



(12) 发明专利申请

(10) 申请公布号 CN 118214644 A

(43) 申请公布日 2024. 06. 18

(21) 申请号 202311721101.3

H04W 24/04 (2009.01)

(22) 申请日 2023.12.14

H04W 84/08 (2009.01)

(30) 优先权数据

18/083,102 2022.12.16 US

(71) 申请人 黑莓有限公司

地址 加拿大安大略省

(72) 发明人 S·J·亨克尔

(74) 专利代理机构 北京市金杜律师事务所

11256

专利代理师 丁君军

(51) Int.Cl.

H04L 41/0631 (2022.01)

H04L 41/069 (2022.01)

H04L 41/16 (2022.01)

H04L 9/40 (2022.01)

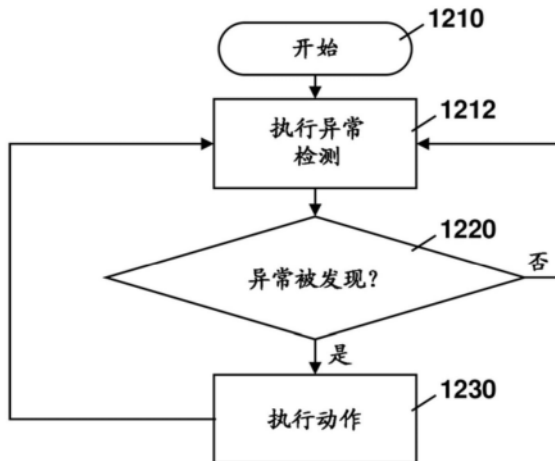
权利要求书2页 说明书13页 附图12页

(54) 发明名称

用于增量式质心集群的方法和系统

(57) 摘要

本公开的实施例涉及用于增量式质心集群的方法和系统。一种在计算设备处用于异常检测的方法,该方法包括在计算设备处存储针对多个数据点的全批次模型,其中多个数据点中的每个数据点与多个集群之一相关联,并且其中存储包括用于存储关于多个集群的信息而不存储多个数据点的表;从硬件传感器接收新数据点;确定新数据点落在全批次模型外,从而检测到异常;以及基于该异常执行动作。



1. 一种在计算设备处用于异常检测的方法,所述方法包括:
在所述计算设备处存储针对多个数据点的全批次模型,其中所述多个数据点中的每个数据点与多个集群之一相关联,并且其中所述存储包括用于存储关于所述多个集群的信息而不存储所述多个数据点的表;
从硬件传感器接收新数据点;
确定所述新数据点落在所述全批次模型外,从而检测到异常;以及
基于所述异常执行动作。
2. 根据权利要求1所述的方法,其中所述计算设备通过以下方式创建所述全批次模型:
选择数据点;
确定所述数据点落在到集群的中心的平均距离的阈值内;
向所述集群添加所述数据点;以及
针对数据点训练集中的后续数据点迭代地执行所述确定和所述添加。
3. 根据权利要求2所述的方法,其中向所述集群添加所述数据点包括:
递增指示所述集群内的数据点的数目的索引;
向针对所述集群的特征总和添加针对所述数据点的信息;
更新所述集群的质心;以及
更新包括所述集群中的点之间的近似距离的值。
4. 根据权利要求2所述的方法,还包括:
确定第二数据点没有落在到所述集群的所述中心的所述平均距离的所述阈值内;以及
创建针对所述数据点的新集群。
5. 根据权利要求4所述的方法,还包括:
确定所述异常检测系统中的集群的数目超过函数;
从所述异常检测系统移除所述集群中具有单个数据点的所有集群;以及
将所述阈值更新为更高的值。
6. 根据权利要求5所述的方法,其中所述函数是低于被处理的数据点的总数目的平方根的函数。
7. 根据权利要求1所述的方法,其中所述计算设备只存储关于每个集群的增量信息。
8. 根据权利要求1所述的方法,其中所述动作包括以下至少一项:
提供警报;以及
自主地控制与所述计算设备相关联的设备。
9. 根据权利要求1所述的方法,其中所述计算设备是物联网系统内的边缘设备。
10. 一种用于异常检测的计算设备,所述计算设备包括:
处理器;以及
通信子系统,其中所述计算设备被配置为:
存储针对多个数据点的全批次模型,其中所述多个数据点中的每个数据点与多个集群之一相关联,并且其中所述存储包括用于存储关于所述多个集群的信息而不存储所述多个数据点的表;
从硬件传感器接收新数据点;
确定所述新数据点落在所述全批次模型外,从而检测到异常;以及

基于所述异常执行动作。

11. 根据权利要求10所述的计算设备, 其中所述计算设备被配置为通过以下方式创建所述全批次模型:

选择数据点;

确定所述数据点落在到集群的中心的平均距离的阈值内;

向所述集群添加所述数据点; 以及

针对训练数据点集中的后续数据点迭代地执行所述确定和所述添加。

12. 根据权利要求11所述的计算设备, 其中所述计算设备被配置为通过以下方式向所述集群添加所述数据点:

递增指示所述集群内的数据点的数目的索引;

向针对所述集群的特征总和添加针对所述数据点的信息;

更新所述集群的质心; 以及

更新包括所述集群中的点之间的近似距离的值。

13. 根据权利要求11所述的计算设备, 其中所述计算设备还被配置为:

确定第二数据点没有落在到所述集群的所述中心的所述平均距离的所述阈值内; 以及
创建针对所述数据点的新集群。

14. 根据权利要求13所述的计算设备, 其中所述计算设备还被配置为:

确定用于异常检测的集群的数目超过函数;

移除所述集群中具有单个数据点的所有集群; 以及

将所述阈值更新为更高的值。

15. 根据权利要求14所述的计算设备, 其中所述函数是低于被处理的数据点的总数目的平方根的函数。

16. 根据权利要求10所述的计算设备, 其中所述计算设备只存储关于每个集群的增量信息。

17. 根据权利要求10所述的计算设备, 其中所述动作包括以下至少一项:

提供警报; 以及

自主地控制与所述计算设备相关联的设备。

18. 根据权利要求10所述的计算设备, 其中所述计算设备是物联网系统内的边缘设备。

19. 一种用于存储用于异常检测的指令代码的计算机存储介质, 所述指令代码在由计算设备上的处理器执行时, 使所述计算设备:

存储针对多个数据点的全批次模型, 其中所述多个数据点中的每个数据点与多个集群之一相关联, 并且其中所述存储包括用于存储关于所述多个集群的信息而不存储所述多个数据点的表;

从硬件传感器接收新数据点;

确定所述新数据点落在所述全批次模型外, 从而检测到异常; 以及

基于所述异常执行动作。

用于增量式质心集群的方法和系统

技术领域

[0001] 本公开涉及异常检测系统,并且具体地涉及用于异常检测系统的机器学习。

背景技术

[0002] 异常检测是由此能够在数据集中标识异常值的过程。它在复杂的软件系统(例如分布式系统)中发挥作用,并且允许对于系统的威胁被检测、错误的根本原因被标识、监控系统的健康、清理数据以及其他益处。

[0003] 传统上,异常检测系统是手动构建的。这涉及重要的领域知识和可能遇到的问题的预见。此外,这种系统往往是静态的,当对于系统的威胁可以演化时,这是不合需求的。

[0004] 在这个方面,机器学习算法最近被用于创建这样的异常检测系统。这样的机器学习算法通常创建能够处理大型数据集、可能随着时间的推移自适应的系统,并且可以比手动创建更好地工作。这样的机器学习算法的示例包括单类支持向量方法(Support Vector Method, SVM)、随机森林、局部异常值因子等等。

[0005] 然而,这样的机器学习算法在处理和存储器要求两方面都具有显著的开销。因此,这些算法用于在分布式系统内实现是不实际的,在这样的系统的边缘上的处理模块可能不具有高效运行这样的算法所需要的存储器或处理能力。

附图说明

[0006] 参考附图,本公开将更好地被理解,其中:

[0007] 图1是示出用于全批次(full batch)异常检测系统的处理和存储的框图,在该系统中处理和存储是在边缘设备处进行的。

[0008] 图2是示出用于全批次异常检测系统的处理和存储的框图,在该系统中处理和存储是在云设备处进行的。

[0009] 图3是示出用于全批次异常检测系统的处理和存储的框图,在该系统中处理和存储是在云设备处进行的,并且该全批次模型被推到边缘设备。

[0010] 图4是示出示例车辆系统的框图。

[0011] 图5是示出用于存储增量式质心集群信息的示例数据结构的框图。

[0012] 图6是示出用于针对增量式质心集群模型的训练的过程的流程图。

[0013] 图7是示出用于向集群添加数据点的过程的流程图。

[0014] 图8是示出针对用于异常检测的增量式质心集群模型的处理和存储的框图,在该模型中处理和存储是在边缘设备处进行的。

[0015] 图9是示出在三维空间中被处理的八个示例集群的图表。

[0016] 图10是示出良性数据点的集群和来自图9的绘图的异常标识的图表。

[0017] 图11是示例混淆矩阵的框图。

[0018] 图12是用于对于异常的检测执行动作的流程图。

[0019] 图13是能够与本公开的实施例一起使用的简化计算设备的框图。

具体实施方式

[0020] 本公开提供了在计算设备处用于异常检测的方法,该方法包括:在计算设备处存储针对多个数据点的全批次模型,其中多个数据点中的每个数据点与多个集群之一相关联,并且其中存储包括用于存储关于多个集群的信息而不存储多个数据点的表;从硬件传感器接收新数据点;确定新数据点落在全批次模型外,从而检测到异常;以及基于该异常执行动作。

[0021] 本公开还提供了用于异常检测的计算设备,该计算设备包括:处理器;以及通信子系统,其中该计算设备被配置为:存储针对多个数据点的全批次模型,其中多个数据点中的每个数据点与多个集群之一相关联,并且其中存储包括用于存储关于所述多个集群的信息而不存储多个数据点的表;从硬件传感器接收新数据点;确定新数据点落在全批次模型外,从而检测到异常;以及基于该异常执行动作。

[0022] 本公开还提供了用于存储用于指令代码的计算机存储介质,该指令代码在由计算设备上的处理器执行该计算机存储介质时,使该计算设备以:存储针对多个数据点的全批次模型,其中多个数据点中的每个数据点与多个集群之一相关联,并且其中存储包括用于存储关于多个集群的信息而不存储多个数据点的表;从硬件传感器接收新数据点;确定新数据点落在全批次模型外,从而检测到异常;以及基于该异常执行动作。

