据序列对长短期记忆网络进行预训练,得到预训练后的长短期记忆网络;

[0007] 对所述多个历史性能数据序列进行分类,得到N个序列类型和与所述N个序列类型一一对应的N个历史性能数据序列子集;

[0008] 用每个历史性能数据序列子集训练一个由所述预训练后的长短期记忆网络和位于所述预训练后的长短期记忆网络后的全连接层构成的性能预判模型,得到与所述N个序列类型一一对应的N个训练后的性能预判模型;

[0009] 通过与所述计算机的待预判性能数据序列的序列类型对应的训练后的性能预判模型,预判所述计算机的性能数据。

[0010] 本申请的第二方面提供一种计算机性能数据确定装置,所述装置包括:

[0011] 获取模块,用于获取计算机的多个历史性能数据序列;

[0012] 第一训练模块,用于使用所述多个历史性能数据序列对长短期记忆网络进行预训练,得到预训练后的长短期记忆网络;

[0013] 分类模块,用于对所述多个历史性能数据序列进行分类,得到N个序列类型和与所述N个序列类型一一对应的N个历史性能数据序列子集;

[0014] 第二训练模块,用于用每个历史性能数据序列子集训练一个由所述预训练后的长短期记忆网络和位于所述预训练后的长短期记忆网络后的全连接层构成的性能预判模型,得到与所述N个序列类型一一对应的N个训练后的性能预判模型;

[0015] 预判模块,用于通过与所述计算机的待预判性能数据序列的序列类型对应的训练

后的性能预判模型,预判所述计算机的性能数据。

[0016] 本申请的第三方面提供一种计算机设备,所述计算机设备包括处理器,所述处理器用于执行存储器中存储的计算机程序时实现所述计算机性能数据确定方法。

[0017] 本申请的第四方面提供一种计算机可读存储介质,其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所述计算机性能数据确定方法。

[0018] 本发明获取计算机的多个历史性能数据序列;使用所述多个历史性能数据序列对长短期记忆网络进行预训练,得到预训练后的长短期记忆网络;对所述多个历史性能数据序列进行分类,得到N个序列类型和与所述N个序列类型一一对应的N个历史性能数据序列子集;用每个历史性能数据序列子集训练一个由所述预训练后的长短期记忆网络和位于所述预训练后的长短期记忆网络后的全连接层构成的性能预判模型,得到与所述N个序列类型一一对应的N个训练后的性能预判模型;通过与所述计算机的待预判性能数据序列的序列类型对应的训练后的性能预判模型,预判所述计算机的性能数据。本发明实现了根据计算机的历史性能数据预判计算机性能。

附图说明

[0019] 图1是本发明实施例提供的计算机性能数据确定方法的流程图。

[0020] 图2是本发明实施例提供的计算机性能数据确定装置的结构图。

[0021] 图3是本发明实施例提供的计算机设备的示意图。

具体实施方式

[0022] 为了能够更清楚地理解本发明的上述目的、特征和优点,下面结合附图和具体实施例对本发明进行详细描述。需要说明的是,在不冲突的情况下,本申请的实施例及实施例中的特征可以相互组合。

[0023] 在下面的描述中阐述了很多具体细节以便于充分理解本发明,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,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,都属于本发明保护的范围。

[0024] 除非另有定义,本文所使用的所有的技术和科学术语与属于本发明的技术领域的技术人员通常理解的含义相同。本文中在本发明的说明书中所使用的术语只是为了描述具体的实施例的目的,不是旨在于限制本发明。

[0025] 优选地,本发明的计算机性能数据确定方法应用在一个或者多个计算机设备中。所述计算机设备是一种能够按照事先设定或存储的指令,自动进行数值计算和/或信息处理的设备,其硬件包括但不限于微处理器、专用集成电路(Application Specific Integrated Circuit,ASIC)、可编程门阵列(Field-Programmable Gate Array,FPGA)、数字处理器(Digital Signal Processor,DSP)、嵌入式设备等。

[0026] 所述计算机设备可以是桌上型计算机、笔记本、掌上电脑及云端服务器等计算设备。所述计算机设备可以与用户通过键盘、鼠标、遥控器、触摸板或声控设备等方式进行人机交互。

[0027] 实施例一

[0028] 图1是本发明实施例一提供的计算机性能数据确定方法的流程图。所述计算机性

能数据确定方法应用于计算机设备,用于根据计算机的历史性能数据预判计算机性能数据,计算机性能数据可以包括计算机CPU数据、计算机GPU数据、计算机内存数据或计算机存储数据。

[0029] 如图1所示,所述计算机性能数据确定方法包括:

[0030] 101,获取计算机的多个历史性能数据序列。

[0031] 在一具体实施例中,所述获取计算机的多个历史性能数据序列包括:

[0032] 获取多个计算机的历史性能数据主序列,历史性能数据主序列中的每个元素为计算机性能的状态向量;

