

基于云服务的统计测试非监督故障诊断预测

林楠¹, 魏涛²

(1. 郑州大学 软件技术学院, 郑州 450002; 2. 河南工程学院 计算机学院, 郑州 450002)

摘要: 为了对系统中的潜在故障进行有效地预测, 提出一种基于统计测试的非监督故障预测方法; 首先, 将云服务系统定义为运行在相同的软/硬件环境下, 具有相同输入数据的并行系统; 在数据预处理过程中, 对性能计数器中的数据进行标准化, 并选取了一定分位数下的计数器数据信息; 最后根据具有相同软/硬件环境和输入数据的节点将产生相同的输出这一原则提出了一种统计测试方法用于系统故障的预测; 实验表明, 文章提出的基于统计测试的故障预测方法与其它相关算法相比, 具有预测准确性高和执行效率快等优点。

关键词: 云服务; 故障诊断; 故障预测; 统计测试

Statistical Test Based Unsupervised Fault Predicting Approach

Lin Nan¹, Wei Tao²

(1. College of Software Technology, Zhengzhou University, Zhengzhou 450002, China;

2. School of Computer Henan Institute of Engineering, Zhengzhou 450002, China)

Abstract: Predicting faults of a Cloud services system before it fails can win time for system operators and other recovery mechanisms, and thus improve the quality of services. In order to predicting the latent faults efficiently for such systems, this paper proposed a statistical test based unsupervised fault predicting approach. First, we defined the Cloud service system as a parallel system running in the same software and hardware environment, and with the same input data. During the process of data preprocessing, we normalized the data in performance counters, and chose a subset under some percentile. Finally, according to the principle of nodes with the same software/hardware environment and input data had the same output, we proposed a statistical test approach for predicting the fault. The experiments show that, the proposed fault predicting approach based on statistical test has better accuracy and quicker execution time compared with other related researches.

Keywords: cloud services; fault diagnosis; fault prediction; statistical test

0 引言

随着云计算应用的普及, 互联网云服务系统往往都由成千上万个甚至更多的节点组成。在如此大的规模下, 云服务系统发生软/硬件故障的概率显著提高^[1]。通常情况下, 云服务系统采用冗余或者故障恢复机制保障系统在发生故障时能正常运行^[2]。然而, 当系统发生软/硬件故障时, 如果不能对故障采取有效的应对措施, 那么该故障会逐渐扩大, 最终导致冗余和故障恢复机制崩溃, 从而使整个系统失效^[3]。因而, 在软/硬件故障累积到系统失效前对故障进行预测, 可以提高云服务的可靠性。对系统的故障行为进行监测的最常用的方法是收集并分析系统的性能计数器^[4]。在云服务系统中, 每个节点定义了大量的性能计数器, 这些性能计数器分属于不同的系统层次, 从系统的最底层—物理层, 如 CPU 使用率, 到系统的最上层—服务层, 如数据库的查询数目。由于云服务系统规模庞大, 人工方式对整个数据中心进行性能监测显然是不可行的。因此, 数据中心在提供云服务时需要自动化的性能计数器监测方法^[5]。

当系统的性能计数器偏离预定义的阈值时, 认定系统发生

故障^[6]。目前, 数据中心所采用的故障检测方法大都是基于规则的^[7-10]。通过看门狗 (Watchdog) 来监测系统的性能计数器, 例如 CPU 的运行温度、内存的使用率、以及磁盘的使用空间等。当这些性能计数器的参数超过预定义的阈值时, 便触发相应的活动, 如向系统操作员进行报警或者自动进行恢复措施。基于规则的故障检测方法存在以下几个方面的缺点。首先, 阈值必须足够的低, 从而使得所有的故障都能被监测到。然而当阈值设的太低时, 可能产生大量的错误检测。此外, 随着系统工作负载的不断变化, 看门狗中的阈值也应该是动态变化的。其次, 不同的云服务或者是同一云服务的不同版本都有可能存在不同的阈值, 这使得系统操作人员需要经常对规则进行人工调整。事实上, 上述规则往往在系统发生故障后对故障数据进行一系列的分析得到的。