[0023] 本公开是指向物联网设备的。例如,这种设备可以包括汽车系统。然而,本文描述的方法和系统可以同样地被用于其他系统中。因此,本公开的技术并不局限于任何特定的系统。

[0024] 在一些情况下,软件程序可以在经由公共网络(例如互联网)连接到或可访问的系统中执行。恶意动作者可能使用各种漏洞技术来影响连接网络的软件程序的行为。在其他一些情况下,具有对系统的访问权限的恶意动作者可以将恶意代码插入到系统软件中,以影响软件的行为。

[0025] 具体地,随着物联网(Internet of Things, IoT)设备的激增持续增加,攻击者可能出于目的而针对的威胁面也增加。虽然基于保护的机制可能防范入侵的开始,但如果这种方法失败了,有方法来在端点上检测异常行为以促使保护端点、业务或用户免于伤害的动作可以是有益的。

[0026] 例如,现代可以具有超过1亿行的软件代码。随着软件在车辆中增长,攻击面也增长,这使这样的软件容易受攻击的影响。软件的每个构造不良的部分都表示可能由攻击者利用的潜在漏洞。

[0027] 用于阻止黑客进入系统的“城堡墙”可能是不充分的,或者随着时间的推移可能由恶意动作者规避。在一些情况下,系统可能无法检测被入侵的设备。

[0028] 在这方面,方法和系统可以被提供,其可以在系统内分析例如日志的数据以检测异常。这样的系统可以使用机器学习来发现数据异常。

[0029] 此外,在例如车辆的边缘系统上,可能存在存储器、通信和处理方面的限制。例如, IoT设备通常不会针对硬盘空间优化,这意味着存储数据的能力可能有限。此外, IoT设备经常具有有限的处理能力,并且不是所有的IoT设备都将容易地能够随时连接到云。

[0030] 可能导致关于日志分析的问题的进一步因素是IoT设备可能具有不同的行为。具体地,取决于用户如何使用IoT设备,或它在系统中如何被配置,在相似的IoT设备之间的正

常行为可能看起来不同。

[0031] 如本文所用,边缘设备和IoT设备可以互换使用。

[0032] 当前的异常检测系统可以使用“全批次模型”算法。这种模型涉及需要立刻对于所有数据进行训练的异常检测系统,并且示例可以包括单类SVM、孤立森林、局部异常值因子等。这种模型需要所有良性数据被存储用于处理。在这方面,将在边缘IoT设备处被需要大量的存储空间,这些空间可能不存在。

[0033] 具体地,参考图1,这示出了在边缘训练102、边缘再训练104和边缘推理106期间需要的步骤,其中全批次模型位于边缘设备上。

[0034] 图1的示例使用车载信息娱乐系统(In Vehicle Infotainment,IVI)110作为示例边缘设备。然而,该系统仅针对说明目的提供,并且本公开并不限于在车载信息娱乐系统中。

[0035] IVI 110创建系统日志112,并且从系统日志创建工程师特征114。

[0036] 在框116处,这样的特征被转换为特征向量。这所有边缘训练102、边缘再训练104和边缘推理106中的所有项发生。

[0037] 针对边缘训练102和边缘再训练104,边缘设备需要在框120处存储向量。这样做是为了最终在聚合上进行集群。针对边缘设备,如果需要的磁盘分配太小,那么将向量存储在边缘设备上的要求可能导致不可行的解决方案。如果模型需要重新训练,这种情况可能加剧。例如,如果向边缘设备提供新软件更新,新应用被安装,等等,则增加了存储问题。

[0038] 在框122处,从所存储的向量训练模型。模型可能需要在所有可用的良性数据上被训练。关于一些全批次模型,例如单类SVM,这可能花费很长时间,并且可能花费许多资源来训练。在一些情况下,模型可能无法在没有失败的情况下训练整个数据集。其他一些模型(例如孤立森林)可能不会遭受这种影响,但与单类SVM相比,它们通常遭受效力问题。

[0039] 从在框122处的训练可以创建全批次模型130。类似地,针对边缘再训练104,重训练模型140可以被用于创建更新的全批次模型130。

[0040] 然后全批次模型130可以在推理期间被使用,如框150所示,其然后可以生成输出152。

[0041] 全批次模型通常不能增量地训练模型,而是可能需要从头开始,这可能花费比边缘设备可以具有的资源更多的资源。

[0042] 因此,根据图1的实施例,主要问题之一是在边缘设备上的空间限制。在这方面,备选的方法是如果该存储和处理被移动到云解决方案。现在可以参考图2。

[0043] 具体地,边缘设备可以向云服务器或服务发送要被分析的持续的日志数据流,但这可能淹没来自边缘设备的通信信道,并且由此实现起来过于昂贵。

[0044] 在图2的情况下,示出了四个过程,具体地,发生在边缘设备上的边缘收集202、发生在云中的云训练204、发生在边缘设备上的边缘再收集206和发生在边缘设备上的边缘推理208(利用来自云的输入)。

[0045] 如关于图1的实施例,在边缘设备IVI 110上创建系统日志112,并且从系统日志创建工程师特征114。

[0046] 在框116处,这样的特征被转换为特征向量。这针对边缘收集202、边缘再收集206和边缘推理208中的所有项发生。

[0047] 针对边缘收集202和边缘再收集206,边缘设备在框210处向云提供向量以供云存储。

[0048] 在框212处,从所存储的向量训练模型。此外,从在框212处的训练可以在云处创建全批次模型220。

[0049] 全批次模型220然后可以在推理期间被使用,如框150所示,其然后可以生成输出152。

[0050] 虽然图2的实施例补救了在边缘设备处存储过多数据的问题,但这是以与云交互的成本为代价的。具体地,边缘设备能够每秒生成数千条记录。虽然可能应用一些过滤、聚合等因素,但这将导致每端点的大量带宽。这样的通信成本可能由预期传输数据的端点的数目而进一步增加。

[0051] 此外,针对在边缘再收集206期间的再训练,更多的数据需要被发送到云,具有与上述问题相同的问题。

[0052] 因此,针对训练和再训练二者,考虑到需要训练的数据的量,用于在云中训练的成本都可能是令人望而却步的。

[0053] 此外,一旦模型被训练,模型就需要将数据发送回端点用于推理,这是数据量庞大的并且还可能增加推理的时延时间,在一些情况下可能使其来不及对于推理采取行动。具体地,与云的通信增加了每次往返的时延,并且进一步要求组件有到云并回来的互联网连接。如果攻击者能够使端点离线,异常检测功能将停止工作。

[0054] 基于上文,用于边缘设备的另一种方法是一旦训练完成,将全批次模型移动到边缘。现在可以参考图3。

[0055] 在图3中的情况中,示出了四个过程,具体地是发生在边缘设备上的边缘收集302、发生在云中的云训练304,发生在边缘设备上的边缘再收集306和发生在边缘设备上的边缘推理308。

[0056] 如关于图1和图2的实施例,在边缘设备IVI 110上创建系统日志112,并且从系统日志创建工程师特征114。

[0057] 在框116处,这样的特征被转换为特征向量。这针对所有边缘收集302、边缘再收集306和边缘推理308中的所有项发生。

[0058] 针对边缘收集302和边缘再收集306,边缘设备在框210处向云提供向量以供云存储。

[0059] 框212处,从所存储的向量训练模型。此外,从框212处的训练,可以在云处创建全批次模型220。

[0060] 全批次模型220然后可以被部署到边缘,如用框310所示,其然后可以创建可以在推理期间使用的全批次模型320,如在框330处所示,其然后可以生成输出332。

[0061] 虽然图3的实施例降低了通过在推理期间关闭互联网连接来停止异常检测机制的能力,但这种方法仍然有一些值得关心的领域。具体地,仍然有从端点向云发送大量数据的问题,这在通信功能和成本方面可能令人望而却步。此外,再训练遭受与图2的实施例相同的问题。针对所有全批次模型,终端训练也面临着相同的限制。

[0062] 因此,本公开提供了用于在复杂系统上异常检测的增量式质心集群方法和系统。

[0063] 车辆软件系统

[0064] 虽然本公开的增量式质心集群方法和系统可以被用于任何计算系统,例如任何物联网系统,但在一个实施例中,它可以被用在车辆系统内。然而,本公开并不限于车辆系统,并且本文仅出于说明目的提供车辆系统。

[0065] 现在参考示出示例车辆系统的图4。

[0066] 在图4的实施例中,车辆400可以配备计算系统和通信系统。计算系统的一部分可以包括域410,域410可以具有创建日志数据的应用。此外,车辆400上的计算系统的一部分执行异常检测,如下所述。

[0067] 在图4的示例中,车辆400利用通信系统与蜂窝基站通信,如在图4中的eNB 420所示。基站可与核心网络430通信,该核心网络430然后可以通过网络432向云服务提供方440转发通信。例如,网络432可以是广域网,例如互联网。

[0068] 在其他一些实施例中,与特定蜂窝或无线通信协议相关联的任何技术都可以使用,而不是核心网络430。

[0069] 在一些实施例中,云服务440可以针对域提供处理,或者可以提供在域内使用的信息。

[0070] 此外,设备460可以被用于异常检测,而不是车辆。设备460可以是任何计算设备,并且可以包括物联网设备、移动设备、医疗装备、车辆或与车辆相关联的装备等选项。设备460可以通过网络432利用各种有线或无线技术通信,该技术包括但不限于以太网、光纤、蜂窝、Wi-Fi、卫星等选项。

[0071] 在图4的实施例中,在车辆400上的域410和设备450可以被视为边缘设备,并且在一些情况下具有有限的存储和/或处理能力。

[0072] 增量式质心集群

[0073] 如上所示,在边缘设备上现有机器学习模型的使用是不切实际的。特别地,当分析日志或表格数据时,利用现有机器学习异常检测算法,主要问题是没有足够的空间来存储用于这种算法所需的信息,没有足够的处理来创建全批次模型和/或没有足够的通信能力来向云提供所有数据。此外,数据是预先需要的,并且很难增量地更新。

[0074] 在这方面,在本公开中描述了用于增量式质心集群的算法。如在以下详细所述的,首个数据点将被用于创建集群,并且后续的点然后将被分析,以确定它们是在该集群内所有点互相之间的平均距离的阈值距离内。如果是,则新数据点被添加到集群中,并且基于新添加来调整集群形状。否则,将形成新集群。

[0075] 此外,在一些实施例中,集群的数目可能被限制为与被处理的点的数目成比例,使得推理时间不会变得太昂贵。在这种情况下,如果超过了期望的最大集群比例阈值,则小集群(例如通常是异常值的大小为1的那些集群)可能会被删除。当这种情况发生时,阈值可以由阈值调整因子提高以促进创建更少的新集群。

