



(12)发明专利申请

(10)申请公布号 CN 107862375 A

(43)申请公布日 2018.03.30

(21)申请号 201711033505.8

(22)申请日 2017.10.30

(71)申请人 北京计算机技术及应用研究所
地址 100854 北京市海淀区永定路51号

(72)发明人 焦亚森 王金龙 方志 郑隰

(74)专利代理机构 中国兵器工业集团公司专利
中心 11011

代理人 王雪芬

(51)Int.Cl.

G06N 3/04(2006.01)

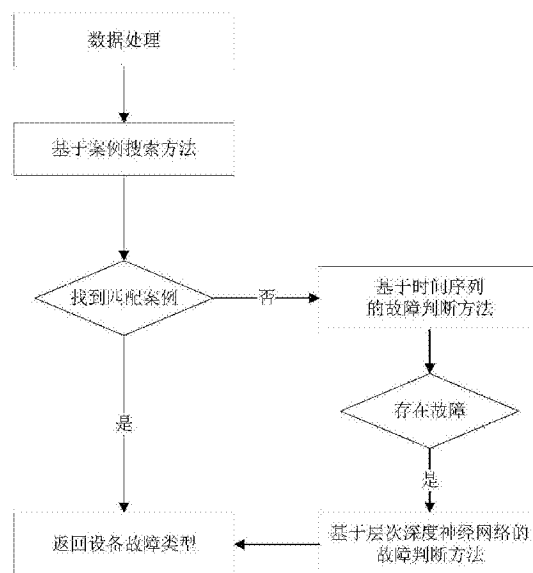
权利要求书2页 说明书5页 附图1页

(54)发明名称

一种两阶段的设备故障诊断方法

(57)摘要

本发明涉及一种两阶段的设备故障诊断方法,涉及设备故障诊断技术领域。本发明可以根据不同的响应时间、计算能力、故障诊断类型细节的要求,调整预测模型的步骤,可以减少对维修人员人为因素的依赖,提高设备故障诊断的效率,为维修人员对设备故障的判断提供重要的参考,在适应不同条件下的设备故障类型诊断中能发挥重要作用。



1. 一种两阶段的设备故障诊断方法,其特征在于,包括以下步骤:

步骤1、数据处理步骤:根据传感器采集到的数据,对数据进行清洗以及统一化处理;

步骤2、基于案例搜索的故障诊断步骤:对于采集到的数据在事先建立好的案例库中通过K近邻方法进行搜索,基于预设的阈值,对于搜索到的案例,返回对应的故障类型信息后结束,对于没有搜索到的案例继续执行步骤3;

步骤3、基于时间特征的故障判断步骤:根据广义自回归条件异方差模型对获得的设备状态特征数据进行计算,根据此值对设备的故障与否进行判断;若判断为存在故障,则继续执行步骤4,否则返回不存在故障的信息后结束;

步骤4、基于层次的深度神经网络故障预测步骤:根据诊断对象的特点,对系统进行分解,建立合适的层次分类模型,利用层次分类模型计算获得对应的故障类型。

2. 如权利要求1所述的方法,其特征在于,步骤1具体包括:

步骤11:采用基于均值的数据补全方法处理缺失数据:存储基于当前时间的前后5条数据,将均值作为数据的填充依据;

步骤12:对于冗余数据,采用序列相似度方法来进行冗余数据的消除:序列相似度采用特征向量之间的欧式距离之和作为衡量标准,并设置阈值,大于阈值的数据则判定为冗余数据被丢弃,序列相似度计算公式如下:

$$\text{Sim}(a_f) = \sum_{e=1}^{e=10} \text{Sim}(a_f, a_e) \quad (1)$$

其中, a_f 表示收到的第f个特征向量, a_e 表示 a_f 前收到的10个特征向量中的第e个, $\text{Sim}(a_f, a_e)$ 表示 a_f 与 a_e 之间的欧式距离;

