(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 110389874 A
(43)申请公布日 2019.10.29

(21)申请号 201810359152.9

(22)申请日 2018.04.20

(71)申请人 比亚迪股份有限公司
地址 518118 广东省深圳市坪山新区比亚迪路3009号

(72)发明人 付瑞林

(74)专利代理机构 北京清亦华知识产权代理事务所(普通合伙) 11201
代理人 张润

(51)Int.Cl.
G06F 11/30(2006.01)

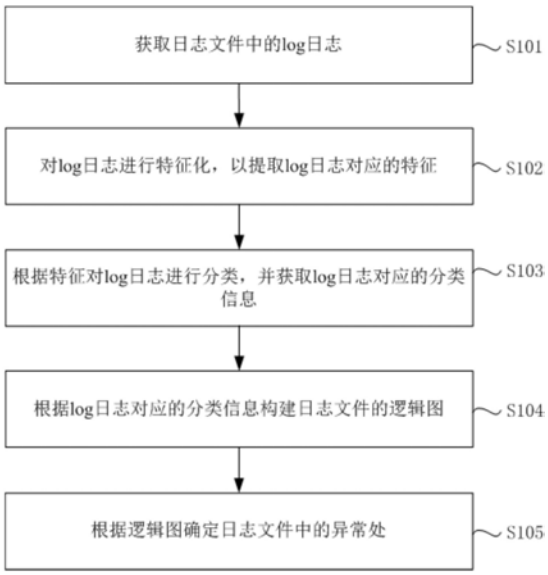
权利要求书2页 说明书9页 附图8页

(54)发明名称

日志文件异常检测方法和装置

(57)摘要

本发明公开了一种日志文件异常检测方法和装置，其中，方法包括：获取日志文件中的log日志；对log日志进行特征化，以提取log日志对应的特征；根据特征对log日志进行分类，并获取log日志对应的分类信息；根据log日志对应的分类信息构建日志文件的逻辑图；以及根据逻辑图确定日志文件中的异常处。本发明实施例的日志文件异常检测方法，通过获取日志文件中的log日志，再对log日志进行特征化，以提取log日志对应的特征，然后根据特征对log日志进行分类，并获取log日志对应的分类信息，再根据log日志对应的分类信息构建日志文件的逻辑图，以及根据逻辑图确定日志文件中的异常处，能够直观地体现出异常处，简单方便地确定系统运行中的问题，提高程序员的维护效率。



CN 110389874 A

1.一种日志文件异常检测方法,其特征在于,包括:

获取日志文件中的log日志;

对所述log日志进行特征化,以提取所述log日志对应的特征,所述特征包括第一编码信息和第二编码信息;

根据所述特征对所述log日志进行分类,并获取所述log日志对应的分类信息;

根据所述log日志对应的分类信息构建所述日志文件的逻辑图;以及

根据所述逻辑图确定所述日志文件中的异常处。

2.如权利要求1所述的方法,其特征在于,对所述log日志进行特征化,以提取所述log日志对应的特征,包括:

基于正则表达式提取所述log日志中的预定格式信息,并生成所述第一编码信息;

对提取预定格式信息后的所述log日志中的文本内容进行编码,以生成所述第二编码信息。

3.如权利要求1所述的方法,其特征在于,根据所述特征对所述log日志进行分类,并获取所述log日志对应的分类信息,包括:

获取所述log日志的第一编码信息的长度;

将所述第一编码信息的长度和所述第一编码信息输入至决策树,利用所述决策树进行分类,并确定所述log日志对应的第一分类编号;

将所述第二编码信息输入至所述决策树,利用所述决策树进行分类,并确定所述log日志对应的第二分类编号;

根据所述第一分类编号和所述第二分类编号生成所述log日志对应的分类信息。

4.如权利要求1所述的方法,其特征在于,根据所述log日志对应的分类信息构建所述日志文件的逻辑图,包括:

将所述log日志对应的分类信息作为所述逻辑图中的节点;

统计所述分类信息之间的跳转概率,并将所述跳转概率作为所述逻辑图中的边。

5.如权利要求4所述的方法,其特征在于,根据所述逻辑图确定所述日志文件中的异常处,包括:

将所述逻辑图中,边所对应的跳转概率与预设概率进行比对,确定跳转概率低于预设概率的边为异常处;或者

将所述逻辑图与历史逻辑图进行比对,确定所述逻辑图与所述历史逻辑图不一致的节点或边为异常处。

6.如权利要求1所述的方法,其特征在于,在根据所述逻辑图确定所述日志文件中的异常处之后,还包括:

生成异常提醒信息。

7.如权利要求3所述的方法,其特征在于,还包括:

在获取所述log日志的第一编码信息的长度之后,根据所述第一编码信息的长度计算所述log日志的长度离差值;

确定所述日志文件中长度离差值最大的log日志;

通过人工检测所述长度离差值最大的log日志是否异常。

8.一种日志文件异常检测装置,其特征在于,包括:

获取模块，用于获取日志文件中的log日志；

提取模块，用于对所述log日志进行特征化，以提取所述log日志对应的特征，所述特征包括第一编码信息和第二编码信息；

分类模块，用于根据所述特征对所述log日志进行分类，并获取所述log日志对应的分类信息；

构建模块，用于根据所述log日志对应的分类信息构建所述日志文件的逻辑图；以及

确定模块，用于根据所述逻辑图确定所述日志文件中的异常处。

9.如权利要求8所述的装置，其特征在于，所述提取模块，用于：

基于正则表达式提取所述log日志中的预定格式信息，并生成所述第一编码信息；

对提取预定格式信息后的所述log日志中的文本内容进行编码，以生成所述第二编码信息。

10.如权利要求8所述的方法，其特征在于，所述分类模块，包括：

获取单元，用于获取所述log日志的第一编码信息的长度；

第一分类单元，用于将所述第一编码信息的长度和所述第一编码信息输入至决策树，利用所述决策树进行分类，并确定所述log日志对应的第一分类编号；

第二分类单元，用于将所述第二编码信息输入至所述决策树，利用所述决策树进行分类，并确定所述log日志对应的第二分类编号；

生成单元，用于根据所述第一分类编号和所述第二分类编号生成所述log日志对应的分类信息。

11.如权利要求8所述的装置，其特征在于，所述构建模块，用于：

将所述log日志对应的分类信息作为所述逻辑图中的节点；

统计所述分类信息之间的跳转概率，并将所述跳转概率作为所述逻辑图中的边。

12.如权利要求11所述的装置，其特征在于，所述确定模块，用于：

将所述逻辑图中，边所对应的跳转概率与预设概率进行比对，确定跳转概率低于预设概率的边为异常处；或者

将所述逻辑图与历史逻辑图进行比对，确定所述逻辑图与所述历史逻辑图不一致的节点或边为异常处。

13.如权利要求8所述的装置，其特征在于，还包括：

提醒模块，用于在根据所述逻辑图确定所述日志文件中的异常处之后，生成异常提醒信息。

14.如权利要求10所述的装置，其特征在于，所述分类模块，还包括：

计算单元，用于在获取所述log日志的第一编码信息的长度之后，根据所述第一编码信息的长度计算所述log日志的长度离差值；

确定单元，用于确定所述日志文件中长度离差值最大的log日志；

检测单元，用于通过人工检测所述长度离差值最大的log日志是否异常。

15.一种非临时性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如权利要求1-7任一项所述的日志文件异常检测方法。