[0076] 一旦模型被训练,在已有集群的阈值距离内的任何预测都被认为是良性的,否则被认为是异常的。

[0077] 在增量式质心集群模型中,所存储的仅有数据是作为最大是被处理的点的数目的平方根的函数的集群,或者每个集群包含单个向量,该单个向量包括算法工作所需的所有信息。

[0078] 此外,增量式质心集群是增量的。与需要所有数据来执行集群相对,增量式质心集

群可能一次对一个数据点进行集群。

[0079] 如下所述,与其他机器学习模型相比,这种算法的效力被测试,并且被发现其接近或优于测试的最佳全批次算法。

[0080] 现在参考图5。图5的实施例示出了用于存储关于集群的信息的数据结构的示例。然而,图5的示例仅是用于存储这种信息的一种选项。存储在示例数据结构中的信息是可以针对集群被存储的信息类型的举例说明。

[0081] 在图5的示例中,具有多个集群的系统可以在表中被表示,其中每个行具有关于一个集群的信息。该信息可以包括关于针对特定维度或特征的集群质心的数据,信息正在针对该特定维度或特征被处理。在图5的示例中,正在处理“n”个维度或特征,并且因此列1到列n包括针对该特定维度或特征的关于集群质心的数据。这些列共同地在图5中示出为列510。

[0082] 接下来的n列存储以下值:该值提供跨属于该集群的所有点的所有特征的总和。特别地,如在下文更多细节中所述,列n+1到列2n,表示为列520,提供了累积地针对集群的、针对正被分析的维度或特征中的每个维度或特征的特征总和。

[0083] 列530提供了集群中的点的数目。换句话说,已经被分析并被存储在该集群中的数据点的数目由列530表示。

[0084] 列540提供了集群中的点之间的近似总距离。换句话说,数据点之间的距离度量可能随着新数据点被添加到集群而被更新。

[0085] 根据图6的实施例,图5的数据结构或类似的数据结构可以被用于训练增量式质心集群算法。

[0086] 具体地,图6的过程在框610处开始并且进行到框612,在其中数据点被选择。如本领域技术人员将理解的,数据点选择可以涉及如以上图1至图3中所描述的使用日志文件中的行,在数据被创建时处理数据,使用特征向量等选项。在一些情况下,数据点表示来自传感器的数据,例如车辆传感器或来自车辆的一组传感器。在一些情况下,数据点可以直接从传感器、电子控制单元(Electronic Control Unit, ECU)或其他要被处理的计算单元接收。在一些情况下,来自传感器或多个ECU的数据可以被存储在日志文件中,并且可以提前执行一些处理。其他选项是可能的。

[0087] 过程从框612进行到框620,在其中进行检查以确定来自框612的所选数据点是否是用于增量式质心集群算法的首个数据点。如果是,则过程继续到框630,在其中数据点被指派给新集群。特别地,用于图5中的列510的值与针对正被传入算法中的数据点而被分析的n个特征或维度。

[0088] 此外,列520被填充有与正被处理的数据点的n个特征对应的值。这是因为在该维度中的特征的总和只具有该一个数据点,并且因此总和是原始值。

[0089] 列530被赋值为1,指示单个数据点。

[0090] 列540被赋值为0,指示当前该集群中的数据点之间没有距离。具体地,由于只有一个数据点,所以数据点之间的距离为0。

[0091] 再次参考图6,过程从框630进行到框632,在其中新集群被添加到针对系统的表中。

[0092] 相对地,从如果数据点不是针对系统的首个数据点,则过程从框620进行到框640,

在其中进行检查,以确定数据点是否在到系统中任何已有集群的集群中心的平均距离的阈值内。

[0093] 到集群中心的平均距离以0开始。为了开始构建集群,本文称为“集群距离”的全局变量可以被使用,其监控来自新点到最近集群的距离。在每次迭代经过首个数据点时,在新数据点与最近的集群之间的距离被添加到该变量中,以0开始。

[0094] 第二变量,本文称为“计数”,可以指示被保存到系统的数据点的总数目。

[0095] 在640框处的检查将“阈值距离”乘以“集群距离”除以“计数”的值。以这种方法,“集群距离”除以“计数”意味着表示每个点到最近集群的平均距离。这是有用的,因为用于“阈值距离”超参数的良好选择可能很难提前确定,并且随着时间的推移,如果集群的密度变得或多或少稠密,条件可能会更灵活。

[0096] 这样的全局变量在一些情况下可以被用于所有的训练。在其他一些情况下,全局变量可以被使用,直到集群具有足够的点为止。在此之后,在640框处检查然后将“阈值距离”乘以列540的值,除以在列530处的计数。具体地,列540的值除以列530处的值意为表示每个点到其集群中心的平均距离。

[0097] 换句话说,在框640处的检查发现数据点是否在数据点到它们的最近集群的平均距离的平均距离的阈值距离内。

[0098] 在一些情况下,阈值距离可能是针对系统赋值的值,这在一些情况下可以被调整,如下所述。此外,在一些实施例,被使用的距离是欧几里得距离。

[0099] 如果在框640处确定数据点不在到集群中心的平均距离的阈值内,则过程进行到框630,在其中数据点被指派给新集群,并且然后进行到框632,在其中新集群被添加到系统。

[0100] 相对地,如果在框640处确定数据点在到集群中心的平均距离的阈值内,则处理继续到框642,在其中数据点被映射到最近的集群。特别地,映射可以根据图7的实施例完成。

[0101] 参考图7,过程在框710处开始,并且进行到框720,在其中针对被映射到集群的新数据点的值被计算并被添加到特征总和列520。该新总和被存储在列520中。

[0102] 过程接下来进行到框730,在其中列530中的值被递增1,指示集群现在具有附加的数据点。

[0103] 过程接下来进行到框740,在其中在框720处被存储的特征值的总和除以在框730处被存储的数据点的总数目。这些值作为质心值被存储在列510中。

[0104] 过程然后进行到框750,在其中新数据点的距离值被添加到先前的集群中心。这些值然后可能除以在框730处被存储的数据点的总数目,并且然后求平均值,以创建针对所有点到集群中心之间的平均距离的近似值,该近似值然后可以被存储在列540中。

[0105] 再次参考图6,因此,表示集群的列可以利用框642处的新数据点来更新。

[0106] 从框632起,如果正在添加新集群,则在一个备选实施例中,可以进行检查以在与数据点的总数目相比较时,查看最大集群计数是否超过预定的阈值。特别地,用于机器学习增量式质心集群算法的超参数可以被设置,这指示最大的集群比例。在一个实施例中,如果集群的总数目超过低于被处理的总点的平方根的函数,则在框650处的检查为真。

[0107] 从框650起,如果检查为真,则过程进行到框652,在其中集群的数目可以被减少。在一些情况下,集群的数目可以通过移除大小为1的所有集群来减少,例如,因为这些集群

指示异常值。

[0108] 此外,为了缓解在大小为一的集群的数目中的未来增长,处理阈值距离的超参数可以提高。具体地,用于在框640处的检查的阈值距离可以提高一个量,以允许数据点位于更高距离处的集群内。

[0109] 在一些情况下,阈值中的提高可以通过某个预设因子,例如被指定为阈值调整超参数的因子。在一些情况下,提高可能基于提高的枚举清单。在一些情况下,该提高可以基于数据集大小或所处理的数据点的数目的某个函数。用于提高大小的其他选项是可能的。

[0110] 从框642、框652或框650起,如果最大的集群比例不被超过,则过程进行到框660,在其中进行检查,以确定是否还有更多的数据点要处理。如果是,则过程进行到框662,在其中下一个数据点被选择,并且然后继续到框640。

[0111] 相对地,如果没有要分析的进一步数据点,则过程进行到框670并且结束。

[0112] 因此,图6的过程示出了用于增量式质心集群算法的训练步骤。一旦训练完成,在推理期间,数据点可以被分析,并且如果它在模型中任何已有的集群的距离阈值内,则可以发现它属于集群。在这种情况下,数据点将被认为是良性的。相反,如果数据点不属于集群,则其被认为是异常的。

[0113] 利用图6和7的过程,针对磁盘使用,需要被保存到磁盘的组件仅有增量式模型,并且没有用于增量式质心集群的其他数据需要被存储。

[0114] 此外,增量式质心集群模型可以一次处理一条记录,保持其峰值资源需求尽可能低。

[0115] 此外,图6和7的实施例不需要云连接性,因为所有训练和推理都发生在边缘设备上。

[0116] 此外,针对每个边缘设备可以训练单独的模型。

[0117] 例如,参考图8,示出了三个过程,分别被称为边缘训练802、边缘再训练804和边缘推理806,所有这些都发生在边缘设备处。

[0118] 图8的示例使用车载信息娱乐系统 (IVI) 110作为示例边缘设备。然而,这种设备仅针对说明目的提供,并且本公开不限于车载信息娱乐系统。

[0119] IVI 110创建系统日志112,并且从系统日志中创建工程师特征114。

[0120] 这些特征在框116处被转换为特征向量。这适用于所有边缘训练802、边缘再训练804和边缘推理806。

[0121] 每条记录然后可以增量地被处理,因此不需要在图1所示的存储。而是,特征向量116可以被用于训练该模型(如在框810处所示),或更新该模型(如在框830处所示)。训练或更新的模型可以被存储(如图5的表)以用于全批次模型820。

[0122] 然后全批次模型820可以在推理期间使用,如在框840处所示,其然后可以生成输出842。

[0123] 因此,根据图8的实施例,缓解了全批次模型的局限性。

[0124] 此外,在端点可能具有非常相似的良性行为的情况下,增量式模型可以与设备联合,并且模型更新可以被应用于定制该端点的任何行为。另外的点由于模型不需要从头开始训练,它们可以利用任何观察到的新行为来更新已有的模型。

[0125] 测试

[0126] 针对测试,图6的过程被用于具有8个集群的数据集上,这些集群具有三维特征,在图9的实施例中示出。特别地,图9的实施例以灰度示出了8个集群,其中图表910、图表920和

图表930是沿着三个轴之一旋转示出的相同的8个集群。

[0127] 在测试用例中,参考940部分示出的集群(在集群中的点用“x”示出)被指定为异常行为,并且其余七个集群被指定为良性行为。因此,图6的过程被运行为在良性行为上的训练模型。