[0033] 以预设长度 h 为滑窗长度,以1为步长,从每个历史性能数据主序列中截取多个历史性能数据序列。

[0034] 可以将每个历史性能数据序列所在的历史性能数据主序列中该历史性能数据序列的下一元素标注为该历史性能数据序列的标签。例如,一个计算机的历史性能数据主序列为 $\{X_1, X_2, \dots, X_i, \dots, X_n\}$,其中, X_i 为时间点 i 的CPU的状态向量;以预设长度 h 为滑窗长度、以1为步长从每个历史性能数据主序列中截取多个历史性能数据序列为 $\{X_1, X_2, \dots, X_h\}$ 、 $\{X_2, X_3, \dots, X_{h+1}\}$ 、 $\{X_3, X_4, \dots, X_{h+2}\}$ 等;历史性能数据序列 $\{X_1, X_2, \dots, X_h\}$ 、 $\{X_2, X_3, \dots, X_{h+1}\}$ 、 $\{X_3, X_4, \dots, X_{h+2}\}$ 的标签分别为 X_{h+1} 、 X_{h+2} 、 X_{h+3} 。

[0035] 每个历史性能数据序列都是一个时间序列。

[0036] 102,使用所述多个历史性能数据序列对长短期记忆网络进行预训练,得到预训练后的长短期记忆网络。

[0037] 具体地,可以初始化所述长短期记忆网络中的参数,将每个历史性能数据序列输入所述长短期记忆网络,计算所述长短期记忆网络根据输入值计算出的输出值与该历史性能数据序列的标签的距离,根据距离优化所述长短期记忆网络中的参数。

[0038] 在另一实施例中,在使用所述多个历史性能数据序列对长短期记忆网络进行预训练之前,所述方法还包括:

[0039] 对所述多个历史性能数据序列进行预处理。

[0040] 可以对所述多个历史性能数据序列进行缺失值预处理、异常数据预处理等。对所述历史性能数据序列集进行缺失值预处理可以使所述多个历史性能序列中不存在缺失值,减小对训练长短期记忆网络的影响。对所述多个历史性能数据序列进行异常数据预处理,增加长短期记忆网络的训练收敛速度,减少训练长短期记忆网络的时间。当一个历史性能数据序列存在缺失值时(由于数据来源错误或软件设计错误等导致),可以采用插值法或替换法对该历史性能数据序列进行缺失值预处理。当一个历史性能数据序列存在异常数据时(即该历史性能数据序列中包括不在预设范围内的值),若异常数据大于预设范围的最大值,则将异常数据设置为预设范围的最大值;若异常数据小于预设范围的最小值,则将异常数据设置为预设范围的最小值。

[0041] 103,对所述多个历史性能数据序列进行分类,得到 N 个序列类型和与所述 N 个序列类型一一对应的 N 个历史性能数据序列子集。

[0042] 在一具体实施例中,所述对所述多个历史性能数据序列进行分类包括:

[0043] 对所述多个历史性能数据序列进行聚类,根据接收的修改指令对聚类结果进行修改;或

- [0044] 用训练好的预设卷积神经网络模型对所述多个历史性能数据序列进行分类。
- [0045] 可以根据DBScan聚类算法对所述历史性能数据序列集中的多个历史性能数据序列进行聚类,其中,DBScan聚类算法中的距离函数使用一维NCC(normalized cross correlation)算法;当聚类结果出现异常时,根据接收的删除指令删除聚类结果中距离聚类中心超出预设范围的历史性能数据序列。
- [0046] 具体地,所述对所述多个历史性能数据序列进行聚类包括:
- [0047] 从所述多个历史性能数据序列中选择N个中心点序列;
- [0048] 计算每个历史性能数据序列与所述N个中心点序列的距离,得到与该历史性能数据序列距离最近的中心点序列;
- [0049] 将该历史性能数据序列分划至,与该历史性能数据序列距离最近的中心点序列所属的簇。
- [0050] 具体地,所述对所述多个历史性能数据序列进行聚类包括:
- [0051] 以每个历史性能数据序列为中心确定一个簇,得到W个簇;
- [0052] 若W大于N,计算每两个簇间的距离,合并距离最小的两个簇,得到W-1个簇;
- [0053] 若W-1大于N,计算每两个簇间的距离,合并距离最小的两个簇,得到W-2个簇;依次类推,当W-R等于N时,得到N个簇。
- [0054] 在所述用训练好的预设卷积神经网络模型对所述多个历史性能数据序列进行分类之前,训练预设卷积神经网络模型:
- [0055] 将一个历史性能数据序列输入预设卷积神经网络模型,每个历史性能数据序列对应一个类型标签;
- [0056] 基于预设卷积神经网络模型的输出和该历史性能数据序列的类型标签,对预设卷积神经网络模型进行反向传播,优化预设卷积神经网络模型的参数。
- [0057] 104,用每个历史性能数据序列子集训练一个由所述预训练后的长短期记忆网络和位于所述预训练后的长短期记忆网络后的全连接层构成的性能预判模型,得到与所述N个序列类型一一对应的N个训练后的性能预判模型。
- [0058] 在一具体实施例中,所述用每个历史性能数据序列子集训练一个由所述预训练后的长短期记忆网络和位于所述预训练后的长短期记忆网络后的全连接层构成的性能预判模型包括:
- [0059] 判断该历史性能数据序列的数量是否大于预设阈值;
- [0060] 若该历史性能数据序列的数量大于预设阈值,根据损失函数用该历史性能数据序列子集优化该历史性能数据序列子集对应的性能预判模型的参数;
- [0061] 若该历史性能数据序列的数量不大于预设阈值,根据损失函数用该历史性能数据序列子集优化该历史性能数据序列子集对应的性能预判模型的全连接层的参数。
- [0062] 当该历史性能数据序列的数量不大于预设阈值时,只优化该历史性能数据序列子集对应的性能预判模型的全连接层的参数可以减少出现过拟合的情况,且由于所述性能预判模型经过预训练,不会导致预测准确率低的情况发生。
- [0063] 在另一实施例中,所述用每个历史性能数据序列子集训练一个由所述预训练后的长短期记忆网络和位于所述预训练后的长短期记忆网络后的全连接层构成的性能预判模型包括:

[0064] 根据损失函数用该历史性能数据序列子集优化该历史性能数据序列子集对应的性能预判模型的参数。

[0065] 105,通过与所述计算机的待预判性能数据序列的序列类型对应的训练后的性能预判模型,预判所述计算机的性能数据。

[0066] 在所述通过与所述计算机的待预判性能数据序列的序列类型对应的训练后的性能预判模型,预判所述计算机的性能数据之前,所述方法还包括:

[0067] (1) 获取所述计算机的待预判性能数据序列。

[0068] 例如,计算机CPU性能序列为 $\{X_1, X_2, \dots, X_{15}\}$,其中, $X_1 = (2019, 06, 12, 13, 15, 43)$,前5维为获取时间,第6维为所述计算机在获取时间的CPU使用率; $X_2 = (2019, 06, 12, 13, 16, 80)$; $X_{15} = (2019, 06, 12, 13, 30, 90)$ 。类似地,所述计算机的待预判性能数据序列还包括计算机GPU性能、计算机内存性能或计算机存储性能。

[0069] (2) 根据N个历史性能数据序列子集确定所述待预判性能数据序列的序列类型。

[0070] 在一具体实施例中,所述根据N个历史性能数据序列子集确定所述待预判性能数据序列的序列类型包括:

[0071] 计算N个历史性能数据序列子集的N个中心序列,将距离所述待预判性能数据序列最近的中心序列对应的序列类型确定为所述目标序列类型;或

[0072] 用N个历史性能数据序列子集训练预设神经网络,其中每个历史性能数据序列子集中的每个历史性能数据序列的标签为该历史性能数据序列的序列类型,将所述待预判性能数据序列输入训练后的所述预设神经网络,根据训练后的所述预设神经网络的输出确定所述待预判性能数据序列的序列类型。

[0073] 在训练所述预设神经网络时,可以将历史性能数据序列作为所述预设神经网络的输入,将历史性能数据序列的序列类型作为历史性能数据序列的标签;根据所述预设神经网络的输出和历史性能数据序列的标签通过反向传播算法优化所述预设神经网络的参数。

[0074] 通过与所述计算机的待预判性能数据序列的序列类型对应的训练后的性能预判模型,预判所述计算机的性能数据,例如,计算机CPU性能序列为 $\{X_1, X_2, \dots, X_{15}\}$,其中, $X_1 = (2019, 06, 12, 13, 15, 43)$,前5维为获取时间,第6维为所述计算机在获取时间的CPU使用率; $X_2 = (2019, 06, 12, 13, 16, 80)$; $X_{15} = (2019, 06, 12, 13, 30, 90)$ 。计算机CPU性能序列对应性能预判模型A,性能预判模型A更能准确地提取计算机CPU性能序列的特征(因为训练过程中也根据序列类型对性能预判模型A进行训练)。将计算机CPU性能序列输入性能预判模型A,通过性能预测模型A预判所述计算机的CPU性能,得到 $X_{16} = (2019, 06, 12, 13, 31, 95)$ 。

[0075] 实施例一的计算机性能数据确定方法获取计算机的历史性能数据序列集,所述历史性能数据序列集包括多个历史性能数据序列和每个历史性能数据序列的标签;使用所述历史性能数据序列集对长短期记忆网络进行预训练,得到预训练后的长短期记忆网络;对所述多个历史性能数据序列进行分类,得到N个序列类型和与所述N个序列类型一一对应的N个历史性能数据序列子集;用每个历史性能数据序列子集训练一个由所述预训练后的长短期记忆网络和位于所述预训练后的长短期记忆网络后的全连接层构成的性能预判模型,得到与所述N个序列类型一一对应的N个性能预判模型;获取所述计算机的待预判性能数据序列;确定所述待预判性能数据序列的序列类型;通过与所述待预判性能数据序列的序列类型对应的性能预判模型预判所述计算机的性能数据。实施例一根据计算机的历史性能数

据预判计算机性能。

[0076] 需要强调的是,为进一步保证上述性能数据的私密和安全性,上述性能数据还可以存储于一区块链的节点中。

[0077] 在另一实施例中,所述方法还包括:

[0078] 若所述计算机的性能数据预测结果不在预设正常状态范围,返回计算机性能异常提醒。在异常发生时,停止运行所述计算机中的新任务。

[0079] 例如,所述计算机的性能数据预测结果为 $X_{16} = (2019, 06, 12, 13, 31, 95)$,95为预测的所述计算机的CPU使用率,预设正常状态范围为0-90,返回计算机CPU异常提醒给用户;并停止运行所述计算机中的新任务。

[0080] 在另一实施例中,所述方法还包括:

[0081] (1) 获取计算机的多个历史CPU使用率序列。

[0082] 例如,一个计算机的历史CPU使用率主序列为 $\{X_1, X_2, \dots, X_i, \dots, X_n\}$,其中, X_i 为时间点 i 的CPU的状态向量;以预设长度 h 为滑窗长度、以1为步长从每个历史CPU使用率主序列中截取多个历史CPU使用率序列为 $\{X_1, X_2, \dots, X_h\}$ 、 $\{X_2, X_3, \dots, X_{h+1}\}$ 、 $\{X_3, X_4, \dots, X_{h+2}\}$ 等;历史CPU使用率序列 $\{X_1, X_2, \dots, X_h\}$ 、 $\{X_2, X_3, \dots, X_{h+1}\}$ 、 $\{X_3, X_4, \dots, X_{h+2}\}$ 的标签分别为 X_{h+1} 、 X_{h+2} 、 X_{h+3} 。

[0083] 每个历史CPU使用率序列都是一个时间序列。

[0084] (2) 使用所述多个历史CPU使用率序列对长短期记忆网络进行预训练,得到预训练后的长短期记忆网络。

[0085] 具体地,可以初始化所述长短期记忆网络中的参数,将每个历史CPU使用率序列输入所述长短期记忆网络,计算所述长短期记忆网络根据输入值计算出的输出值与该历史CPU使用率序列的标签的距离,根据距离优化所述长短期记忆网络中的参数。

[0086] (3) 对所述多个历史CPU使用率序列进行分类,得到 N 个序列类型和与所述 N 个序列类型一一对应的 N 个历史CPU使用率序列子集。

[0087] 在一具体实施例中,所述对所述多个历史CPU使用率序列进行分类包括:

[0088] 对所述多个历史CPU使用率序列进行聚类,根据接收的修改指令对聚类结果进行修改;或

[0089] 用训练好的预设卷积神经网络模型对所述多个历史CPU使用率序列进行分类。

[0090] (4) 用每个历史CPU使用率序列子集训练一个由所述预训练后的长短期记忆网络和位于所述预训练后的长短期记忆网络后的全连接层构成的CPU使用率预判模型,得到与所述 N 个序列类型一一对应的 N 个训练后的CPU使用率预判模型。

[0091] (5) 通过与所述计算机的待预判CPU使用率序列的序列类型对应的训练后的CPU使用率预判模型,预判所述计算机的CPU性能。

[0092] 获取所述计算机的待预判CPU使用率序列。

[0093] 例如,计算机CPU性能序列为 $\{X_1, X_2, \dots, X_{15}\}$,其中, $X_1 = (2019, 06, 12, 13, 15, 43)$,前5维为获取时间,第6维为所述计算机在获取时间的CPU使用率; $X_2 = (2019, 06, 12, 13, 16, 80)$; $X_{15} = (2019, 06, 12, 13, 30, 90)$ 。

[0094] 根据 N 个历史CPU使用率序列子集确定所述待预判CPU使用率序列的序列类型。

[0095] 计算机CPU性能序列对应CPU使用率预判模型A。将计算机CPU性能序列输入CPU使

用率预判模型A,通过CPU使用率预判模型A预判所述计算机的CPU性能,得到 $X_{16} = (2019, 06, 12, 13, 31, 95)$ 。

[0096] 实施例二

[0097] 图2是本发明实施例二提供的计算机性能数据确定装置的结构图。所述计算机性能数据确定装置20应用于计算机设备。所述计算机性能数据确定装置20用于根据计算机的历史性能数据预判计算机性能数据,计算机性能数据可以包括计算机CPU数据、计算机GPU数据、计算机内存数据或计算机存储数据。

[0098] 如图2所示,所述计算机性能数据确定装置20可以包括获取模块201、第一训练模块202、分类模块203、第二训练模块204、预判模块205。

[0099] 获取模块201,用于获取计算机的多个历史性能数据序列。

[0100] 在一具体实施例中,所述获取计算机的多个历史性能数据序列包括:

[0101] 获取多个计算机的历史性能数据主序列,历史性能数据主序列中的每个元素为计算机性能的状态向量;

[0102] 以预设长度 h 为滑窗长度,以1为步长,从每个历史性能数据主序列中截取多个历史性能数据序列。

[0103] 可以将每个历史性能数据序列所在的历史性能数据主序列中该历史性能数据序列的下一元素标注为该历史性能数据序列的标签。例如,一个计算机的历史性能数据主序列为 $\{X_1, X_2, \dots, X_i, \dots, X_n\}$,其中, X_i 为时间点 i 的CPU的状态向量;以预设长度 h 为滑窗长度、以1为步长从每个历史性能数据主序列中截取多个历史性能数据序列为 $\{X_1, X_2, \dots, X_h\}$ 、 $\{X_2, X_3, \dots, X_{h+1}\}$ 、 $\{X_3, X_4, \dots, X_{h+2}\}$ 等;历史性能数据序列 $\{X_1, X_2, \dots, X_h\}$ 、 $\{X_2, X_3, \dots, X_{h+1}\}$ 、 $\{X_3, X_4, \dots, X_{h+2}\}$ 的标签分别为 X_{h+1} 、 X_{h+2} 、 X_{h+3} 。