16.一种终端，包括处理器、存储器及存储在所述存储器上并可在所述处理器上运行的计算机程序，所述处理器用于执行如权利要求1-7任一项所述的日志文件异常检测方法。

日志文件异常检测方法和装置

技术领域

[0001] 本发明涉及信息处理技术领域,尤其涉及一种日志文件异常检测方法和装置。

背景技术

[0002] 随着信息化时代的来临,越来越多的领域都已开始使用智能控制系统来代替传统的人工控制方式,来实现难以解决的复杂系统的控制问题。例如:轨道交通中的ATS(Automatic Train Supervision,自动列车监控系统)等复杂系统中,多个运算主体和多种程序会按照自身的逻辑不间断运行。在系统的研发和调试过程中,非常容易出现系统表现异常,但关键问题无从查起的状况。目前,主要采取记录系统log日志的方式,通过log日志中出现的标志信息标定出产生问题的位置,进而实现错误定位。然而,通过上述方法,需要程序员基于自身经验,人工定位并分析系统的问题,不够方便、直观,效率低。

发明内容

[0003] 本发明提供一种日志文件异常检测方法和装置,以解决上述技术问题中的至少一个。

[0004] 本发明实施例提供一种日志文件异常检测方法,包括:

[0005] 获取日志文件中的log日志;

[0006] 对所述log日志进行特征化,以提取所述log日志对应的特征,所述特征包括第一编码信息和第二编码信息;

[0007] 根据所述特征对所述log日志进行分类,并获取所述log日志对应的分类信息;

[0008] 根据所述log日志对应的分类信息构建所述日志文件的逻辑图;以及

[0009] 根据所述逻辑图确定所述日志文件中的异常处。

[0010] 可选的,对所述log日志进行特征化,以提取所述log日志对应的特征,包括:

[0011] 基于正则表达式提取所述log日志中的预定格式信息,并生成所述第一编码信息;

[0012] 对提取预定格式信息后的所述log日志中的文本内容进行编码,以生成所述第二编码信息。

[0013] 可选的,根据所述特征对所述log日志进行分类,并获取所述log日志对应的分类信息,包括:

[0014] 获取所述log日志的第一编码信息的长度;

[0015] 将所述第一编码信息的长度和所述第一编码信息输入至决策树,利用所述决策树进行分类,并确定所述log日志对应的第一分类编号;

[0016] 将所述第二编码信息输入至所述决策树,利用所述决策树进行分类,并确定所述log日志对应的第二分类编号;

[0017] 根据所述第一分类编号和所述第二分类编号生成所述log日志对应的分类信息。

[0018] 可选的,根据所述log日志对应的分类信息构建所述日志文件的逻辑图,包括:

[0019] 将所述log日志对应的分类信息作为所述逻辑图中的节点;

[0020] 统计所述分类信息之间的跳转概率，并将所述跳转概率作为所述逻辑图中的边。

[0021] 可选的，根据所述逻辑图确定所述日志文件中的异常处，包括：

[0022] 将所述逻辑图中，边所对应的跳转概率与预设概率进行比对，确定跳转概率低于预设概率的边为异常处；或者

[0023] 将所述逻辑图与历史逻辑图进行比对，确定所述逻辑图与所述历史逻辑图不一致的节点或边为异常处。

[0024] 可选的，在根据所述逻辑图确定所述日志文件中的异常处之后，还包括：

[0025] 生成异常提醒信息。

[0026] 可选的，方法还包括：

[0027] 在获取所述log日志的第一编码信息的长度之后，根据所述第一编码信息的长度计算所述log日志的长度离差值；

[0028] 确定所述日志文件中长度离差值最大的log日志；

[0029] 通过人工检测所述长度离差值最大的log日志是否异常。

[0030] 本发明另一实施例提供一种日志文件异常检测装置，包括：

[0031] 获取模块，用于获取日志文件中的log日志；

[0032] 提取模块，用于对所述log日志进行特征化，以提取所述log日志对应的特征，所述特征包括第一编码信息和第二编码信息；

[0033] 分类模块，用于根据所述特征对所述log日志进行分类，并获取所述log日志对应的分类信息；

[0034] 构建模块，用于根据所述log日志对应的分类信息构建所述日志文件的逻辑图；以及

[0035] 确定模块，用于根据所述逻辑图确定所述日志文件中的异常处。

[0036] 可选的，所述提取模块，用于：

[0037] 基于正则表达式提取所述log日志中的预定格式信息，并生成所述第一编码信息；

[0038] 对提取预定格式信息后的所述log日志中的文本内容进行编码，以生成所述第二编码信息。

[0039] 可选的，所述分类模块，包括：

[0040] 获取单元，用于获取所述log日志的第一编码信息的长度；

[0041] 第一分类单元，用于将所述第一编码信息的长度和所述第一编码信息输入至决策树，利用所述决策树进行分类，并确定所述log日志对应的第一分类编号；

[0042] 第二分类单元，用于将所述第二编码信息输入至所述决策树，利用所述决策树进行分类，并确定所述log日志对应的第二分类编号；

[0043] 生成单元，用于根据所述第一分类编号和所述第二分类编号生成所述log日志对应的分类信息。

[0044] 可选的，所述构建模块，用于：

[0045] 将所述log日志对应的分类信息作为所述逻辑图中的节点；

[0046] 统计所述分类信息之间的跳转概率，并将所述跳转概率作为所述逻辑图中的边。

[0047] 可选的，所述确定模块，用于：

[0048] 将所述逻辑图中，边所对应的跳转概率与预设概率进行比对，确定跳转概率低于

预设概率的边为异常处;或者

[0049] 将所述逻辑图与历史逻辑图进行比对,确定所述逻辑图与所述历史逻辑图不一致的节点或边为异常处。

[0050] 可选的,还包括:

[0051] 提醒模块,用于在根据所述逻辑图确定所述日志文件中的异常处之后,生成异常提醒信息。

[0052] 可选的,装置还包括:

[0053] 推荐模块,用于基于预设规则向直播平台中的观众推荐分类后的优质主播。

[0054] 可选的,所述分类模块,还包括:

[0055] 计算单元,用于在获取所述log日志的第一编码信息的长度之后,根据所述第一编码信息的长度计算所述log日志的长度离差值;

[0056] 确定单元,用于确定所述日志文件中长度离差值最大的log日志;

[0057] 检测单元,用于通过人工检测所述长度离差值最大的log日志是否异常。

[0058] 本发明还一实施例提供一种非临时性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如本发明第一方面实施例所述的日志文件异常检测方法。

[0059] 本发明又一实施例提供一种电子设备,包括处理器、存储器及存储在所述存储器上并可在所述处理器上运行的计算机程序,所述处理器用于执行本发明第一方面实施例所述的日志文件异常检测方法。

[0060] 本发明实施例提供的技术方案可以包括以下有益效果:

[0061] 通过获取日志文件中的log日志,再对log日志进行特征化,以提取log日志对应的特征,然后根据特征对log日志进行分类,并获取log日志对应的分类信息,再根据log日志对应的分类信息构建日志文件的逻辑图,以及根据逻辑图确定日志文件中的异常处,能够直观地体现出异常处,简单方便地确定系统运行中的问题,提高程序员的维护效率。

[0062] 本发明附加的方面和优点将在下面的描述中部分给出,部分将从下面的描述中变得明显,或通过本发明的实践了解到。

附图说明

[0063] 本发明上述的和/或附加的方面和优点从下面结合附图对实施例的描述中将变得明显和容易理解,其中:

[0064] 图1是根据本发明一个实施例的日志文件异常检测方法的流程图;

[0065] 图2是根据本发明一个实施例的获取log日志对应的分类信息的流程图;