[0128] 然后推理过程在剩余的数据点上运行。在这种情况下,图10的图表被创建,其中图表1010、1020和1030示出了沿着不同维度的轴之一旋转的相同集群。特别地,如在图10中所示,许多重叠的集群覆盖了良性活动,但视觉上的异常行为(用“x”符号示出,并且通常用附图标记1040示出)几乎没有受到影响。

[0129] 该过程可以基于混淆矩阵被评估,例如关于图11所示。特别地,如在图11中所示,矩阵比较了预测结果与实际结果。如果针对数据点的预测和实际值都是数据点是良性的,则这个数据点被分类为真阳性1110。

[0130] 如果预测数据点是异常的,但针对数据点的实际值是数据点是良性的,则这个数据点被分类为假阴性1120。

[0131] 如果预测数据点是良性的,但针对数据点的实际值是数据点是异常的,则这个数据点被分类为假阳性1130。

[0132] 如果针对数据点的预测和实际值都是数据点是异常的,则这个数据点被分类为真阴性1140。

[0133] 基于图11的混淆矩阵,系统目标是使真阳性1110和真阴性1140值最大化,同时使假阴性1120和假阳性1130值最小化。

[0134] 以图5和图6的示例为例,混淆矩阵被发现为:

[0135]		数目
	真阳性	375
	假阴性	2
	假阳性	3
	真阴性	30

[0136] 表1:针对测试用例的混淆矩阵

[0137] 因此,从表1中,针对良性数据点的精确度是99.2%,并且针对异常数据点的精确度是93.3%。

[0138] 此外,针对测试的召回率,特别是在总阳性中什么百分比被预测为阳性,针对良性数据点的召回率是99.5%,并且针对异常数据点的召回率是90.9%。

[0139] 针对测试的F1得分考虑了假阳性和假阴性两者,并且其是精确率和召回率的调和平均值。它可以被表示为以下式1:

$$[0140] \quad F1 \text{ 得分} = \frac{2}{\frac{1}{\text{精确度}} + \frac{1}{\text{召回率}}} \quad (1)$$

[0141] 针对良性数据点的F1得分是99.3,并且针对异常数据点的F1得分是92.3。

[0142] 因此,测试的总体准确率是98.8%,并且调和F1得分是95.7,这指示分类效力较高。

[0143] 因此,该模型可能以高效力被用于具有有限存储和/或处理的计算设备上。针对多个集群,仅需要的存储是关于该表的。特别地,要存储的数据仅有集群,其是最大是被处理的点的数目的平方根的函数,或者每个集群包含单个向量,该单个向量包括过程工作所需

的所有信息。

[0144] 当集群比例超过阈值时,处理可能通过配对减少集群的数目并且然后通过提高所使用的距离阈值的大小来辅助。这进一步有助于调整算法的参数。

[0145] 该过程是增量的,并且不需要所有数据来执行集群。具体地,该方法可能一次对一个数据点进行集群。

[0146] 比较

[0147] 在测试中,本公开的增量式质心集群的效力被发现接近或优于所测试的全批次算法中的最佳算法。

[0148] 具体地,在对上述内容的进一步测试中,增量式质心集群模型针对单类SVM模型被测试。使用相同的训练数据、验证数据和测试数据。使用训练数据来训练每个模型。验证数据被用于找到产生异常检测的合理分离的超参数的最佳选择,并且测试数据被用于比较结果。

[0149] 针对单类SVM模型,调整了以下超参数:

[0150]	超参数	值
	内核(指定要使用的内核类型)	[“linear” , “poly” , “rbf” , “Sigmoid”]
	次数(如果被选择,多项式内核的次数)	[3, 4, 5]
	Nu(可以被认为是污染因子)	[0.001, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5]
	Gamma(用于“rbf”、“poly”和“Sigmoid”的内核系数)	[“scale” , “auto”]
	Tol(用于停止准则的公差)	[1e-2, 1e-3, 1e-4]

[0151] 表2:单类SVM的超参数针对增量式质心集群,调整了以下超参数:

[0152]	超参数	值
	阈值(从质心的距离与在该集群中所有点之间的平均距离成比例)	[0.3, 0.5, 0.7, 1]
	最大集群比例(确保集群仍然是低于被处理的总点的平方根的函数)	[0.5, 0.6, 0.7, 0.75, 0.8, 0.85]
	阈值调整(当集群的数目超过最大集群比例时,阈值应该提高多少)	[1.05, 1.1, 1.15, 1.2, 1.3, 1.4, 1.5]
	距离阈值(一旦定义了集群,期望属于集群的点集群可以离质心多远)	[1.5, 1.8, 2.0, 2.2, 2.5]

[0153] 表3:用于增量式质心集群的超参数

[0154] 多次运行测试改变超参数,用于单类SVM的针对数据集的最佳性能被发现:RBF内核,Gamma设置为“scale”,公差是无关紧要的,并且Nu设置为0.001。针对这个超参数集,可以发现以下混淆矩阵:

[0155]		数目
--------	--	----

[0156]	真阳性	344
	假阴性	3
	假阳性	12
	真阴性	18

[0157] 表4:针对单类SVM的混淆矩阵

[0158] 从以上可知,针对良性数据的准确率是96.6,并且针对异常数据的准确率是85.7。针对良性数据的召回率是99.1,并且针对异常数据的召回率是60.0。针对良性数据的F1得分是97.9,并且针对异常数据的F1得分是70.6。

[0159] 总体上,单类SVM的准确率是96.0,调和F1是82.0,并且曲线下面积 (Area Under Curve, AUC) 是79.6。

[0160] 类似地,多次运行测试改变超参数,当阈值过低 (0.3) 并且集群比例过高 (大于0.8) 时,模型表现更差。针对这个模型,可以发现以下混淆矩阵:

[0161]		数目
	真阳性	345
	假阴性	2
	假阳性	5
	真阴性	25

[0162] 表5:用于增量式质心集群的混淆矩阵

[0163] 从以上可知,针对良性数据的准确率是98.6,并且针对异常数据的准确率是92.6。针对良性数据的召回率是99.4,并且针对异常数据的召回率是83.3。针对良性数据的F1得分是99.0,并且针对异常数据的F1得分是87.7。

[0164] 总体上,增量式质心集群的准确率是98.1,调和F1是93.0,并且AUC是91.46。

[0165] 基于这些结果,在调整增量式质心集群模型之后,拟合比单类SVM更准确。进一步,发现增量式质心集群比单类SVM模型运行更快。

[0166] 异常检测的使用

[0167] 在实践中,异常检测的使用在边缘设备、服务器、云和/或其他计算设备处进行以执行各种操作。具体地,现在参考图12。

[0168] 图12的实施例在框1210处开始并且进行到框1212,在其中异常检测被执行。在这种情况下,所执行的异常检测可以使用在以上描述的图6和7的实施例中创建的全批次模型。新数据点然后可以例如从日志文件被接收,该新数据点可以基于来自硬件传感器的输入被创建或在一些情况下基于软件过程。

[0169] 通过确定该点是否是来自在全批次模型中的任何已有集群的距离阈值,新数据点可以被检查以确定其是否落在全批次模型中。如果是,则该数据点被认为是良性数据点。如果不是,则数据点被认为是异常数据点。这可以在数据点基于日志文件内容等选项被创建时被重复。

[0170] 一旦异常检测已经被执行 (或当其正在执行连续分析的系统中被执行时),处理进行到框1220,在其中进行检查以确定在正在被处理的数据中是否发现任何异常。如果不是,

则过程可以进行回到框1212以继续执行异常检测。

[0171] 相对地,如果在框1220处发现异常,则过程可以进行到框1230,在其中可以执行动作。正在被执行的动作可以特定于边缘设备中被部署在其中的应用的类型。例如,在车辆系统中,如果检测到异常,一个可能的动作是向车辆的驾驶员提供警报。例如,警报可以包括信息娱乐系统上、在汽车的仪表盘上的消息,向驾驶员发出的音频警报,方向盘上的触觉反馈等选项。警报可以指示驾驶员应该使车辆被维修、应该停车并且停止驾驶车辆、应该联系制造商或经销商、应该重置系统等选项。

[0172] 在其他一些情况下,该动作可以是对第三方而不是驾驶员的警报。这种警报可以被发送给汽车制造商、经销商、车队运营商、业务IT部门等选项。该警报可以包括导致异常的数据、围绕异常的上下文、其他车辆传感器读数等这样的信息。这然后可以允许第三方做出关于车辆的决定,例如是否使车辆被维修、从道路上移除车辆、重置车辆上的软件、是否软件更新应该被推送到车辆等这样的动作。

[0173] 在一些情况下,动作可以涉及车辆本身自主地采取动作。例如,如果车辆具有自主功能,车辆可以停车;在一些情况下,车辆可以忽略异常传感器读数;在一些情况下,车辆可能被限制速度;等等。

[0174] 针对非车辆情况,动作可以类似地包括基于异常数据检测来警告用户、第三方或在边缘设备上执行一些功能。

[0175] 在一些情况下,该动作可以在再训练该模型中使用新数据点。

[0176] 本公开并不意味着被限于可以基于异常的检测被执行的任何特定动作,并且提供以上示例仅用于说明。

[0177] 计算设备

[0178] 以上域、网络元件、IoT或边缘设备、云服务、节点和其他计算平台可以使用任何计算设备实现。关于图13示出了计算设备的简化图解。图13的计算设备可以是任何固定或移动的计算设备。

[0179] 在图13中,设备1310包括处理器1320和通信子系统1330,其中处理器1320和通信子系统1330协同以执行上述实施例的方法。通信子系统1330允许设备1310与其他设备或网络元件通信,并且基于正在执行的通信类型可能不同。此外,通信子系统1330可以包括多个通信技术,包括任何有线或无线通信技术。

[0180] 处理器1320被配置为执行可编程逻辑,这些逻辑可以连同数据一起被存储在设备1310上,并且在图13的示例中示出为存储器1332。存储器1332可能是任何有形的、非暂态的计算机可读存储介质,该介质存储指令代码,该指令代码在由处理器1320执行时使设备1310执行本公开的方法。计算机可读存储介质可以是有形的或暂态的/非暂态的介质,例如光学介质(例如,CD、DVD等)、磁性介质(例如,磁带)、闪存驱动器、硬盘驱动器或其他在本领域中已知的存储器。