针对基于规则的故障预测方法存在的不足, 文献 [11-12] 提出通过历史标记数据对检测器进行学习的方法。这种方法虽然能解决基于规则的故障预测方法存在的不足, 但是在实际中却很难获得系统运行的带标记的历史数据。同时, 该方法只能应用于已有的服务, 当服务发生变化时, 历史标记数据变得不可用。此外, Patan 等人 [13] 提出一种基于神经网络的故障诊断方法, 采用注入代码的方式周期性地对服务进行监测。然而, 这种方式是侵入式的, 在多数情况下变得不可用。本文将云服务系统定义为运行在相同的软/硬件环境下, 具有相同输入数据的并行系统, 并根据具有相同软/硬件环境和输入数据的节点将产生相同的输出这一原则提出了一种统计测试方法用于系统故障的预测。

收稿日期: 2014-09-03; 修回日期: 2014-10-06。

基金项目: 国家科技型中小企业技术创新基金项目 (10C26214102198)。

作者简介: 林楠 (1973-), 女, 河南郑州人, 硕士研究生, 副教授, 主要从事下一代互联网技术与嵌入式系统, 故障检测等方向的研究。

1 故障预测方法

1.1 云服务系统

云服务的可靠性往往通过多副本的方式来实现。服务或者数据被复制到多个节点上,并通过负载均衡技术进行工作负载的划分。例如 Hadoop 平台通常保存数据的 3 个副本,并保存在不同的节点上供恢复机制使用。本文假设系统的所有节点使用相同的软硬件对相同的数据进行处理,这些节点具有相同的角色,它们的运行行为可通过性能计数器获得。在进行性能监测时,计数器数据分别来自于系统的硬件、操作系统和应用服务。当某个节点的性能计数器长时间不同于其它节点时,认定该节点为故障节点。

令云服务系统的节点集合为 M , 包含 $|M|$ 个节点, 每个节点的性能计数器集合为 C 且包含 $|C|$ 个性能计数器。在时刻 t , 节点 m 的计数器向量为 $x(m, t)$ 。对于标量 s , $|s|$ 为 s 的绝对值。向量 x 的 L_2 范式为 $\|x\|$, $x \cdot x'$ 为 x 和 x' 的内积。令 T 为性能计数器的采样时间点集合, t 和 t' 表示具体的采样时间点。

本文假设节点 m 在正常运行情况下其性能计数器的随机变量为 $X(t)$, 并将它的采样值记为 $x(m, t)$ ($t = 1, 2, \dots$)。系统在正常运行过程中, 服务的软硬件和输入数据是相同的, 因而节点的性能计数器应该是相同的。当某个节点的性能计数器向量取值与其它节点的性能计数器向量取值不同时, 认定该节点发生故障。