[0066] 图3是根据本发明一个实施例的日志文件的逻辑图的效果示意图;

[0067] 图4是根据本发明另一个实施例的日志文件异常检测方法的流程图;

[0068] 图5是根据本发明又一个实施例的日志文件异常检测方法的流程图;

[0069] 图6是根据本发明一个实施例的日志文件异常检测装置的结构框图;

[0070] 图7是根据本发明另一个实施例的日志文件异常检测装置的结构框图;

[0071] 图8是根据本发明又一个实施例的日志文件异常检测装置的结构框图;

[0072] 图9是根据本发明一个实施例的电子设备的结构框图。

具体实施方式

[0073] 下面详细描述本发明的实施例，所述实施例的示例在附图中示出，其中自始至终相同或类似的标号表示相同或类似的元件或具有相同或类似功能的元件。下面通过参考附图描述的实施例是示例性的，旨在用于解释本发明，而不能理解为对本发明的限制。

[0074] 下面参考附图描述本发明实施例的日志文件异常检测方法和装置。

[0075] 图1是根据本发明一个实施例的日志文件异常检测方法的流程图。

[0076] 如图1所示，该日志文件异常检测方法包括：

[0077] S101，获取日志文件中的log日志。

[0078] 随着信息化时代的来临，控制系统也越来越复杂。在系统的运维过程中，程序员主要通过系统的log日志，来查找、定位并分析出系统运行时产生的问题。然而，log日志仅仅是对问题进行定位，并不能提供更直观的一些数据帮助程序员进行分析。因此，本发明提出一种日志文件异常检测方法，实现快速检测系统运行中存在的问题。

[0079] 在本发明的一个实施例中，可获取日志文件中的log日志。在系统运行过程中，每天都会产生大量的日志文件，每个日志文件都会包含多个条目的log日志。而这些日志文件通常会保存到日志服务器中。因此，在需要对日志文件进行异常检测的时候，可从日志服务器中提取待分析的日志文件。

[0080] S102，对log日志进行特征化，以提取log日志对应的特征。

[0081] 其中，特征可包括第一编码信息和第二编码信息。

[0082] 具体地，可基于正则表达式提取log日志中的预定格式信息，并生成第一编码信息，然后再对提取预定格式信息后的log日志中的文本内容进行编码，以生成第二编码信息。其中，log日志中通常会包括进程信息、源程序信息、时间戳等具有一定格式的信息，同时这些信息的内容可以为程序员提供数据支持。因此，对log日志进行特征化，就是提取特征并对特征进行编码的过程。

[0083] 举例来说：某个条目的log日志为：

[0084] “u’2017-09-23 06:37:57.270[main]info

[0085] o.s.c.a.annotationconfigapplicationcontext-refreshing

[0086]
org.springframework.context.annotation.annotationconfigapplicationcontext@7637f22:startup date[sat sep 23 06:37:57cst 2017];root of context hierarchy\n’”

[0087] 经过正则表达式提取后，获得的信息如下：[2017-09-23 06:37:57.270]、[[main]]、[info]以及[o.s.c.a.annotationconfigapplicationcontext]，可分别对其进行编码为01、02、03以及04。此为第一编码信息。

[0088] 而在此之后，可对log日志中的文本内容进行编码，生成第二编码信息。其目的主要在于提取文本内容中的两种关键特征，即词频和词序。例如：某log日志为

[0089] sessionid:1,remoteaddr:/172.24.0.18:49366{"header_info":{"inface_type":,"send_vender":,"receive_vender":……"t_stamp":{"sec":,,"usec":"zzl":,}，其中，"inface_type"、"send_vender"、"receive_vender"、"t_stamp"等属于同一

级别的特征，可编码为1；而"sec"、"usec"、"zzl"为"t_stamp"下一级别的特征，为了做区分，可编码为0。因此，生成的编码信息(第二编码信息)为1111000。

[0090] 当然，还可以按照词序的编码方式进行编码。可将log日志中的header_info、inface_type、receive_vender、remoteaddr、sec、sessionid、send_vender、t_stamp、usec、zzl等，基于预设的编码表，按照顺序对其进行编码，则可得到编码信息(第二编码信息)7 4 1 2 6 3 8 5 910。其中，预设的编码表可如表1所示。

[0091]

header_info	1
inface_type	2
receive_vender	3
remoteAddr	4
sec	5
send_vender	6
sessionId	7
t_stamp	8
usec	9
zzl	10

[0092] 表1

[0093] 当log日志中出现了此前未出现过的词组时，可先采用代码-1，-2，……按顺序增加的方式来表示这些词组。在当日运营结束后，或系统其他停运时间，可对编码表进行更新。

[0094] S103，根据特征对log日志进行分类，并获取log日志对应的分类信息。

[0095] 在提取log日志对应的特征之后，便可以根据特征对log日志进行分类，并获取log日志对应的分类信息。

[0096] 具体地，如图2所示，可包括以下步骤：

[0097] S201，获取log日志的第一编码信息的长度。

[0098] S202，将第一编码信息的长度和第一编码信息输入至决策树，利用决策树进行分类，从而确定log日志对应的第一分类编号。

[0099] S203，将第二编码信息输入至决策树，利用决策树进行分类，并确定log日志对应的第二分类编号。

[0100] S204，根据第一分类编号和第二分类编号生成log日志对应的分类信息。

[0101] 在分类时，可以使用决策树或者KNN(k-nearestneighbor，邻近分类算法)两种算法进行聚类。

[0102] 在本实施例中，主要采用决策树算法来对log日志进行分类。原因在于系统中的大部分的log日志的信息长度都不一样，可以使用词序的编码方式配以决策树的分叉方式，可以有效降低运算量，提高运算速度。

[0103] 具体地，可计算每个log日志的特征编码长度$l_i, i \in \{1,2,\cdots,N\}$，即第一编码信息的长度。其中，n为算log日志的总数。可将第一编码信息的长度和第一编码信息输入至决策树，通过决策树进行分叉，最终满足条件的节点，即为log日志对应的第一分类编号。此时，

每个log日志已经被分配了一个分类节点。在相同分类节点上的log日志，具有同样的编码长度、源信息、进城信息等。之后，可针对每个分类节点，将log日志的顺序编码(第二编码信息)作为特征，输入至决策树进行分叉，直至每个分类节点的叶子节点中的log日志都包含有同样的编码信息，即分类结束。通过上述方法，每个log日志均都被分配了两个分类编号，特征信息的分类编号(第一分类编号)和编码信息的分类编号(第二分类编号)，这两个分类编号共同确定了log日志所属的分类信息。格式如下：(第二分类编号.第一分类编号)，例如(-1.0)、(3204.0)等等。

[0104] S104，根据log日志对应的分类信息构建日志文件的逻辑图。

[0105] 在生成log日志对应的分类信息之后，便可以根据log日志对应的分类信息构建日志文件的逻辑图。具体地，可将log日志对应的分类信息作为逻辑图中的节点，然后统计分类信息之间的跳转概率，并将跳转概率作为逻辑图中的边。其中，每个分类信息都包括流入和流出。流入节点为日志文件的第一条log日志所属的分类，流出节点为日志文件的最后一条log日志所属的分类。例如：针对分类1cl_1和分类3cl_3，分类1cl_1流入分类3cl_3的概率可表示为$Pr\{cl_1 \rightarrow cl_3 cl_1\} = p_{1 \rightarrow 3}$，也即是分类1出现后分类3马上出现的概率。这样就形成了节点1到节点3的一条边。通过统计分析出日志文件中所有节点的流入和流出，便可形成如图3所示的完整的逻辑图。从图3可以看出，节点(-1,0)流入到节点(211.0)的概率为0.02，而反向的，节点(211.0)流入到节点(-1,0)的概率则为0.0012。