步骤13:采用0-1值法对数据进行离散化:对于采集到的全部数据的每一维计算平均值作为阈值,对于大于等于阈值的数据置为1,小于阈值的数据置为0。

3. 如权利要求2所述的方法,其特征在于,步骤2具体包括:

步骤21:推理步骤:输入当前设备的基本信息,在案例库中检索相似案例,若有相似案例则给出故障诊断结论与解决措施,若无相似案例则转入基于时间序列的深度神经网络模型中,案例采取基于K-近邻匹配的检索方式进行检索,案例采取基于K-近邻匹配的检索方式进行检索具体步骤为:

每个案例包含m种特征,故障案例 C_i ($i=1, 2, \dots, n$)可用一个m维向量表示: $A_i = (a_{i1}, a_{i2}, \dots, a_{im})$, a_{ij} ($j=1, 2, \dots, m$)是故障案例 C_i 的第j个特征的取值;

案例间的相似度定义为:

$$\text{Sim}(C_i, C_j) = \sum_{k=1}^m (\omega_k \times \text{Sim}(a_{ik}, a_{jk})) / \sum_{k=1}^m \omega_k \quad (2)$$

其中, $0 \leq \text{Sim}(C_i, C_j) \leq 1$; ω_k 表示案例特征向量中第k个特征的权值,且 $\sum_{k=1}^m \omega_k = 1$ 。

4. 如权利要求3所述的方法,其特征在于,步骤3具体包括:

步骤31:针对获取的设备状态信息序列的时间序列化的特点,采用广义自回归条件异方差模型GARCH对设备按时间序列反馈的数据特征向量进行建模,根据得到的计算结果判断是否故障;

采用广义自回归条件异方差模型GARCH对设备按时间序列反馈的数据特征向量进行建模具体为:

时间序列 X_t :

$$X_t = E\{X_t | \Psi_{t-1}\} + \varepsilon_t \quad (3)$$

其中 Ψ_{t-1} 代表在 $t-1$ 时间获得的所有时间序列 X_1, \dots, X_{t-1} , ε_t 代表残差,对残差建立描述方程:

$$\varepsilon_t = \sqrt{h_t} Z_t \quad (4)$$

$$h_t = \alpha_0 + \sum_{i=1}^p \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^q \beta_j h_{t-j} \quad (5)$$

式中: Z_t 是均值为零,方差为1的随机变量; p, q 分别为模型的阶数; α_i 和 β_j 为模型的待估参数,为使条件方差 $h_t > 0$ 要求 α_i 和 β_j 都大于0,同时为了使模型是宽平稳的, α_i 和 β_j 还需满足条件:

$$\sum_i \alpha_i + \sum_j \beta_j < 1 \quad (6)$$

在GARCH模型中,采用最大似然原理对条件方差的参数 $\alpha_0, \alpha_1, \beta_1$ 进行估计,如果 $\{X_1, X_2, \dots, X_T\}$ 是由GARCH模型产生的信号,那么似然函数由如下公式定义:

$$L(\alpha_0, \alpha_1, \beta_1) = f_{X_2, \dots, X_T | X_1, h_1}(X_2, \dots, X_T | X_1, h_1) \quad (7)$$

$$= \prod_{j=2}^T \frac{1}{\sqrt{2\pi h_j}} \exp\left[-\frac{X_j^2}{2h_j}\right] \quad (8)$$

这里 h_t 由递归方法获得,对上式取对数得到对数似然函数为:

$$l(\alpha_0, \alpha_1, \beta_1 | X, h) = -\frac{1}{2} \sum_{j=2}^T \left(\log h_j + \frac{X_j^2}{h_j} \right) \quad (9)$$

其中: $X = (X_1, \dots, X_j)^T$, $h = (h_1, \dots, h_j)^T$,限制条件为公式(6),模型的参数 $\alpha_0, \alpha_1, \beta_1$ 通过最大化公式(9)求出;

根据模型得出的结果,对设备是否故障做出判断;若诊断出无故障则直接返回无故障信息,若有故障则转到步骤4。

5.如权利要求4所述的方法,其特征在于,步骤4中,采用神经网络构造对应于故障分类层次的分类模型,每个人工神经网络均采用三层BP网络,利用层次结构神经网络计算得到的结果匹配故障的具体类型。