1.2 数据预处理

在根据节点的性能计数器数据对故障进行预测前, 需要对原始的计数器数据进行预处理。给定原始计数器数据后, 数据预处理过程主要清洗掉不正确的数据, 并对数据进行规范化处理, 数据预处理算法的细节见算法 1。计数器向量预处理伪代码如下:

```

初始化  $\theta^1, \theta^2$  和  $\theta^3$ ;
令  $n_c(m) \leftarrow$  节点  $m$  关于计数器  $c$  的个数;
令  $z_c(m, t) \leftarrow t - 1$  时刻节点  $m$  关于计数器  $c$  的个数;
For each  $c \in C$  do
     $v_c = \text{mean}_{m \in M, t \in T}(z_c(m, t))$ ;
     $\sigma_c = \text{STD}_{m \in M, t \in T}(z_c(m, t))$ ;
For each  $m \in M \wedge t \in T$  do
     $y_c(m, t) = \frac{z_c(m, t) - v_c}{\sigma_c}$ ;
For each  $m \in M$  do
     $\mu_c(m) = \text{median}_{t \in T}(y_c(m, t))$ ;
     $\text{mad}_c(m) = \text{median}_{t \in T}(|y_c(m, t) - \mu_c(m)|)$ ;
     $n_c = \text{median}_{m \in M}(n_c(m))$ ;
     $\mu_c = \text{median}_{m \in M}(\mu_c(m))$ ;
     $\varphi_c^1 = p_{90} \left( \left| \frac{n_c(m) - n_c}{n_c} \right| \right)$ ;
     $\varphi_c^2 = p_{90} \left( \left| \frac{\mu_c(m) - \mu_c}{\text{mad}_c(m)} \right| \right)$ ;
If  $\varphi_c^1 \leq \theta^1 \wedge \varphi_c^2 \leq \theta^2 \wedge n_c \geq \theta^3$ 
For each  $m \in M \wedge t \in T$ 
     $x_c(m, t) = y_c(m, t)$ ;
    
```

当节点的性能计数器为事件驱动的计数器时, 这些计数器的值是非周期的。这些数据从统计学的角度来看可能会产生偏

差, 因此需要进行预处理。对于计数器 c , 测量并记录每个节点 m 记录该数据的次数 $n_c(m)$ 。在 t 时刻之前, 令节点 m 关于计数器 c 的取值为 $z_c(m, t)$ 。对于每个计数器 c , 根据所有的 $z_c(m, t)$ 值得到均值和标准差 (公式 (1)), 并对每个 $z_c(m, t)$ 值进行标准化 (公式 (2))。于是, $y_c(m, t)$ 为一个均值为 0, 标准差为 1 的概率分布。

$$\begin{cases} v_c = \text{mean}_{m \in M, t \in T}(z_c(m, t)) = \frac{\sum_{m \in M, t \in T} (z_c(m, t) - \bar{z})}{|M| \cdot |T|} \\ \sigma_c = \text{STD}_{m \in M, t \in T}(z_c(m, t)) = \sqrt{\frac{\sum_{m \in M, t \in T} (z_c(m, t) - \bar{z})^2}{|M| \cdot |T|}} \end{cases} \quad (1)$$

$$\begin{aligned} \text{其中: } \bar{z} &= \frac{\sum_{m \in M, t \in T} z_c(m, t)}{|M| \cdot |T|} \\ y_c(m, t) &= \frac{z_c(m, t) - v_c}{\sigma_c} \end{aligned} \quad (2)$$

在周期记录的计数器中, 所有的节点的记录个数是相似的。为了使算法适应不同节点的记录个数的不同, 采用分位数的概念消除节点中过多的计数器。对于每个节点 m , 得到中位数 $\mu_c(m) = \text{median}_{t \in T}(y_c(m, t))$, 以及中位数绝对偏差 $\text{mad}_c(m)$ (公式 (3))。

$$\text{mad}_c(m) = \text{median}_{t \in T}(|y_c(m, t) - \mu_c(m)|) \quad (3)$$

然后, 根据所有节点的记录次数 $n_c(m)$ 以及 $\mu_c(m)$, 得到中位数 $n_c = \text{median}_{m \in M}(n_c(m))$ 和 $\mu_c = \text{median}_{m \in M}(\mu_c(m))$, 并据此计算 90 分位数 φ_c^1 和 φ_c^2 (公式 (4))。

$$\begin{cases} \varphi_c^1 = p_{90} \left(\left| \frac{n_c(m) - n_c}{n_c} \right| \right) \\ \varphi_c^2 = p_{90} \left(\left| \frac{\mu_c(m) - \mu_c}{\text{mad}_c(m)} \right| \right) \end{cases} \quad (4)$$