[0106] 构建好的逻辑图往往能够直观地反映出系统中运行的层级、并行及通信关系等。如：预定格式信息中的进程信息，能够反映出产生log日志对应的程序来自网络中的哪个计算机或者在计算机的哪个进程，通常能够反映出程序的并发情况甚至系统的归属信息。通过逻辑图的构建，便可以为程序员呈现最为细致的程序流转过程，令负责系统调试的程序员能够对各个运算主体的log日志之间的流转关系有整体的认识，为程序员分析log日志提供帮助。

[0107] S105，根据逻辑图确定日志文件中的异常处。

[0108] 在构建逻辑图成功之后，可根据逻辑图确定日志文件中的异常处。

[0109] 具体地，可将逻辑图中，边所对应的跳转概率与预设概率进行比对，确定跳转概率低于预设概率的边为异常处。例如：某条边的跳转概率为0.0012，低于了预设概率0.01，那么说明由这条边的起始节点跳转到这条边的目的节点，即发生对应的事件的概率过低，则说明此处异常。

[0110] 当然，也可以将逻辑图与历史逻辑图进行比对，确定逻辑图与历史逻辑图不一致的节点或边为异常处。例如，今日的逻辑图中，出现了昨日的逻辑图中未出现的跳转关系，可能暗示了某一部分程序没在运行，导致系统异常。可直观地从图中便可看出异常处。

[0111] 本发明实施例的日志文件异常检测方法，通过获取日志文件中的log日志，再对log日志进行特征化，以提取log日志对应的特征，然后根据特征对log日志进行分类，并获取log日志对应的分类信息，再根据log日志对应的分类信息构建日志文件的逻辑图，以及根据逻辑图确定日志文件中的异常处，能够直观地体现出异常处，简单方便地确定系统运行中的问题，提高程序员的维护效率。

[0112] 如图4所示，该日志文件异常检测方法还可包括：

[0113] S106，生成异常提醒信息。

[0114] 在根据逻辑图确定日志文件中的异常处之后，可生成异常提醒信息，从而对程序员进行提醒，帮助程序员能够及时处理系统运行过程中的问题。

[0115] 如图5所示，该日志文件异常检测方法还可包括：

[0116] S205，根据第一编码信息的长度计算log日志的长度离差值。

[0117] 在计算每个log日志的第一编码信息的长度之后，还可计算出第一编码信息的长度的均值μ，每两个log日志的第一编码信息的长度的方差σ^2，进而计算出log日志的长度离差$\frac{(l_i-\mu)^2}{\sigma^2}$。

[0118] S206，确定日志文件中长度离差值最大的log日志。

[0119] S207，通过人工检测长度离差值最大的log日志是否异常。

[0120] 大多数log日志的长度都会在一定范围内。此处的一定范围指的是一个认知限度，如一条log日志有400行的内容，实际情况很可能是多个事件堆积在这一条log日志中。因此，可抽检日志文件中长度离差值最大的log日志，由程序员人工确定此log日志是否异常。如果程序员认为此log日志异常，则可以对log日志进行分析，处理相应的故障，从而保证系统正常运行。

[0121] 为了实现上述实施例，本发明还提出了一种日志文件异常检测装置，图6是根据本发明一个实施例的日志文件异常检测装置的结构框图，如图6所示，该装置包括获取模块610、提取模块620、分类模块630、构建模块640和确定模块650。

[0122] 其中，获取模块610，用于获取日志文件中的log日志。

[0123] 提取模块620，用于对log日志进行特征化，以提取log日志对应的特征，特征包括第一编码信息和第二编码信息。

[0124] 分类模块630，用于根据特征对log日志进行分类，并获取log日志对应的分类信息。

[0125] 构建模块640，用于根据log日志对应的分类信息构建日志文件的逻辑图。

[0126] 确定模块650，用于根据逻辑图确定日志文件中的异常处。

[0127] 其中，分类模块630进一步包括获取单元631、第一分类单元632、第二分类单元633和生成单元634。

[0128] 获取单元631，用于获取log日志的第一编码信息的长度。

[0129] 第一分类单元632，用于将第一编码信息的长度和第一编码信息输入至决策树，利用决策树进行分类，并确定log日志对应的第一分类编号。

[0130] 第二分类单元633，用于将第二编码信息输入至决策树，利用决策树进行分类，并确定log日志对应的第二分类编号。

[0131] 生成单元634，用于根据第一分类编号和第二分类编号生成log日志对应的分类信息。

[0132] 如图7所示，日志文件异常检测装置还可包括提醒模块660。

[0133] 提醒模块660，用于在根据逻辑图确定日志文件中的异常处之后，生成异常提醒信息。

[0134] 如图8所示，分类模块630还可包括计算单元635、确定单元636和检测单元637。

[0135] 其中，计算单元635，用于在获取log日志的第一编码信息的长度之后，根据第一编

码信息的长度计算log日志的长度离差值。

[0136] 确定单元636，用于确定日志文件中长度离差值最大的log日志。

[0137] 检测单元637，用于通过人工检测长度离差值最大的log日志是否异常。

[0138] 需要说明的是，前述对日志文件异常检测方法的解释说明，也适用于本发明实施例的日志文件异常检测装置，本发明实施例中未公布的细节，在此不再赘述。

[0139] 本发明实施例的日志文件异常检测装置，通过获取日志文件中的log日志，再对log日志进行特征化，以提取log日志对应的特征，然后根据特征对log日志进行分类，并获取log日志对应的分类信息，再根据log日志对应的分类信息构建日志文件的逻辑图，以及根据逻辑图确定日志文件中的异常处，能够直观地体现出异常处，简单方便地确定系统运行中的问题，提高程序员的维护效率。

[0140] 为了实现上述实施例，本发明还提出了一种电子设备。

[0141] 如图9所示，电子设备900包括处理器910、存储器920及存储在存储器920上并可在处理器910上运行的计算机程序901，处理器910用于执行本发明第一方面实施例的日志文件异常检测方法。

[0142] 例如，计算机程序可被处理器执行以完成以下步骤的日志文件异常检测方法：

[0143] S101’，获取日志文件中的log日志。

[0144] S102’，对log日志进行特征化，以提取log日志对应的特征，特征包括第一编码信息和第二编码信息。

[0145] S103’，根据特征对log日志进行分类，并获取log日志对应的分类信息。

[0146] S104’，根据log日志对应的分类信息构建日志文件的逻辑图。

[0147] S105’，根据逻辑图确定日志文件中的异常处。

[0148] 在本说明书的描述中，参考术语“一个实施例”、“一些实施例”、“示例”、“具体示例”、或“一些示例”等的描述意指结合该实施例或示例描述的具体特征、结构、材料或者特点包含于本发明的至少一个实施例或示例中。在本说明书中，对上述术语的示意性表述不必须针对的是相同的实施例或示例。而且，描述的具体特征、结构、材料或者特点可以在任一个或多个实施例或示例中以合适的方式结合。此外，在不相互矛盾的情况下，本领域的技术人员可以将本说明书中描述的不同实施例或示例以及不同实施例或示例的特征进行结合和组合。

[0149] 此外，术语“第一”、“第二”仅用于描述目的，而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量。由此，限定有“第一”、“第二”的特征可以明示或者隐含地包括至少一个该特征。在本发明的描述中，“多个”的含义是至少两个，例如两个，三个等，除非另有明确具体的限定。