[0181] 备选地,或除了存储器1332之外,设备1310可以从外部存储介质访问数据或可编程逻辑,例如通过通信子系统1330。

[0182] 在图13的示例中,一个或多个传感器1340可以与计算设备相关联。然而,这是可选的,并且在一些情况下,计算设备1310将不会与传感器相关联。

[0183] 在一个实施例中,在设备1310的各种元素之间的通信可以通过内部总线1360进

行。然而,其他形式的通信是可能的。

[0184] 本文描述的实施例是具有与本申请的技术元素对应的元素的结构、系统或方法的示例。本说明书可以使本领域技术人员能够制造和使用具有同样与本申请的技术元素对应的备选元素的实施例。因此,本申请的技术的预期范围包括与本文描述的本申请的技术没有区别的其他结构、系统或方法,并且还包括与本文描述的本申请的技术有非实质性差异的其他结构、系统或方法。

[0185] 虽然操作在图中是以特定顺序描述的,但这不应该被理解为要求这些操作以示出的特定次序或顺序执行,或者所有说明的操作都被执行,以实现期望的结果。在某些情况下,可以采用多任务和并行处理。此外,在以上描述的实现中各种系统组件的分离不应该被理解为在所有实现中都需要这种分离,并且其应该理解被描述的程序组件和系统通常可以被集成在单个软件产品中或打包成多个软件产品。

[0186] 此外,在各种实现中作为分立或单独描述和示出的技术、系统、子系统和方法可以与其他系统、模块、技术或方法组合或集成。被示出或讨论为彼此耦合或直接耦合或通信的其他项目可以通过某个接口、设备或中间组件间接耦合或通信,无论是电的、机械的还是其他方式。本领域技术人员可以确定并且可以做出改变、替换和变更的其他示例。

[0187] 虽然以上详细描述已经示出、描述并指出了应用于各种实现的本公开的基本新颖特征,但是应当理解,所示系统的形式和细节的各种省略、替换和改变可以由本领域技术人员进行。另外,方法步骤的顺序并不由它们在权利要求中出现的顺序来暗示。

[0188] 当消息被发送到电子设备或从电子设备发送消息时,这样的操作可能不是立即的或直接来自服务器的。它们可以从支持本文描述的设备/方法/系统的服务器或其他计算系统基础设施同步或异步地传送。前述步骤可以全部或部分地包括去往/来自设备/基础设施的同步/异步通信。此外,来自电子设备的通信可以是到网络上的一个或多个端点。这些端点可以由服务器、分布式计算系统、流处理器等来服务。内容分发网络(CDN)还可以提供与电子设备的通信。例如,除了典型的服务器响应之外,服务器还可以为内容分发网络(CDN)提供或指示数据以等待电子设备在稍后的时间(诸如电子设备的后续活动)下载。因此,数据可以直接从作为系统的一部分或与系统分离服务器或其他基础设施(例如分布式基础设施或CDN)发送。

[0189] 通常,存储介质可以包括以下项中任何或某种组合:半导体存储器设备,例如动态或静态随机存取存储器(DRAM或SRAM)、可擦除可编程只读存储器(EPROM)、电可擦除可编程只读存储器(EEPROM)和闪存;磁盘,例如固定盘、软盘和可移除磁盘;另一磁性介质,包括磁带;光学介质,例如光盘(CD)或数字视频光盘(DVD);或其他类型的存储设备。注意,上面讨论的指令可以被提供在一个计算机可读或机器可读存储介质上,或者备选地,可以被提供在分布在可能具有多个节点的大型系统中的多个计算机可读或机器可读存储介质上。这样的计算机可读或机器可读存储介质被认为是物品(或制品)的一部分。物品或制品可以指任何制造的单个组件或多个组件。一个或多个存储介质可以位于运行机器可读指令的机器中,或者位于可以通过网络从其下载机器可读指令以供执行的远程站点。

[0190] 在前面的描述中,阐述了许多细节以提供对本文所公开的主题的理解。然而,在沒有这些细节中的一些细节的情况下也可以实践实现。其他实现可以包括对上面讨论的细节的修改和变化。所附权利要求旨在涵盖此类修改和变化。