最后, 根据预定义的规则将每个节点的 $y_c(m, t)$ 赋值为 t 时刻的计数器向量 $x_c(m, t)$ 。在预处理结束后, 每个节点只留下小部分有用的计数器。

1.3 统计测试方法

在对节点的性能计数器进行预处理后, 采用统计方法检测计数器中的异常信息, 从而进行系统故障的检测。

令 m 和 m' 为两个运行相同服务的节点, $x(m, t)$ 和 $x(m', t)$ 为相应节点在时刻 t 的性能计数器向量。由于节点 m 和 m' 运行相同的服务, 那么 $x(m, t)$ 和 $x(m', t)$ 是相似的, 即 $x(m, t) - x(m', t) \approx \vec{0}$ 。当 m 和 m' 运行在正常状态下, $\frac{x(m, t) - x(m', t)}{\|x(m, t) - x(m', t)\|} \approx \vec{0}$, 并且在有限的时间步 $t_1 \leq t \leq t_2$ 内, $\sum_{t_1 \leq t \leq t_2} \frac{x(m, t) - x(m', t)}{\|x(m, t) - x(m', t)\|} \approx \vec{0}$ 。

将运行相同服务的节点扩展到 M , $x(m, t)$ 的含义同上。对于每一个 $m \in M$, 定义如下所示的向量:

$$v(m) = \frac{1}{T(M-1)} \sum_{t \in T} \sum_{m' \in M} \frac{x(m, t) - x(m', t)}{\|x(m, t) - x(m', t)\|} \quad (5)$$

对于 $\forall m \in M$ 和 $\forall t \in T$, $x(m, t)$ 是关于 $X(t)$ 的相互独立的采样, 那么对于任意的 $\gamma > 0$, 都有如下不等式成立 [14]:

$$\Pr(\exists m \in M, \|v(m)\| \geq \hat{E}[\|v(m)\|] + \gamma) \leq (M+1)e^{-\frac{\gamma M^2}{2\sqrt{M+2}}} \quad (6)$$

其中: $\hat{E}[\|v(m)\|]$ 为 $\|v(m)\|$ 的样本均值。

公式 (6) 说明, 当所有的节点都正常工作时, $\|v(m)\|$ 偏离样本均值 $\hat{E}[\|v(m)\|]$ 的概率很低, 这说明错误检测的概率很小。采用统计方法进行故障测试的算法见算法 2。统计故障测试算法

输入: 显著性水平 α

输出: 那个产生故障的节点列表

For each $m \in M$ do

$$v(m) = \frac{1}{T(M-1)} \sum_{t \in T} \sum_{m' \in M} \frac{x(m,t) - x(m',t)}{\|x(m,t) - x(m',t)\|};$$

$$s(m) = \|v(m)\|;$$

For each $m \in M$ do

$$\gamma = \min(0, \text{mean}(s) - s(m));$$

$$p(m) = (M+1) \exp\left(\frac{-TM\gamma^2}{2(\sqrt{M}+2)^2}\right);$$

If $p(m) < \alpha$

将 m 加入到可以故障节点列表。

2 实验设计与结果分析

为了对本文提出的方法进行测试, 实验建立了一个含有 20 个节点的 Hadoop 集群, 每个节点是一个含有 2.16 GHz 英特尔酷睿双核处理器, 1 GB 内存和 500G 磁盘存储空间的个人计算机, 运行的操作系统为 CentOS v6.0。

实验采用的数据集为 Tencent 数据集。在该实验环境中, 采用迭代方式计算每个节点的 PageRank 值。在 Hadoop 环境下, 每个 Hadoop 线程的软/硬件环境都是相同的, 并且具有相同的输入数据, 因此适用于本文提出的基于统计测试的故障预测方法。在程序运行过程中, 由于系统发生软/硬件故障的概率非常低, 我们通过人工注入故障的方式加速实验的进行。