[0150] 流程图中或在此以其他方式描述的任何过程或方法描述可以被理解为，表示包括一个或更多个用于实现特定逻辑功能或过程的步骤的可执行指令的代码的模块、片段或部分，并且本发明的优选实施方式的范围包括另外的实现，其中可以不按所示出或讨论的顺序，包括根据所涉及的功能按基本同时的方式或按相反的顺序，来执行功能，这应被本发明的实施例所属技术领域的技术人员所理解。

[0151] 在流程图中表示或在此以其他方式描述的逻辑和/或步骤，例如，可以被认为是用于实现逻辑功能的可执行指令的定序列表，可以具体实现在任何计算机可读介质中，以供

指令执行系统、装置或设备(如基于计算机的系统、包括处理器的系统或其他可以从指令执行系统、装置或设备取指令并执行指令的系统)使用,或结合这些指令执行系统、装置或设备而使用。就本说明书而言,"计算机可读介质"可以是任何可以包含、存储、通信、传播或传输程序以供指令执行系统、装置或设备或结合这些指令执行系统、装置或设备而使用的装置。计算机可读介质的更具体的示例(非穷尽性列表)包括以下:具有一个或多个布线的电连接部(电子装置),便携式计算机盘盒(磁装置),随机存取存储器(ram),只读存储器(rom),可擦除可编辑只读存储器(eprom或闪速存储器),光纤装置,以及便携式光盘只读存储器(cdrom)。另外,计算机可读介质甚至可以是可在其上打印程序的纸或其他合适的介质,因为可以例如通过对纸或其他介质进行光学扫描,接着进行编辑、解译或必要时以其他合适方式进行处理来以电子方式获得程序,然后将其存储在计算机存储器中。

[0152] 应当理解,本发明的各部分可以用硬件、软件、固件或它们的组合来实现。在上述实施方式中,多个步骤或方法可以用存储在存储器中且由合适的指令执行系统执行的软件或固件来实现。例如,如果用硬件来实现,和在另一实施方式中一样,可用本领域公知的下列技术中的任一项或他们的组合来实现:具有用于对数据信号实现逻辑功能的逻辑门电路的离散逻辑电路,具有合适的组合逻辑门电路的专用集成电路,可编程门阵列(pga),现场可编程门阵列(fpga)等。

[0153] 本技术领域的普通技术人员可以理解实现上述实施例方法携带的全部或部分步骤是可以通过程序来指令相关的硬件完成,的程序可以存储于一种计算机可读存储介质中,该程序在执行时,包括方法实施例的步骤之一或其组合。

[0154] 此外,在本发明各个实施例中的各功能单元可以集成在一个处理模块中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个模块中。上述集成的模块既可以采用硬件的形式实现,也可以采用软件功能模块的形式实现。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时,也可以存储在一个计算机可读取存储介质中。

[0155] 上述提到的存储介质可以是只读存储器,磁盘或光盘等。尽管上面已经示出和描述了本发明的实施例,可以理解的是,上述实施例是示例性的,不能理解为对本发明的限制,本领域的普通技术人员在本发明的范围内可以对上述实施例进行变化、修改、替换和变型。

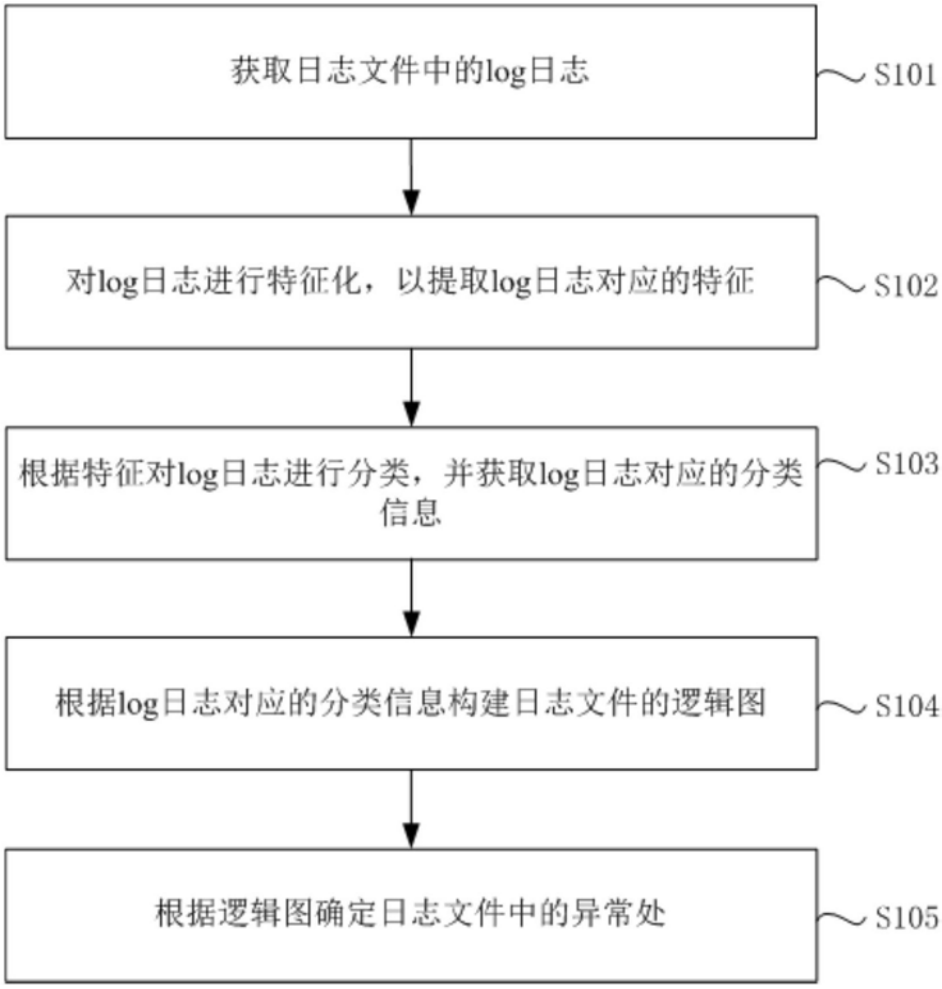

获取日志文件中的log日志
S101
对log日志进行特征化，以提取log日志对应的特征
S102
根据特征对log日志进行分类，并获取log日志对应的分类信息
S103
根据log日志对应的分类信息构建日志文件的逻辑图
S104
根据逻辑图确定日志文件中的异常处
S105