[0104] 每个历史性能数据序列都是一个时间序列。

[0105] 第一训练模块202,用于使用所述多个历史性能数据序列对长短期记忆网络进行预训练,得到预训练后的长短期记忆网络。

[0106] 具体地,可以初始化所述长短期记忆网络中的参数,将每个历史性能数据序列输入所述长短期记忆网络,计算所述长短期记忆网络根据输入值计算出的输出值与该历史性能数据序列的标签的距离,根据距离优化所述长短期记忆网络中的参数。

[0107] 在另一实施例中,在使用所述多个历史性能数据序列对长短期记忆网络进行预训练之前,所述方法还包括:

[0108] 对所述多个历史性能数据序列进行预处理。

[0109] 可以对所述多个历史性能数据序列进行缺失值预处理、异常数据预处理等。对所述历史性能数据序列集进行缺失值预处理可以使所述多个历史性能序列中不存在缺失值,减小对训练长短期记忆网络的影响。对所述多个历史性能数据序列进行异常数据预处理,增加长短期记忆网络的训练收敛速度,减少训练长短期记忆网络的时间。当一个历史性能数据序列存在缺失值时(由于数据来源错误或软件设计错误等导致),可以采用插值法或替换法对该历史性能数据序列进行缺失值预处理。当一个历史性能数据序列存在异常数据时(即该历史性能数据序列中包括不在预设范围内的值),若异常数据大于预设范围的最大值,则将异常数据设置为预设范围的最大值;若异常数据小于预设范围的最小值,则将异常数据设置为预设范围的最小值。

- [0110] 分类模块203,用于对所述多个历史性能数据序列进行分类,得到N个序列类型和与所述N个序列类型一一对应的N个历史性能数据序列子集。
- [0111] 在一具体实施例中,所述对所述多个历史性能数据序列进行分类包括:
- [0112] 对所述多个历史性能数据序列进行聚类,根据接收的修改指令对聚类结果进行修改;或
- [0113] 用训练好的预设卷积神经网络模型对所述多个历史性能数据序列进行分类。
- [0114] 可以根据DBScan聚类算法对所述历史性能数据序列集中的多个历史性能数据序列进行聚类,其中,DBScan聚类算法中的距离函数使用一维NCC(normalized cross correlation)算法;当聚类结果出现异常时,根据接收的删除指令删除聚类结果中距离聚类中心超出预设范围的历史性能数据序列。
- [0115] 具体地,所述对所述多个历史性能数据序列进行聚类包括:
- [0116] 从所述多个历史性能数据序列中选择N个中心点序列;
- [0117] 计算每个历史性能数据序列与所述N个中心点序列的距离,得到与该历史性能数据序列距离最近的中心点序列;
- [0118] 将该历史性能数据序列分划至,与该历史性能数据序列距离最近的中心点序列所属的簇。
- [0119] 具体地,所述对所述多个历史性能数据序列进行聚类包括:
- [0120] 以每个历史性能数据序列为中心确定一个簇,得到W个簇;
- [0121] 若W大于N,计算每两个簇间的距离,合并距离最小的两个簇,得到W-1个簇;
- [0122] 若W-1大于N,计算每两个簇间的距离,合并距离最小的两个簇,得到W-2个簇;依次类推,当W-R等于N时,得到N个簇。
- [0123] 在所述用训练好的预设卷积神经网络模型对所述多个历史性能数据序列进行分类之前,训练预设卷积神经网络模型:
- [0124] 将一个历史性能数据序列输入预设卷积神经网络模型,每个历史性能数据序列对应一个类型标签;
- [0125] 基于预设卷积神经网络模型的输出和该历史性能数据序列的类型标签,对预设卷积神经网络模型进行反向传播,优化预设卷积神经网络模型的参数。
- [0126] 第二训练模块204,用于用每个历史性能数据序列子集训练一个由所述预训练后的长短期记忆网络和位于所述预训练后的长短期记忆网络后的全连接层构成的性能预判模型,得到与所述N个序列类型一一对应的N个训练后的性能预判模型。
- [0127] 在一具体实施例中,所述用每个历史性能数据序列子集训练一个由所述预训练后的长短期记忆网络和位于所述预训练后的长短期记忆网络后的全连接层构成的性能预判模型包括:
- [0128] 判断该历史性能数据序列的数量是否大于预设阈值;
- [0129] 若该历史性能数据序列的数量大于预设阈值,根据损失函数用该历史性能数据序列子集优化该历史性能数据序列子集对应的性能预判模型的参数;
- [0130] 若该历史性能数据序列的数量不大于预设阈值,根据损失函数用该历史性能数据序列子集优化该历史性能数据序列子集对应的性能预判模型的全连接层的参数。
- [0131] 当该历史性能数据序列的数量不大于预设阈值时,只优化该历史性能数据序列子

集对应的性能预判模型的全连接层的参数可以减少出现过拟合的情况,且由于所述性能预判模型经过预训练,不会导致预测准确率低的情况发生。

[0132] 在另一实施例中,所述用每个历史性能数据序列子集训练一个由所述预训练后的长短期记忆网络和位于所述预训练后的长短期记忆网络后的全连接层构成的性能预判模型包括:

[0133] 根据损失函数用该历史性能数据序列子集优化该历史性能数据序列子集对应的性能预判模型的参数。

[0134] 预判模块205,用于通过与所述计算机的待预判性能数据序列的序列类型对应的训练后的性能预判模型,预判所述计算机的性能数据。

[0135] 在所述通过与所述计算机的待预判性能数据序列的序列类型对应的训练后的性能预判模型,预判所述计算机的性能数据之前,所述方法还包括:

[0136] (1) 获取所述计算机的待预判性能数据序列。

[0137] 例如,计算机CPU性能序列为 $\{X_1, X_2, \dots, X_{15}\}$,其中, $X_1 = (2019, 06, 12, 13, 15, 43)$,前5维为获取时间,第6维为所述计算机在获取时间的CPU使用率; $X_2 = (2019, 06, 12, 13, 16, 80)$; $X_{15} = (2019, 06, 12, 13, 30, 90)$ 。类似地,所述计算机的待预判性能数据序列还包括计算机GPU性能、计算机内存性能或计算机存储性能。

[0138] (2) 根据N个历史性能数据序列子集确定所述待预判性能数据序列的序列类型。

[0139] 在一具体实施例中,所述根据N个历史性能数据序列子集确定所述待预判性能数据序列的序列类型包括:

[0140] 计算N个历史性能数据序列子集的N个中心序列,将距离所述待预判性能数据序列最近的中心序列对应的序列类型确定为所述目标序列类型;或

[0141] 用N个历史性能数据序列子集训练预设神经网络,其中每个历史性能数据序列子集中的每个历史性能数据序列的标签为该历史性能数据序列的序列类型,将所述待预判性能数据序列输入训练后的所述预设神经网络,根据训练后的所述预设神经网络的输出确定所述待预判性能数据序列的序列类型。

[0142] 在训练所述预设神经网络时,可以将历史性能数据序列作为所述预设神经网络的输入,将历史性能数据序列的序列类型作为历史性能数据序列的标签;根据所述预设神经网络的输出和历史性能数据序列的标签通过反向传播算法优化所述预设神经网络的参数。

[0143] 通过与所述计算机的待预判性能数据序列的序列类型对应的训练后的性能预判模型,预判所述计算机的性能数据,例如,计算机CPU性能序列为 $\{X_1, X_2, \dots, X_{15}\}$,其中, $X_1 = (2019, 06, 12, 13, 15, 43)$,前5维为获取时间,第6维为所述计算机在获取时间的CPU使用率; $X_2 = (2019, 06, 12, 13, 16, 80)$; $X_{15} = (2019, 06, 12, 13, 30, 90)$ 。计算机CPU性能序列对应性能预判模型A,性能预判模型A更能准确地提取计算机CPU性能序列的特征(因为训练过程中也根据序列类型对性能预判模型A进行训练)。将计算机CPU性能序列输入性能预判模型A,通过性能预测模型A预判所述计算机的CPU性能,得到 $X_{16} = (2019, 06, 12, 13, 31, 95)$ 。

[0144] 实施例二的计算机性能数据确定装置20根据计算机的历史性能数据预判计算机性能。

[0145] 在另一实施例中,所述计算机性能数据确定装置20还包括返回模块,用于若所述计算机的性能数据预测结果不在预设正常状态范围,返回计算机性能异常提醒。

[0146] 在异常发生时,停止运行所述计算机中的新任务。

[0147] 例如,所述计算机的性能数据预测结果为 $X_{16} = (2019, 06, 12, 13, 31, 95)$, 95为预测的所述计算机的CPU使用率,预设正常状态范围为0-90,返回计算机CPU异常提醒给用户;并停止运行所述计算机中的新任务。

[0148] 实施例三

[0149] 本实施例提供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,该计算机程序被处理器执行时实现上述计算机性能数据确定方法实施例中的步骤,例如图1所示的步骤101-105。

[0150] 或者,该计算机程序被处理器执行时实现上述装置实施例中各模块的功能,例如图2中的模块201-205。

[0151] 实施例四

[0152] 图3为本发明实施例三提供的计算机设备的示意图。所述计算机设备30包括存储器301、处理器302以及存储在所述存储器301中并可在所述处理器302上运行的计算机程序303,例如计算机性能数据确定程序。所述处理器302执行所述计算机程序303时实现上述计算机性能数据确定方法实施例中的步骤,例如图1所示的101-105。

[0153] 或者,该计算机程序被处理器执行时实现上述装置实施例中各模块的功能,例如图2中的模块201-205。

[0154] 示例性的,所述计算机程序303可以被分割成一个或多个模块,所述一个或者多个模块被存储在所述存储器301中,并由所述处理器302执行,以完成本方法。所述一个或多个模块可以是能够完成特定功能的一系列计算机程序指令段,该指令段用于描述所述计算机程序303在所述计算机设备30中的执行过程。例如,所述计算机程序303可以被分割成图2中的获取模块201、第一训练模块202、分类模块203、第二训练模块204、预判模块205,各模块具体功能参见实施例二。

[0155] 本领域技术人员可以理解,所述示意图3仅仅是计算机设备30的示例,并不构成对计算机设备30的限定,可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件,例如所述计算机设备30还可以包括输入输出设备、网络接入设备、总线等。

[0156] 所称处理器302可以是中央处理单元(Central Processing Unit,CPU),还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现场可编程门阵列(Field-Programmable Gate Array,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器302也可以是任何常规的处理器等,所述处理器302是所述计算机设备30的控制中心,利用各种接口和线路连接整个计算机设备30的各个部分。