一种两阶段的设备故障诊断方法

技术领域

[0001] 本发明涉及设备故障诊断技术领域,具体涉及一种两阶段的设备故障诊断方法。

背景技术

[0002] 在设备维修保障的整个流程中,设备的故障现象(或故障征兆)和故障原因(或故障单元)之间的关系复杂,且具有随机性和不确定性等特点。

[0003] 传统的故障诊断方法因其缺乏不精确条件下的推理能力和自主学习能力,暴露出故障诊断效率低、成本高等弊病。

[0004] 早期的故障诊断方法过分依赖于领域专家或经验丰富的维修人员的判断,造成诊断故障的过程具有极强的主观性且成本过高。

[0005] 一般的预测方法,将所有特征序列的获取时间先后顺序因素忽略,导致失去了时间信息,使得预测的准确性以及合理性大幅度的下降。

[0006] 大多数的故障诊断方法将故障类型与无故障作为同一个层次的数据进行处理,然而在现实情况下,正常情况要远远多于故障情况。将不平衡的数据一起处理将会导致过拟合的现象出现,使得故障预测效果不具有可信性。

发明内容

[0007] (一)要解决的技术问题

[0008] 本发明要解决的技术问题是:如何设计针对设备维修保障的故障诊断方法,提升预测的准确度和可解释性。

[0009] (二)技术方案

[0010] 为了解决上述技术问题,本发明提供了一种两阶段的设备故障诊断方法,包括以下步骤:

[0011] 步骤1、数据处理步骤:根据传感器采集到的数据,对数据进行清洗以及统一化处理;

[0012] 步骤2、基于案例搜索的故障诊断步骤:对于采集到的数据在事先建立好的案例库中通过K近邻方法进行搜索,基于预设的阈值,对于搜索到的案例,返回对应的故障类型信息后结束,对于没有搜索到的案例继续执行步骤3;

[0013] 步骤3、基于时间特征的故障判断步骤:根据广义自回归条件异方差模型对获得的设备状态特征数据进行计算,根据此值对设备的故障与否进行判断;若判断为存在故障,则继续执行步骤4,否则返回不存在故障的信息后结束;

[0014] 步骤4、基于层次的深度神经网络故障预测步骤:根据诊断对象的特点,对系统进行分解,建立合适的层次分类模型,利用层次分类模型计算获得对应的故障类型。

[0015] 优选地,步骤1具体包括:

[0016] 步骤11:采用基于均值的数据补全方法处理缺失数据:存储基于当前时间的前后5条数据,将均值作为数据的填充依据;

[0017] 步骤12:对于冗余数据,采用序列相似度方法来进行冗余数据的消除:序列相似度采用特征向量之间的欧式距离之和作为衡量标准,并设置阈值,大于阈值的数据则判定为冗余数据被丢弃,序列相似度计算公式如下:

$$[0018] \quad \text{Sim}(a_f) = \sum_{e=1}^{e=10} \text{Sim}(a_f, a_e) \quad (1)$$

[0019] 其中, a_f 表示收到的第 f 个特征向量, a_e 表示 a_f 前收到的10个特征向量中的第 e 个, $\text{Sim}(a_f, a_e)$ 表示 a_f 与 a_e 之间的欧式距离;

[0020] 步骤13:采用0-1值法对数据进行离散化:对于采集到的全部数据的每一维计算平均值作为阈值,对于大于等于阈值的数据置为1,小于阈值的数据置为0。

[0021] 优选地,步骤2具体包括:

[0022] 步骤21:推理步骤:输入当前设备的基本信息,在案例库中检索相似案例,若有相似案例则给出故障诊断结论与解决措施,若无相似案例则转入基于时间序列的深度神经网络模型中,案例采取基于 K -近邻匹配的检索方式进行检索,案例采取基于 K -近邻匹配的检索方式进行检索具体步骤为:

[0023] 每个案例包含 m 种特征,故障案例 C_i ($i=1, 2, \dots, n$)可用一个 m 维向量表示: $A_i = (a_{i1}, a_{i2}, \dots, a_{im})$, a_{ij} ($j=1, 2, \dots, m$)是故障案例 C_i 的第 j 个特征的取值;

[0024] 案例间的相似度定义为:

$$[0025] \quad \text{Sim}(C_i, C_j) = \frac{\sum_{k=1}^m (\omega_k \times \text{Sim}(a_{ik}, a_{jk}))}{\sum_{k=1}^m \omega_k} \quad (2)$$

[0026] 其中, $0 \leq \text{Sim}(C_i, C_j) \leq 1$; ω_k 表示案例特征向量中第 k 个特征的权值,且 $\sum_{k=1}^m \omega_k = 1$ 。

[0027] 优选地,步骤3具体包括:

[0028] 步骤31:针对获取的设备状态信息序列的时间序列化的特点,采用广义自回归条件异方差模型GARCH对设备按时间序列反馈的数据特征向量进行建模,根据得到的计算结果判断是否故障;

[0029] 采用广义自回归条件异方差模型GARCH对设备按时间序列反馈的数据特征向量进行建模具体为:

[0030] 时间序列 X_t :

$$[0031] \quad X_t = E\{X_t | \Psi_{t-1}\} + \varepsilon_t \quad (3)$$

[0032] 其中 Ψ_{t-1} 代表在 $t-1$ 时间获得的所有时间序列 X_1, \dots, X_{t-1} , ε_t 代表残差,对残差建立描述方程:

$$[0033] \quad \varepsilon_t = \sqrt{h_t} Z_t \quad (4)$$

$$[0034] \quad h_t = \alpha_0 + \sum_{i=1}^p \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^q \beta_j h_{t-j} \quad (5)$$

[0035] 式中: Z_t 是均值为零,方差为1的随机变量; p, q 分别为模型的阶数; α_i 和 β_j 为模型的待估参数,为使条件方差 $h_t > 0$ 要求 α_i 和 β_j 都大于0,同时为了使模型是宽平稳的, α_i 和 β_j 还需满足条件:

$$[0036] \quad \sum_i \alpha_i + \sum_j \beta_j < 1 \quad (6)$$

[0037] 在GARCH模型中,采用最大似然原理对条件方差的参数 $\alpha_0, \alpha_1, \beta_1$ 进行估计,如果

$\{X_1, X_2, \dots, X_T\}$ 是由GARCH模型产生的信号,那么似然函数由如下公式定义:

$$L(\alpha_0, \alpha_1, \beta_1) = f_{X_2, \dots, X_T | X_1, h_1}(X_2, \dots, X_T | X_1, h_1) \quad (7)$$

$$= \prod_{j=2}^T \frac{1}{\sqrt{2\pi h_j}} \exp\left[-\frac{X_j^2}{2h_j}\right] \quad (8)$$

这里 h_t 由递归方法获得,对上式取对数得到对数似然函数为:

$$l(\alpha_0, \alpha_1, \beta_1 | X, h) = -\frac{1}{2} \sum_{j=2}^T \left(\log h_j + \frac{X_j^2}{h_j} \right) \quad (9)$$

其中: $X = (X_1, \dots, X_j)^T$, $h = (h_1, \dots, h_j)^T$, 限制条件为公式(6), 模型的参数 $\alpha_0, \alpha_1, \beta_1$ 通过最大化公式(9)求出;

根据模型得出的结果,对设备是否故障做出判断;若诊断出无故障则直接返回无故障信息,若有故障则转到步骤4。

优选地,步骤4中,采用神经网络构造对应于故障分类层次的分层模型,每个人工神经网络均采用三层BP网络,利用层次结构神经网络计算得到的结果匹配故障的具体类型。

(三) 有益效果

本发明提出了针对设备维修保障的两阶段智能故障诊断方法。在基于案例搜索的基础上,引入了基于时间序列的故障判断方法,并结合了多层次神经网络的故障类型分析。本发明充分利用了设备状态信息的时间特征以及将判断设备是否故障和预测故障的具体类型分开进行,使得数据可以有效地避免了故障数据与非故障数据不平衡的情况,提升了预测的准确度和可解释性。基于时间序列的预测方法此前多用于金融领域,本发明首次将其应用于设备故障诊断领域。此外,本发明的多步骤自适应的预测方法也可以根据程序运行环境有效地做出适配,良好地适用于多种硬件环境。