图1

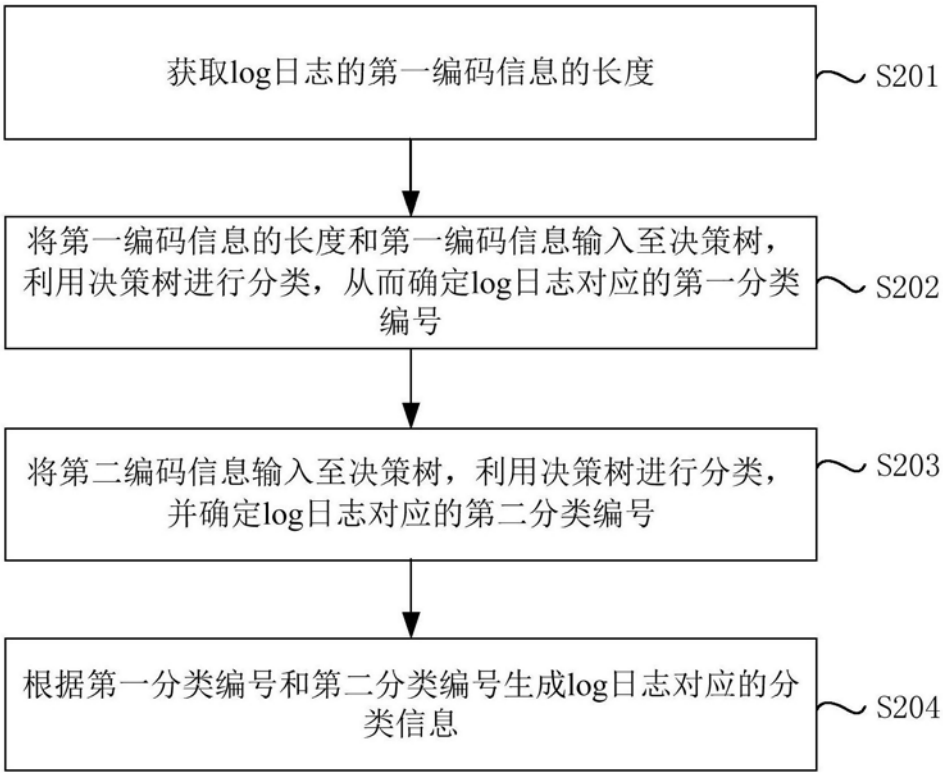

获取log日志的第一编码信息的长度
S201
将第一编码信息的长度和第一编码信息输入至决策树，利用决策树进行分类，从而确定log日志对应的第一分类编号
S202
将第二编码信息输入至决策树，利用决策树进行分类，并确定log日志对应的第二分类编号
S203
根据第一分类编号和第二分类编号生成log日志对应的分类信息
S204

图2

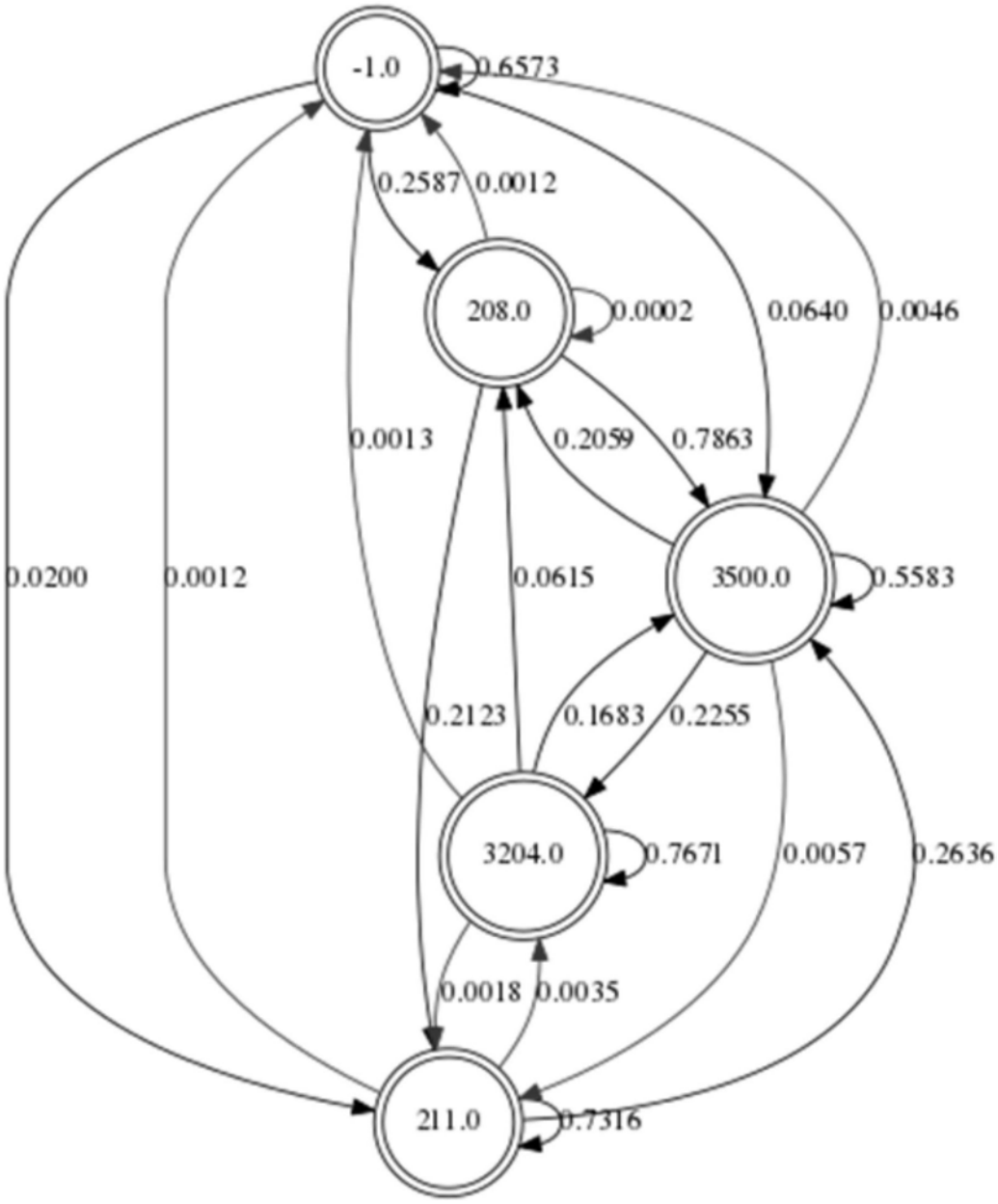

-1.0
0.6573
0.2587
0.0012
208.0
0.0002
0.0640
0.0046
0.0013
0.2059
0.7863
0.0200
0.0012
0.0615
3500.0
0.5583
0.2123
0.1683
0.2255
3204.0
0.7671
0.0057
0.2636
0.0018
0.0035
211.0
0.7316