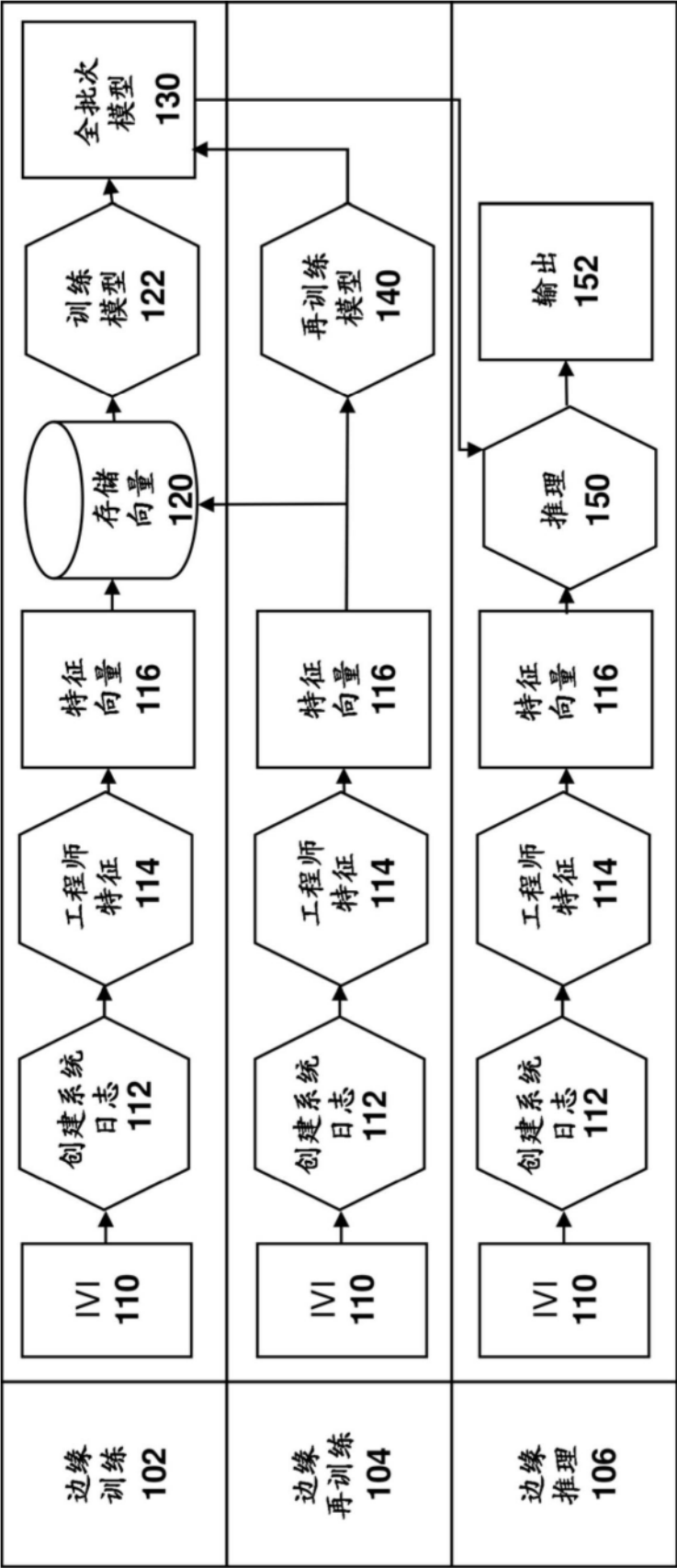


图1

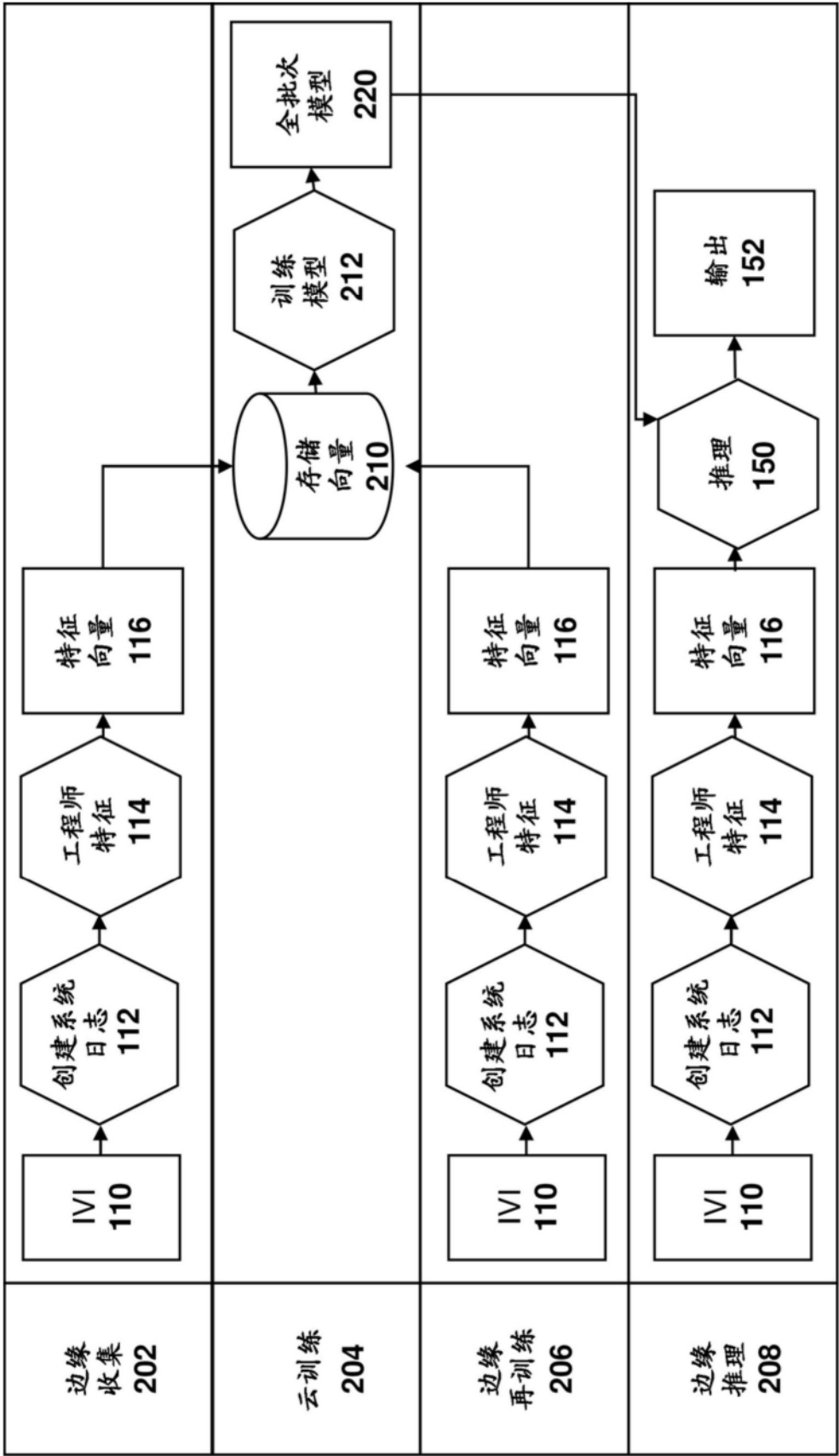


图2

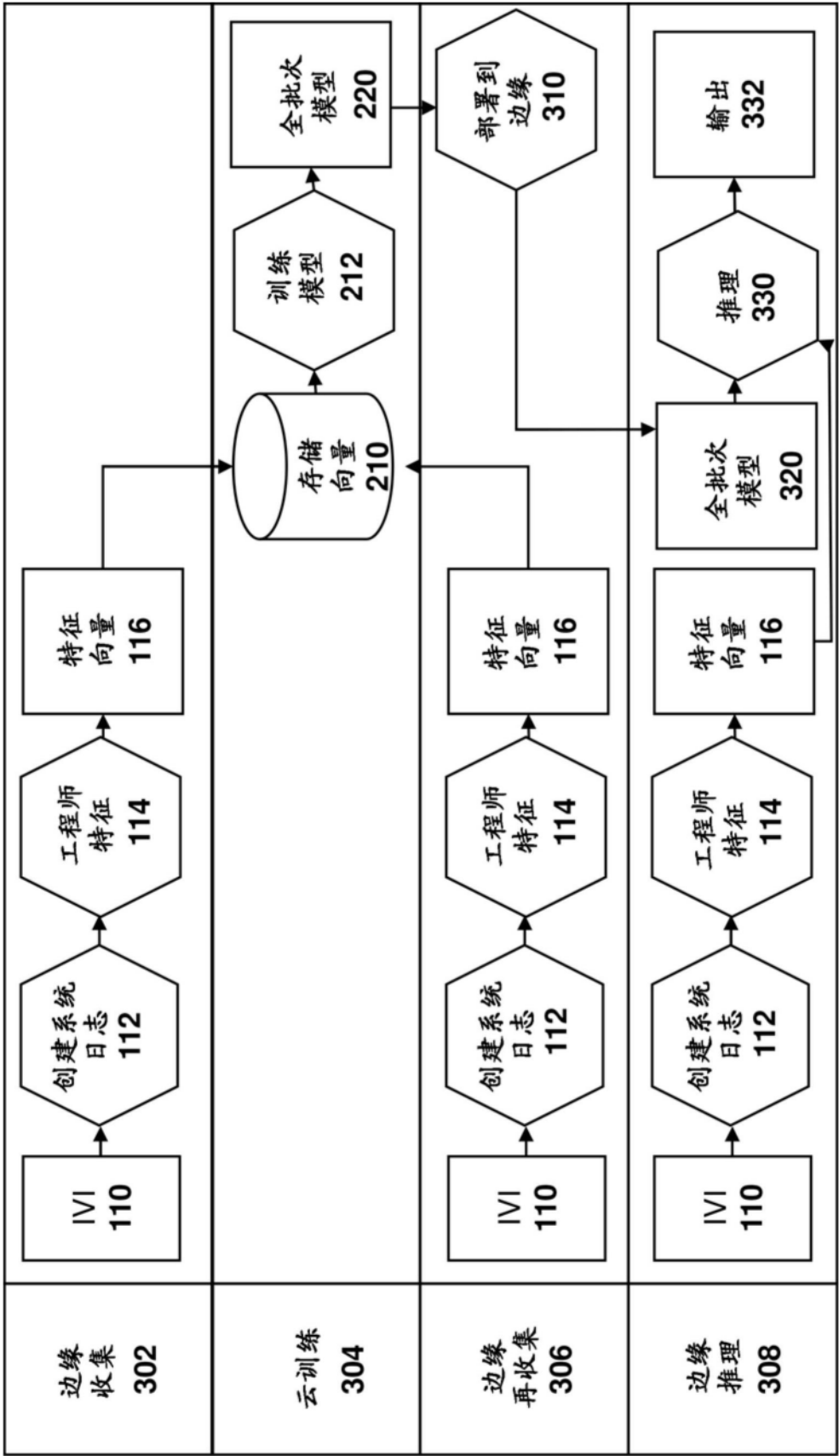


图3

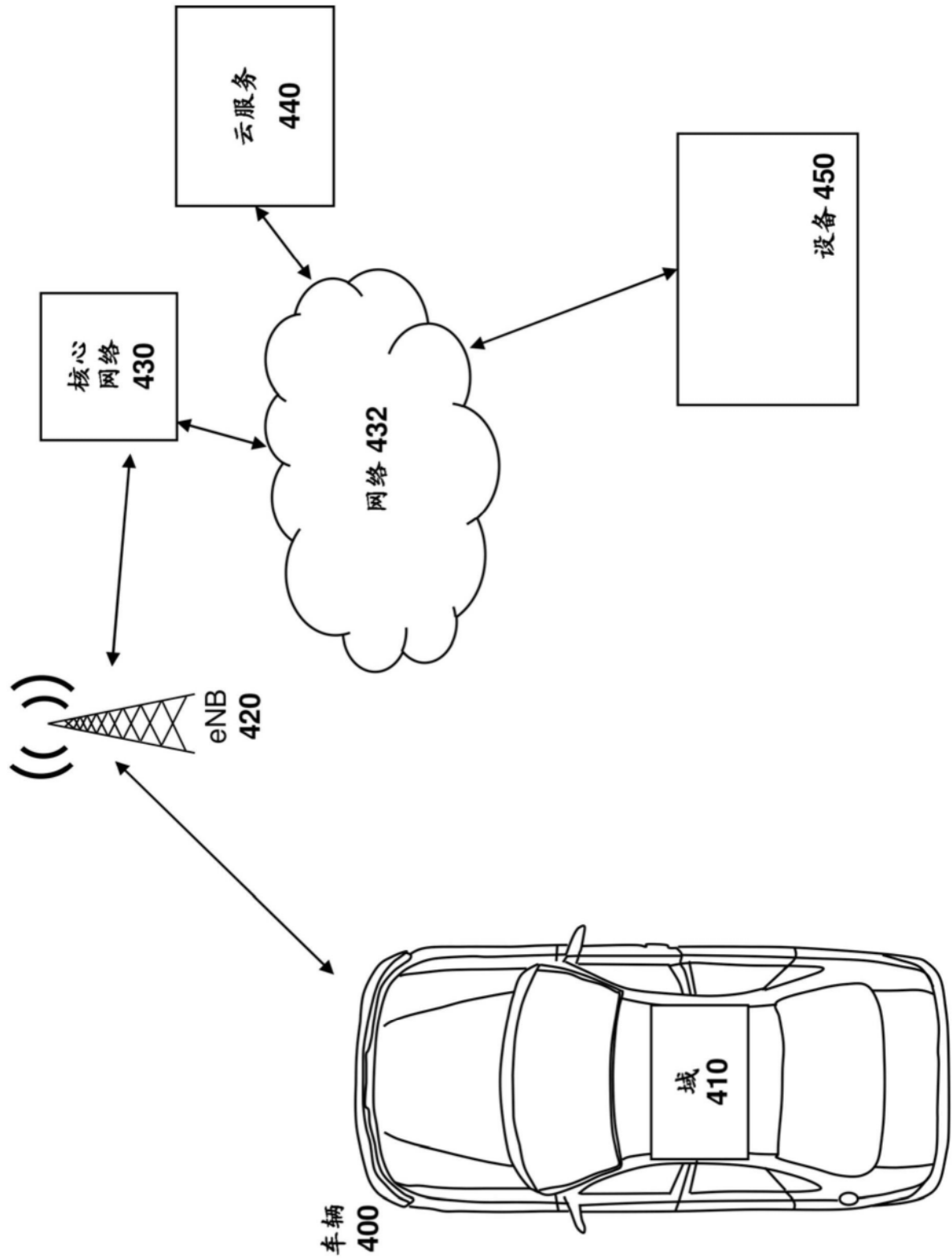


图4

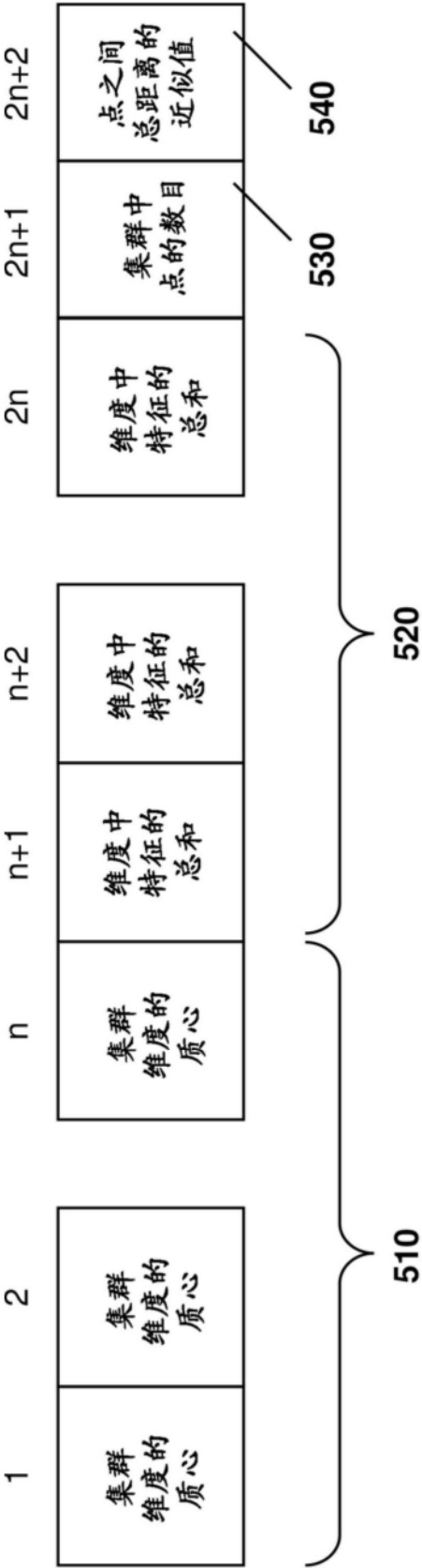