附图说明

图1是本发明的方法流程图;

图2是本发明的方法中建立层次故障诊断模型示意图。

具体实施方式

为使本发明的目的、内容和优点更加清楚,下面结合附图和实施例,对本发明的具体实施方式作进一步详细描述。

如图1所示,本发明的两阶段的设备故障诊断方法包括如下步骤:

步骤1、数据处理步骤:根据传感器采集到数据,对数据进行清洗以及统一化处理;

步骤2、基于案例搜索的故障诊断步骤:对于采集到的数据在事先建立好的故障案例库中通过K近邻方法进行搜索。基于预设的阈值,对于搜索到的案例,返回对应的故障类型信息后结束,对于没有搜索到的案例继续执行步骤3;

步骤3、基于时间特征的故障判断步骤:根据广义自回归条件异方差模型对获得的设备状态特征数据进行计算,根据此值对设备的故障与否进行判断;若判断为存在故障,则

继续执行步骤4,否则返回不存在故障的信息后结束;

[0053] 步骤4、基于层次的神经网络故障预测步骤:根据诊断对象的特点,对系统进行分解,建立合适的综合层次分类模型,利用该模型计算获得对应的故障类型。

[0054] 所述数据获取步骤1中,包括:

[0055] 步骤11:采用基于均值的数据补全方法处理缺失数据:存储基于当前时间的前后5条数据,将均值作为数据的填充依据。本步骤在节省了空间以及计算时间的基础上,对数据的缺失进行了弥补。

[0056] 步骤12:对于冗余数据,实施序列相似度方法来进行冗余数据的消除:序列相似度采用特征向量之间的欧式距离之和作为衡量标准,并根据经验设置阈值,大于阈值的数据则判定为冗余数据被丢弃。序列相似度计算公式如下:

$$[0057] \quad \text{Sim}(a_f) = \sum_{e=1}^{e=10} \text{Sim}(a_f, a_e) \quad (1)$$

[0058] 其中, a_f 表示收到的第 f 个特征向量, a_e 表示 a_f 前收到的10个特征向量中的第 e 个。 $\text{Sim}(a_f, a_e)$ 表示 a_f 与 a_e 之间的欧式距离。

[0059] 步骤13:采用0-1值法对数据进行离散化:对于采集到的全部数据的每一维计算平均值作为阈值,对于大于等于阈值的数据置为1,小于阈值的数据置为0。

[0060] 所述基于案例搜索的故障诊断步骤2中,包括:

[0061] 步骤21:推理步骤。输入当前设备的基本信息,在案例库中检索相似案例,若有相似案例则给出故障诊断结论与解决措施,若无相似案例则转入基于时间序列的深度神经网络模型中。案例采取基于KNN(K-近邻)匹配的检索方式进行检索。KNN方法介绍如下:

[0062] 每个案例包含 m 种特征,故障案例 $C_i = (i = 1, 2, \dots, n)$ 可用一个 m 维向量表示: $A_i = (a_{i1}, a_{i2}, \dots, a_{im})$, a_{ij} ($j = 1, 2, \dots, m$) 是故障案例 C_i 的第 j 个特征的取值。

[0063] 案例间的相似度定义为:

$$[0064] \quad \text{Sim}(C_i, C_j) = \sum_{k=1}^m (\omega_k \times \text{Sim}(a_{ik}, a_{jk})) / \sum_{k=1}^m \omega_k \quad (2)$$

[0065] 其中, $0 \leq \text{Sim}(C_i, C_j) \leq 1$; ω_k 表示案例特征向量中第 k 个特征的权值,且 $\sum_{k=1}^m \omega_k = 1$ 。