图3

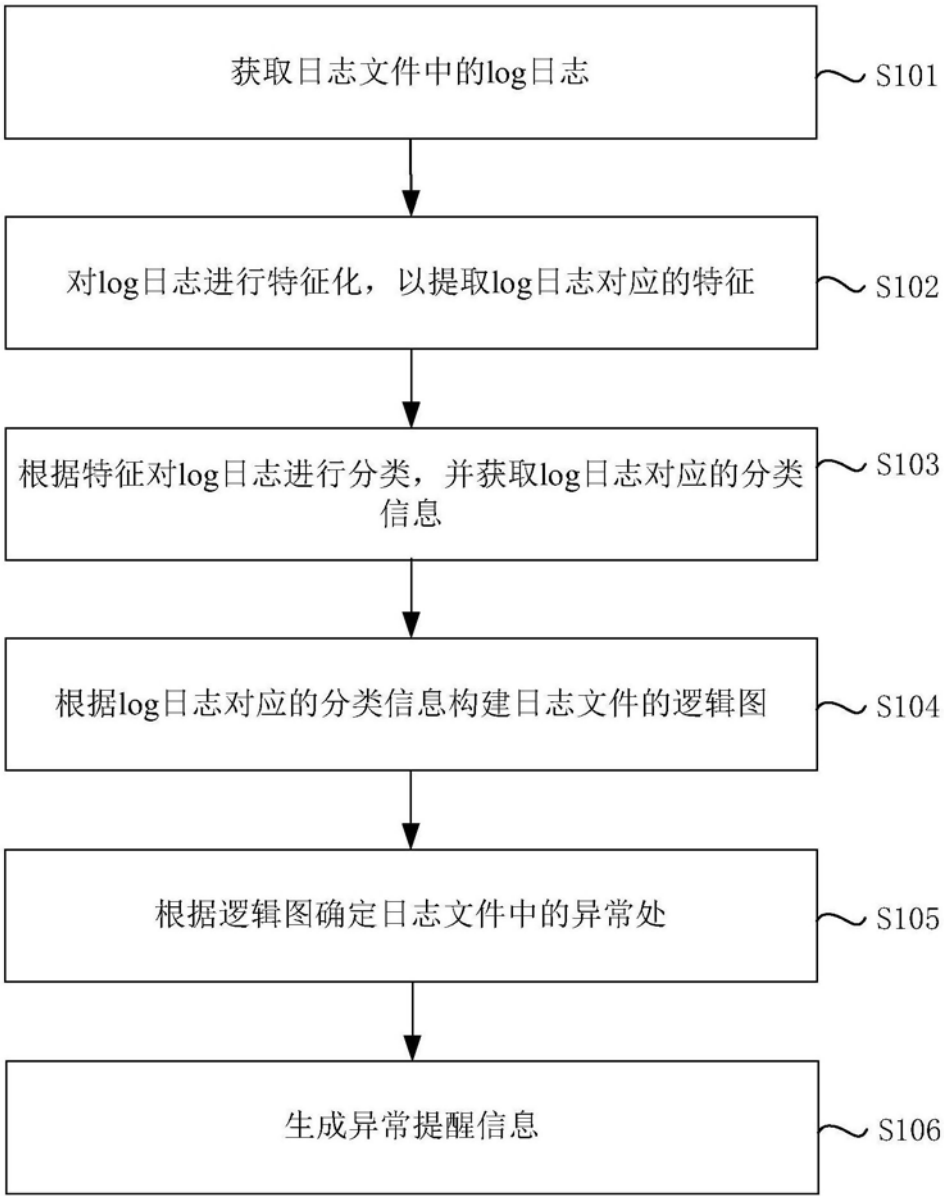

获取日志文件中的log日志
S101
对log日志进行特征化，以提取log日志对应的特征
S102
根据特征对log日志进行分类，并获取log日志对应的分类信息
S103
根据log日志对应的分类信息构建日志文件的逻辑图
S104
根据逻辑图确定日志文件中的异常处
S105
生成异常提醒信息
S106

图4

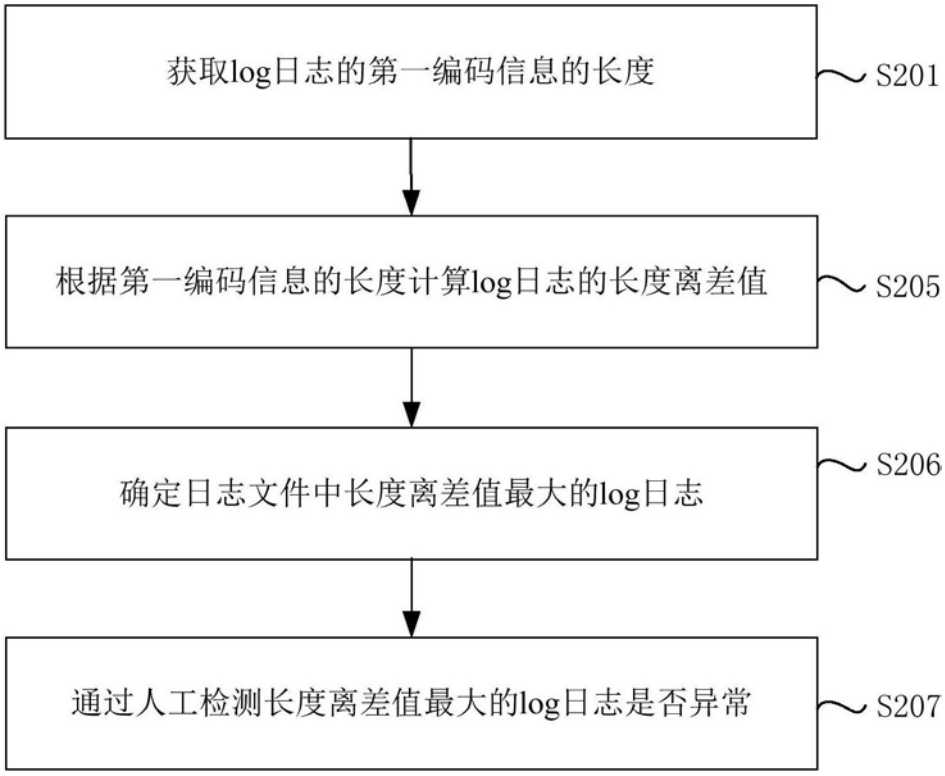

获取log日志的第一编码信息的长度
S201
根据第一编码信息的长度计算log日志的长度离差值
S205
确定日志文件中长度离差值最大的log日志
S206
通过人工检测长度离差值最大的log日志是否异常
S207