[0157] 所述存储器301可用于存储所述计算机程序303,所述处理器302通过运行或执行存储在所述存储器301内的计算机程序或模块,以及调用存储在存储器301内的数据,实现所述计算机设备30的各种功能。所述存储器301可主要包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需的应用程序(比如声音播放功能、图像播放功能等等);存储数据区可存储根据计算机设备30的使用所创建的数据等。此外,存储器301可以包括非易失性存储器,例如硬盘、内存、插接式硬盘,智能存储卡(Smart Media

Card, SMC), 安全数字 (Secure Digital, SD) 卡, 闪存卡 (Flash Card)、至少一个磁盘存储器件、闪存器件、或其他非易失性固态存储器件。

[0158] 所述计算机设备30集成的模块如果以软件功能模块的形式实现并作为独立的产品销售或使用, 可以存储在一个计算机可读取存储介质中。基于这样的理解, 本发明实现上述实施例方法中的全部或部分流程, 也可以通过计算机程序来指令相关的硬件来完成, 所述的计算机程序可存储于一计算机可读存储介质中, 该计算机程序在被处理器执行时, 可实现上述各个方法实施例的步骤。其中, 所述计算机程序包括计算机程序代码, 所述计算机程序代码可以为源代码形式、对象代码形式、可执行文件或某些中间形式等。所述计算机可读介质可以包括: 能够携带所述计算机程序代码的任何实体或装置、记录介质、U盘、移动硬盘、磁碟、光盘、计算机存储器、只读存储器 (ROM, Read-Only Memory)。

[0159] 进一步地, 所述计算机可用存储介质可主要包括存储程序区和存储数据区, 其中, 存储程序区可存储操作系统、至少一个功能所需的应用程序等; 存储数据区可存储根据区块链节点的使用所创建的数据等。

[0160] 本发明所指区块链是分布式数据存储、点对点传输、共识机制、加密算法等计算机技术的新型应用模式。区块链 (Blockchain), 本质上是一个去中心化的数据库, 是一串使用密码学方法相关联产生的数据块, 每一个数据块中包含了一批网络交易的信息, 用于验证其信息的有效性 (防伪) 和生成下一个区块。区块链可以包括区块链底层平台、平台产品服务层以及应用服务层等。

[0161] 在本发明所提供的几个实施例中, 应该理解到, 所揭露的系统, 装置和方法, 可以通过其它的方式实现。例如, 以上所描述的装置实施例仅仅是示意性的, 例如, 所述模块的划分, 仅仅为一种逻辑功能划分, 实际实现时可以有另外的划分方式。

[0162] 所述作为分离部件说明的模块可以是或者也可以不是物理上分开的, 作为模块显示的部件可以是或者也可以不是物理模块, 即可以位于一个地方, 或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。

[0163] 另外, 在本发明各个实施例中的各功能模块可以集成在一个处理模块中, 也可以是各个模块单独物理存在, 也可以两个或两个以上模块集成在一个模块中。上述集成的模块既可以采用硬件的形式实现, 也可以采用硬件加软件功能模块的形式实现。

[0164] 上述以软件功能模块的形式实现的集成的模块, 可以存储在一个计算机可读取存储介质中。上述软件功能模块存储在一个存储介质中, 包括若干指令用以使得一台计算机设备 (可以是个人计算机, 服务器, 或者网络设备等) 或处理器 (processor) 执行本发明各个实施例所述方法的部分步骤。

[0165] 对于本领域技术人员而言, 显然本发明不限于上述示范性实施例的细节, 而且在不背离本发明的精神或基本特征的情况下, 能够以其他的具体形式实现本发明。因此, 无论从哪一点来看, 均应将实施例看作是示范性的, 而且是非限制性的, 本发明的范围由所附权利要求而不是上述说明限定, 因此旨在将落在权利要求的等同要件的含义和范围内的所有变化涵括在本发明内。不应将权利要求中的任何附关联图标记视为限制所涉及的权利要求。此外, 显然“包括”一词不排除其他模块或步骤, 单数不排除复数。系统权利要求中陈述的多个模块或装置也可以由一个模块或装置通过软件或者硬件来实现。第一, 第二等词语

用来表示名称,而并不表示任何特定的顺序。

[0166] 最后应说明的是,以上实施例仅用以说明本发明的技术方案而非限制,尽管参照较佳实施例对本发明进行了详细说明,本领域的普通技术人员应当理解,可以对本发明的技术方案进行修改或等同替换,而不脱离本发明技术方案的精神和范围。