[0066] 所述基于时间特征的故障判断步骤3中,包括:

[0067] 步骤31:针对获取的设备状态信息序列的时间序列化的特点,实施广义自回归条件异方差模型 (generalized autoregressive conditional heteroscedasticity model, 简记为GARCH模型)。本发明中对设备按时间序列反馈的数据特征向量进行GARCH建模,根据得到的计算结果判断是否故障。

[0068] 广义自回归条件异方差模型 (GARCH) 其简要描述如下。

[0069] 时间序列 X_t :

$$[0070] \quad X_t = E\{X_t | \Psi_{t-1}\} + \varepsilon_t \quad (3)$$

[0071] 其中 Ψ_{t-1} 代表在 $t-1$ 时间获得的所有时间序列 X_1, \dots, X_{t-1} , ε_t 代表残差。对残差建立描述方程:

$$[0072] \quad \varepsilon_t = \sqrt{h_t} Z_t \quad (4)$$

$$[0073] \quad h_t = \alpha_0 + \sum_{i=1}^p \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^q \beta_j h_{t-j} \quad (5)$$

[0074] 式中： Z_t 是均值为零，方差为1的随机变量； p, q 分别为模型的阶数； α_i 和 β_j 为模型的待估参数。为使条件方差 $h_t > 0$ 要求 α_i 和 β_j 都大于0。同时为了使模型是宽平稳的， α_i 和 β_j 还需满足条件：

$$[0075] \quad \sum_i \alpha_i + \sum_j \beta_j < 1 \quad (6)$$

[0076] 在本发明的GARCH模型中，采用最大似然原理对条件方差的参数 $\alpha_0, \alpha_1, \beta_1$ 进行估计，如果 $\{X_1, X_2, \dots, X_T\}$ 是由GARCH模型产生的信号，那么似然函数由如下公式定义：

$$L(\alpha_0, \alpha_1, \beta_1) = f_{X_2, \dots, X_T | X_1, h_1}(X_2, \dots, X_T | X_1, h_1) \quad (7)$$

$$[0077] \quad = \prod_{j=2}^T \frac{1}{\sqrt{2\pi h_j}} \exp\left[-\frac{X_j^2}{2h_j}\right] \quad (8)$$

[0078] 这里 h_t 由递归方法获得，对上式取对数得到对数似然函数为：

$$[0079] \quad l(\alpha_0, \alpha_1, \beta_1 | X, h) = -\frac{1}{2} \sum_{j=2}^T \left(\log h_j + \frac{X_j^2}{h_j} \right) \quad (9)$$

[0080] 其中： $X = (X_1, \dots, X_j)^T$ ， $h = (h_1, \dots, h_j)^T$ ，限制条件为公式(6)，模型的参数 $\alpha_0, \alpha_1, \beta_1$ 通过最大化公式(9)求出。

[0081] 根据模型得出的结果，对设备是否故障做出判断。若诊断出无故障则直接返回无故障信息，若有故障则转到步骤4。

[0082] 所述基于层次的深度神经网络故障预测步骤4中，包括：

[0083] 步骤41：层次结构神经网络预测故障具体类型。设备的故障根据主从层次建立树型层次模型，层次模型示意图如图2所示。实施神经网络可以构造对应于故障分类层次的分类模型。每个人工神经网络均采用三层BP网络。利用层次结构神经网络计算得到的结果匹配故障的具体类型。

[0084] 以上所述仅是本发明的优选实施方式，应当指出，对于本技术领域的普通技术人员来说，在不脱离本发明技术原理的前提下，还可以做出若干改进和变形，这些改进和变形也应视为本发明的保护范围。