图5

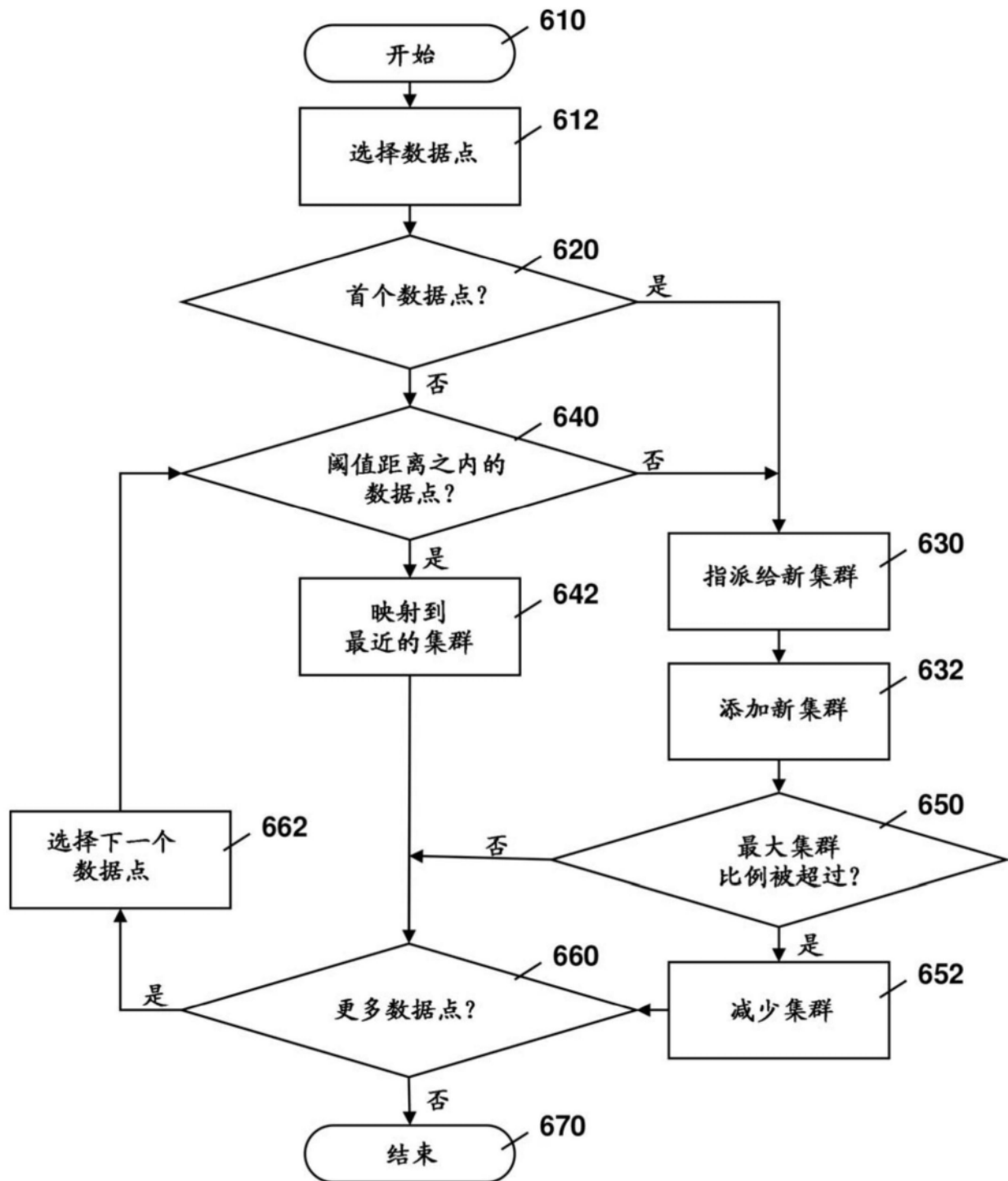


图6

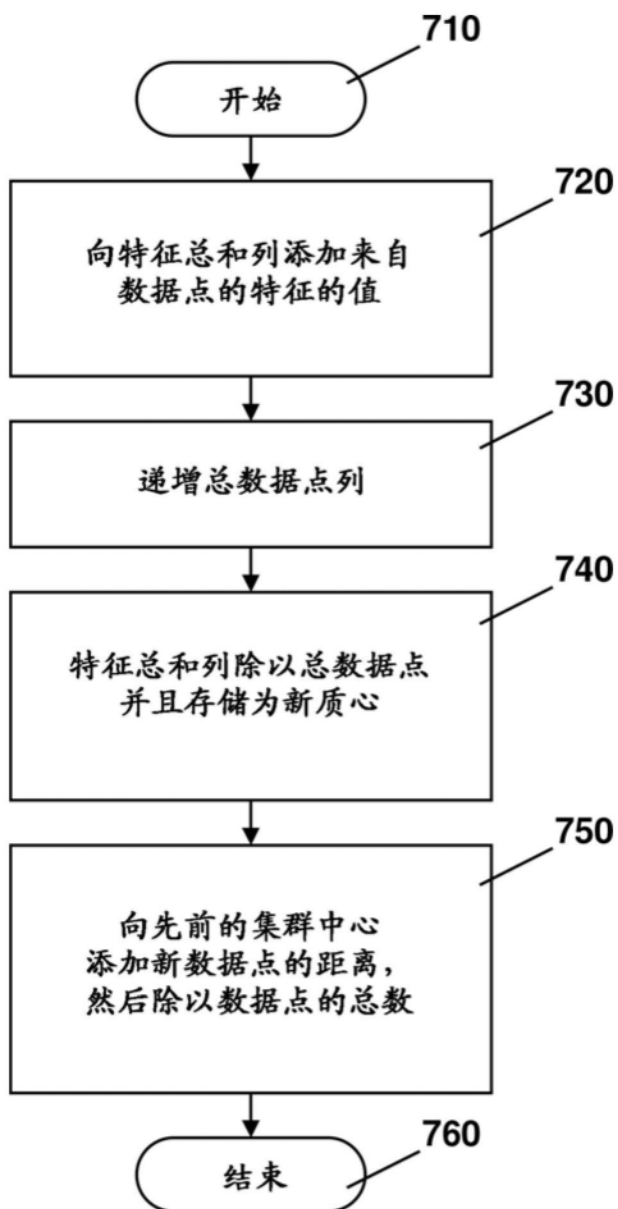


图7

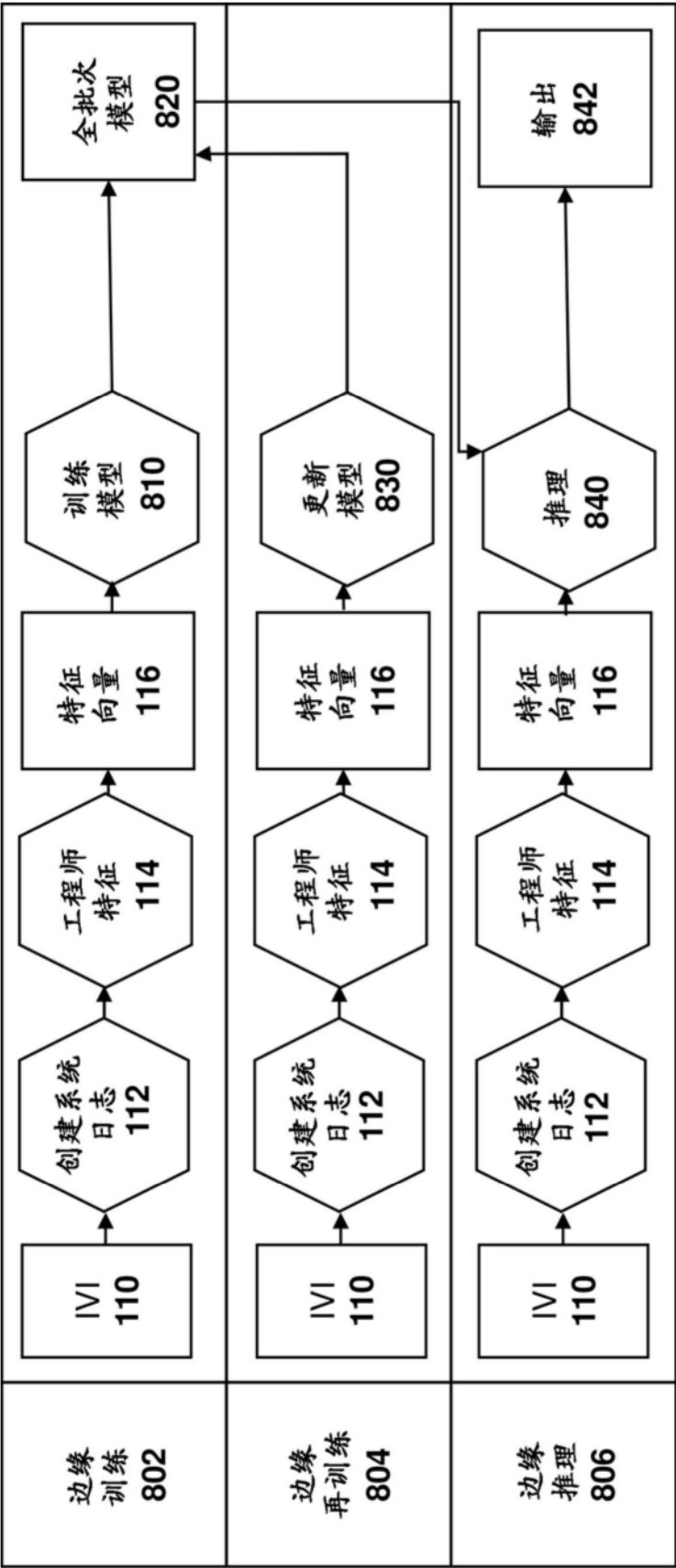


图8

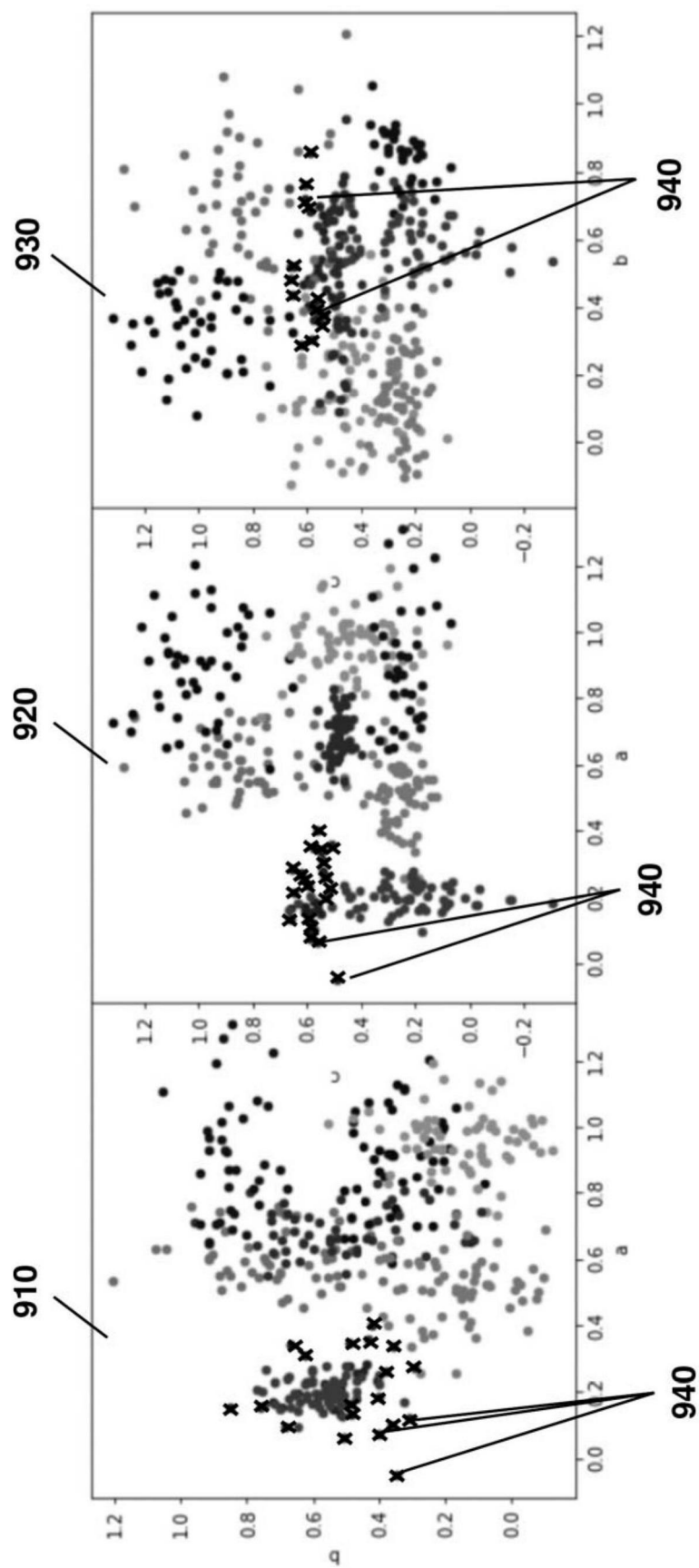


图9