图5

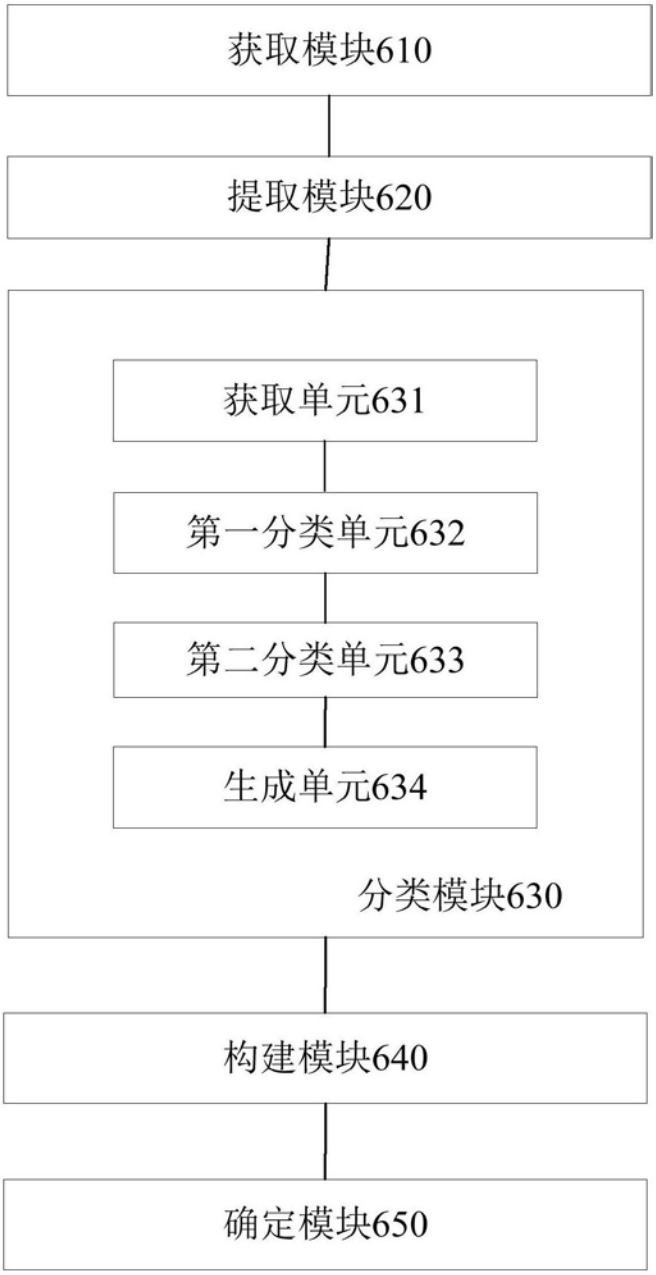

获取模块610
提取模块620
获取单元631
第一分类单元632
第二分类单元633
生成单元634
分类模块630
构建模块640
确定模块650

图6

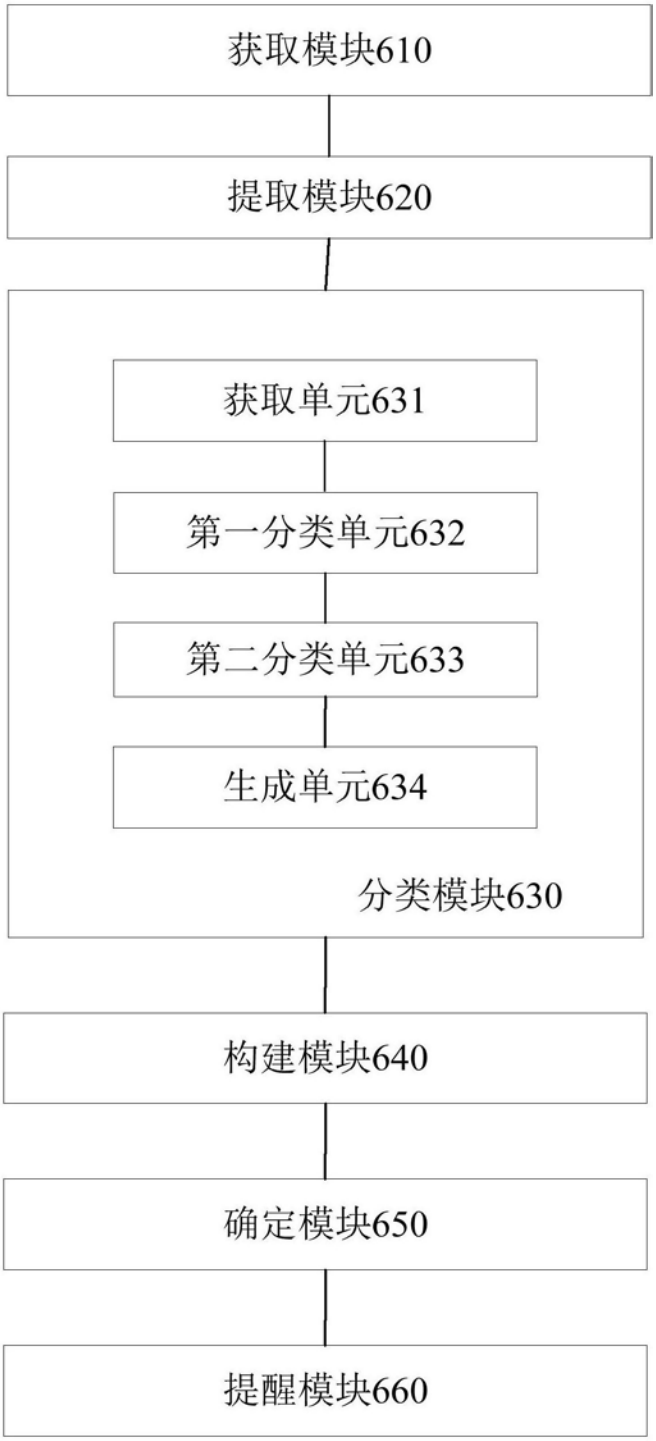

获取模块610
提取模块620
获取单元631
第一分类单元632
第二分类单元633
生成单元634
分类模块630
构建模块640
确定模块650
提醒模块660

图7

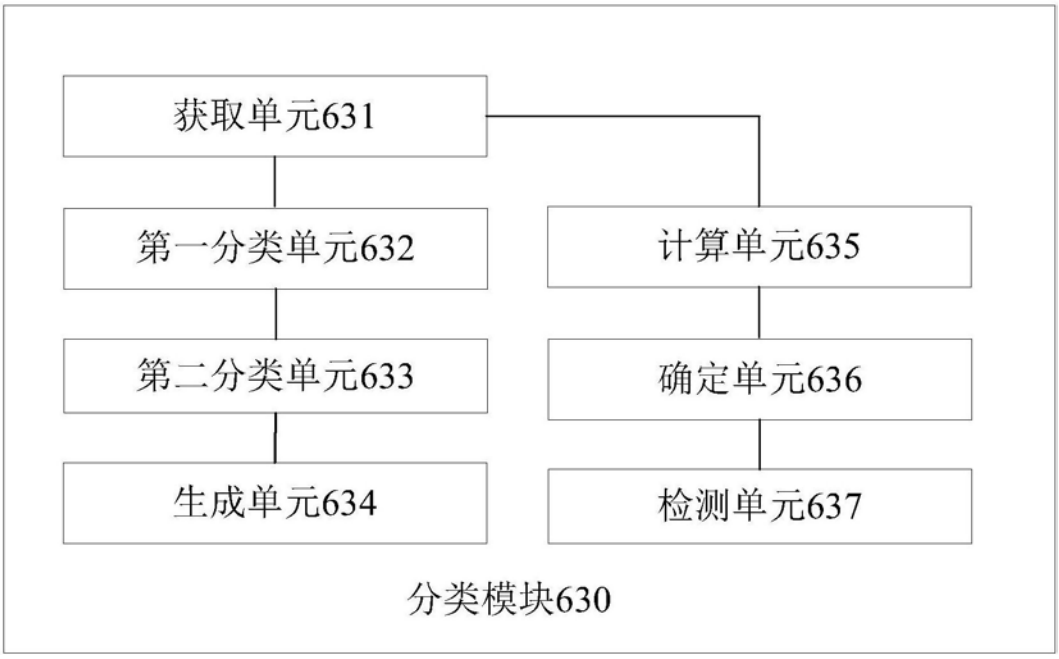

获取单元631
第一分类单元632
第二分类单元633
生成单元634
计算单元635
确定单元636
检测单元637
分类模块630

图8

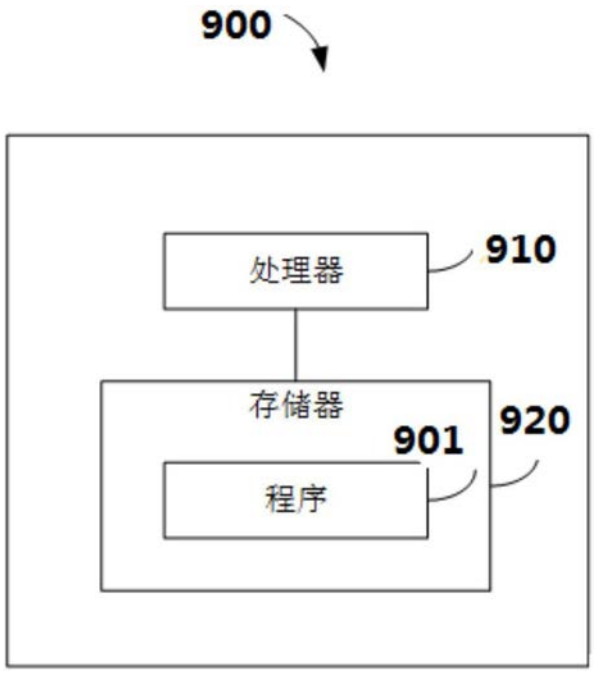

900
处理器
910
存储器
901
920
程序

图9