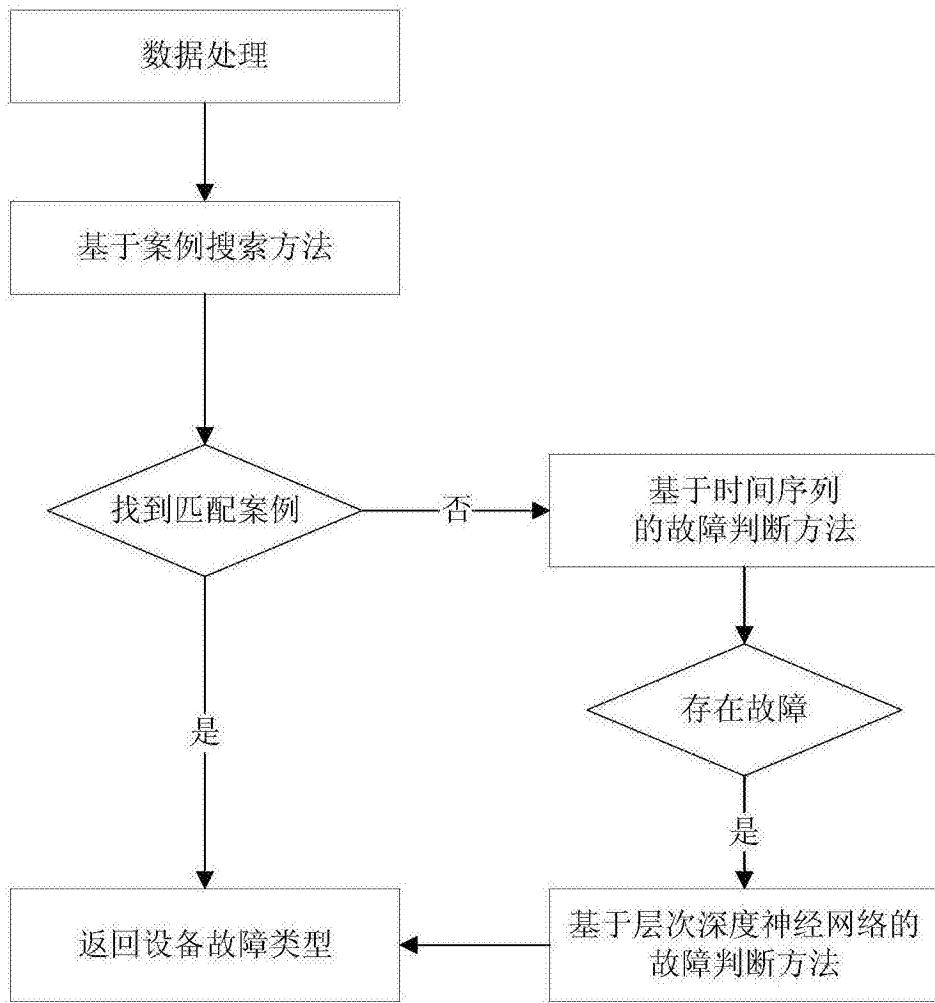


图1

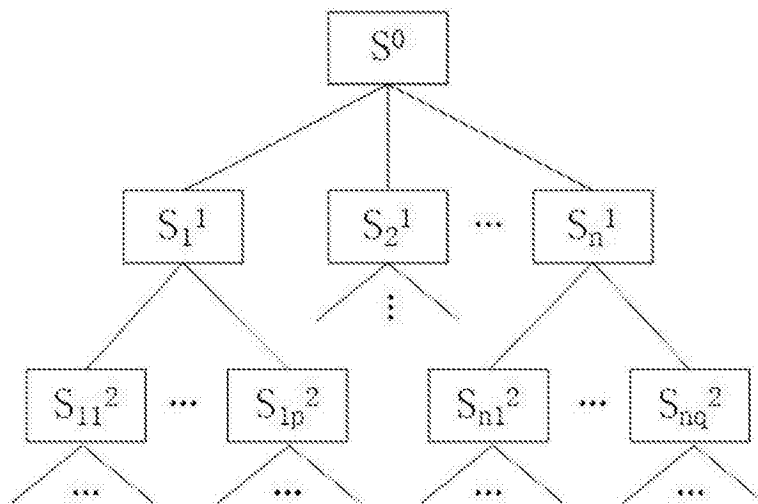


图2