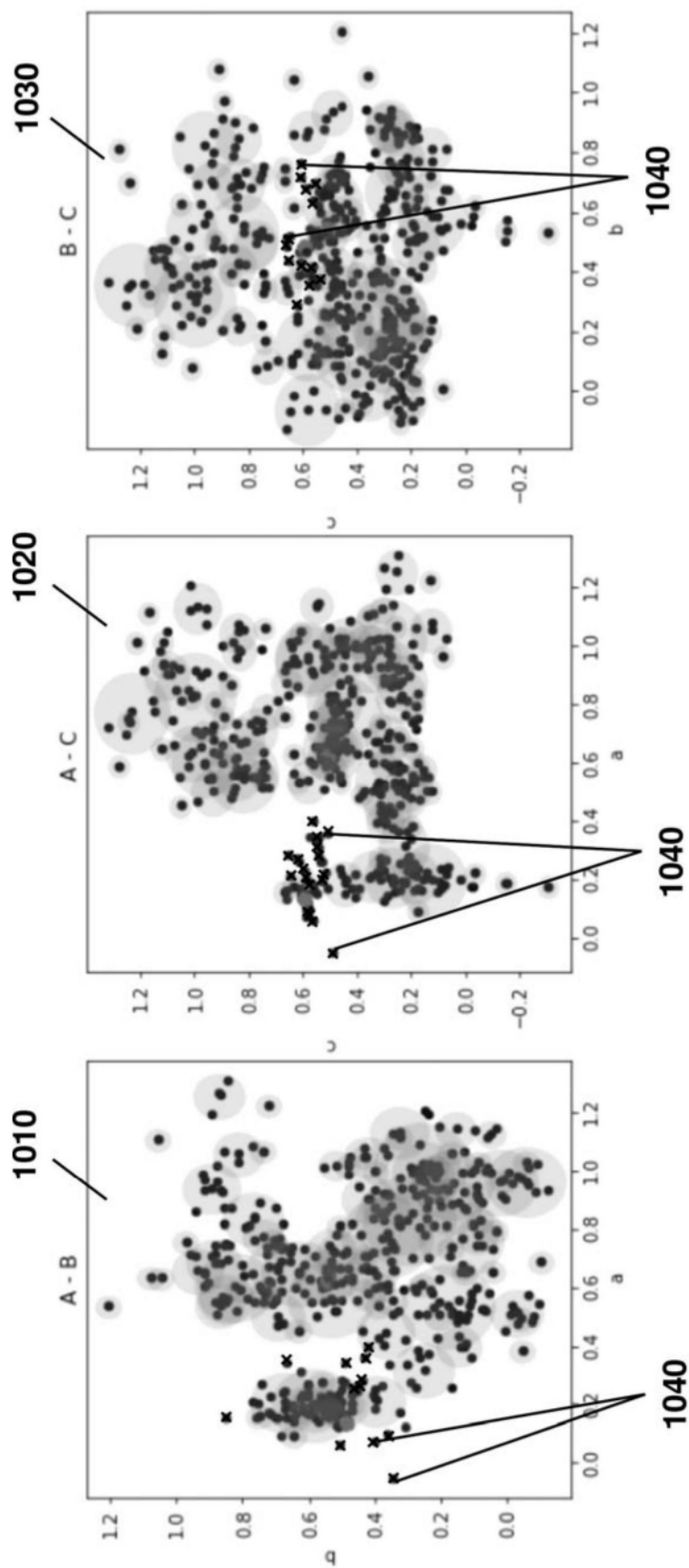


图10

		预测	
		良性的	异常的
实际的	良性的	真阳性 1110	假阴性 1120
	异常的	假阳性 1130	真阴性 1140

图11

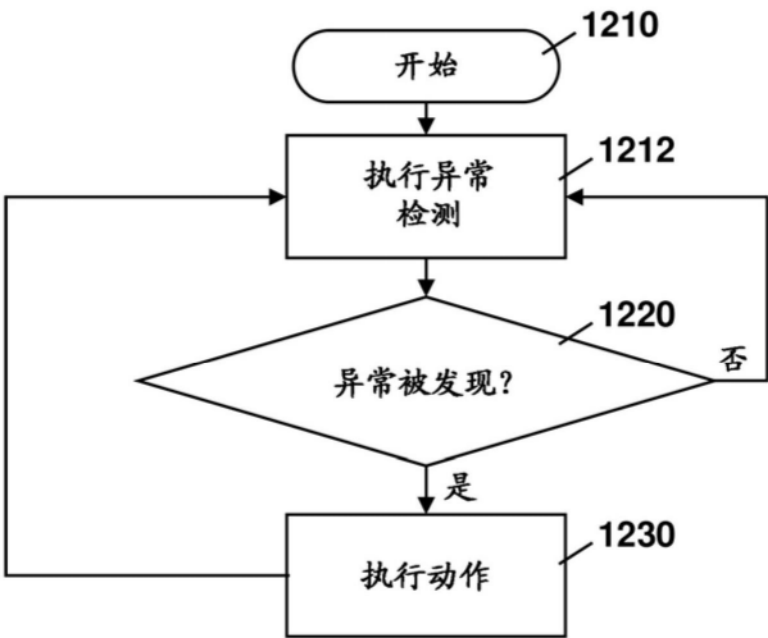


图12

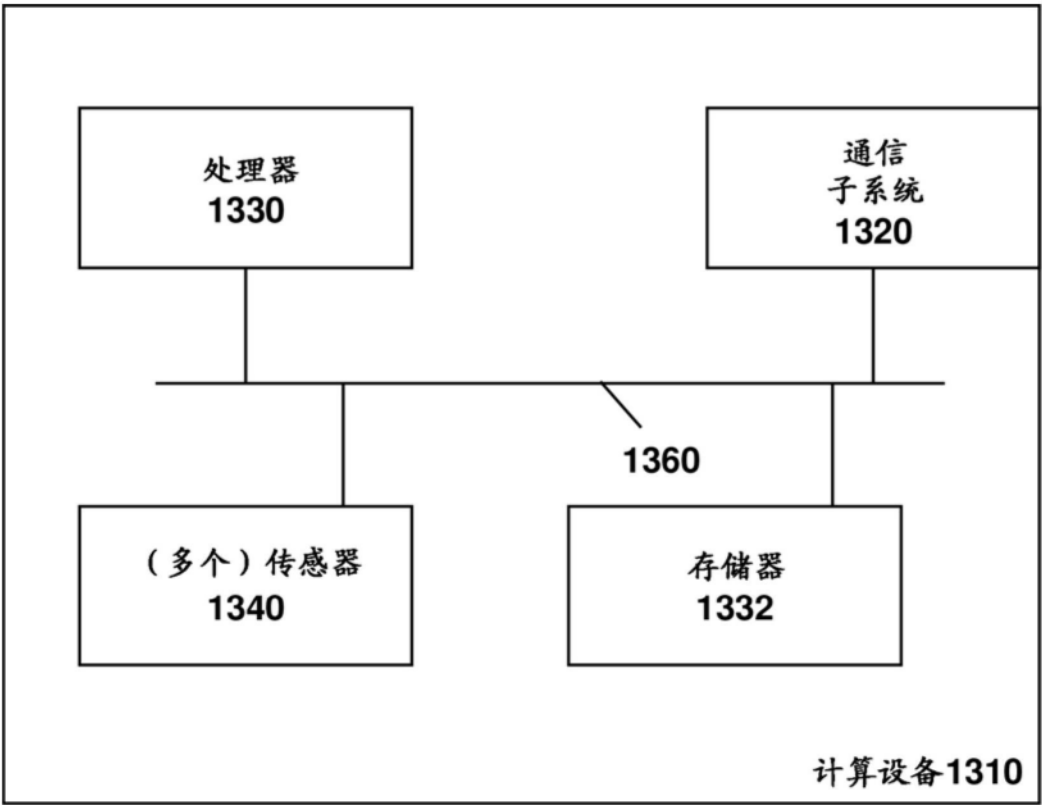


图13