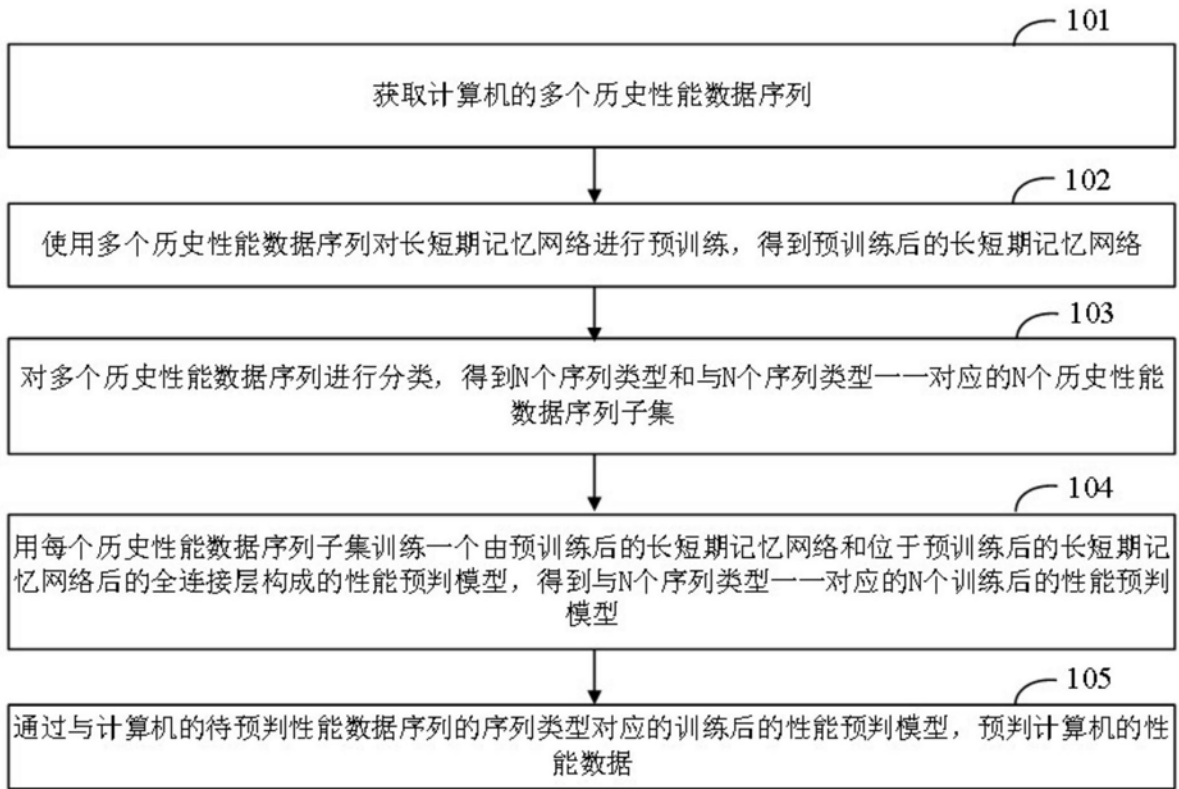


图1

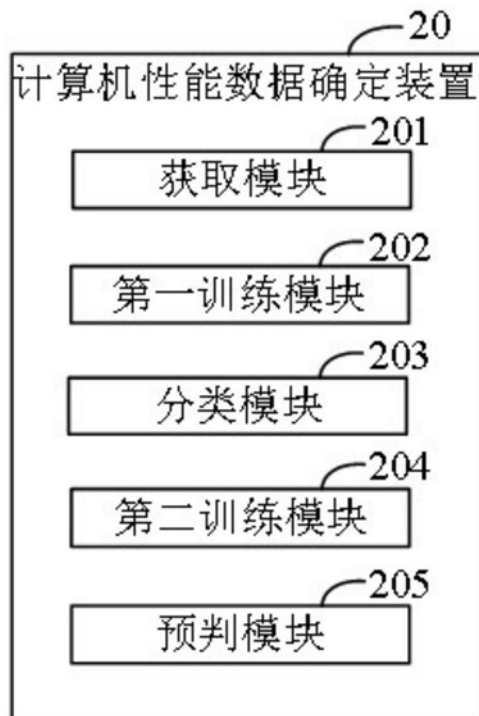


图2

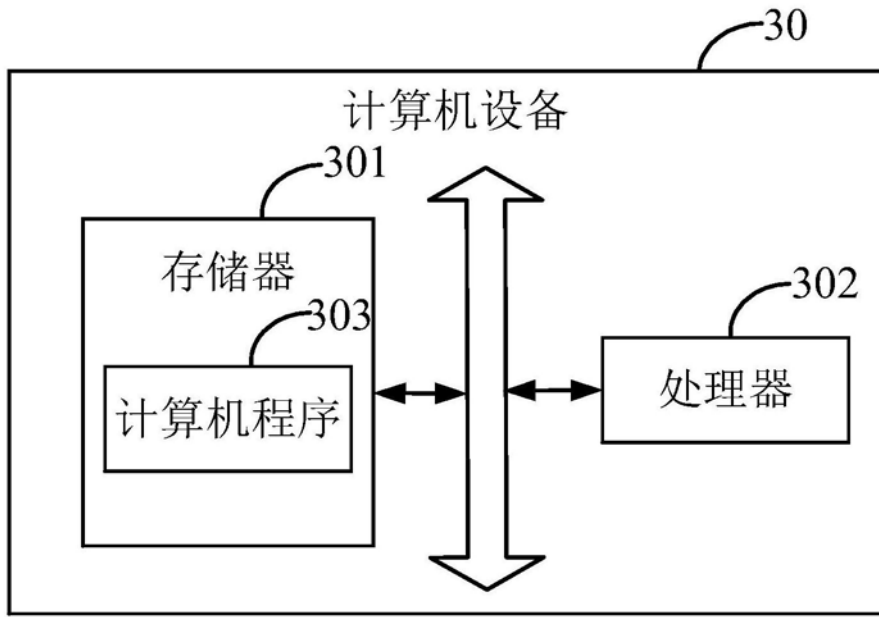


图3