首先, 将本文提出的方法与基于神经网络的故障诊断方法和采用故障矩阵的机器学习算法进行对比。在准确性对比中, 采用的评价标准为算法的失效比例, 即所有的预测结果中预测结果与实际不符的预测所占的比例, 实验结果如图 1 所示。从该图可以看出, 随着时间的推移, 3 种算法在进行故障预测时, 其失效比例都逐渐提高。在 3 种算法的对比中, 基于故障矩阵学习方法的失效比例最高, 但是其增长率却缓慢增加。这是因为随着系统的计数器数据逐渐增多, 其预测模型的准确性也逐渐提高。此外, 由于本实验采用抽样的方法, 故计数器数据量小影响了故障矩阵学习方法的准确性。在对比中, 基于统计测试方法的失效比例最低, 本文提出的统计测试方法采用抽样方式对系统的计数器进行监测, 当数据量大时, 失效比例的增长率不会随之变化。

其次, 实验对比了 3 种算法的运行时间, 采用的评价指标为进行单个故障预测所用的 CPU 运行时间, 实验结果如图 2 所示。在 3 种算法运行时间的对比中, 统计测试进行故障预测所用的时间最短, 神经网络方法次之, 采用故障矩阵进行学习方法所用的时间最长。在统计测试方法中, 算法对系统正常运行的计数器进行采样, 采样分布的数据量大小决定着算法对系统状态的近似程度。本实验采用的采样周期为 5 分钟, 其大大压缩了样本的空间, 因此该算法所用的执行时间最短。当采用系统的故障矩阵进行学习时, 对预测模型的每一次训练不仅包含了当前的计数器读数, 而且包含了全部的历史计数器信息, 因此故障诊断所需的时间是逐渐递增的。在 PageRank 算法结

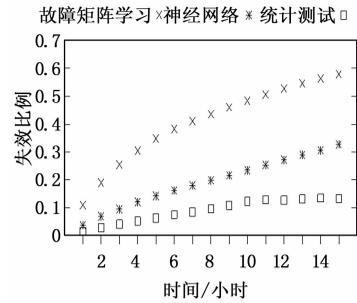


图 1 算法的失效比例对比

束时, 系统中注入了多个故障, 每个故障预测所用的时间是不一样的, 本文应用这些故障预测所用的时间的平均值作为故障矩阵学习方法进行故障预测所用的时间。

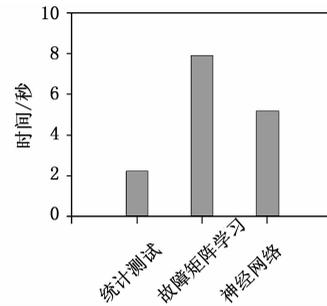


图 2 算法的平均运行时间对比

最后, 实验评估了统计测试方法在进行故障预测时的性能随着时间的变化, 采用的评价指标为 Precision、Recall 和 FPR (false positive rate)。Precision 为预测结果中正确预测所占的比例; Recall 为所有的实际故障中被预测出的故障所占的比例; FPR 为所有的预测结果中预测结果与实际不符的预测所占的比例, 即失效比例。实验结果如图 3 所示。从该图可以看出, 算法的 FPR 随着时间的推移逐渐趋于平稳状态。统计测试方法的 Recall 随着时间的变化在不断波动的情况下缓慢下降。Precision 随着时间的推移波动的很厉害, 这是由于在进行 PageRank 计算时, 每隔一段时间算法在 Map 和 Reduce 间进行切换, 并且 PageRank 在重新启动时需要要对系统进行初始化, 这些因素都影响系统故障诊断的准确性。

3 结束语

云服务平台通常含有成千上万甚至规模更多的节点。在如此大的规模上, 整个系统发生故障的概率显著提高。当系统中的局部节点发生故障后, 如果能迅速地检测到该故障, 并且阻止该故障的蔓延导致整个系统失效, 可以为系统操作人员或者恢复机制赢得处理时间。本文提出一种基于统计测试的非监督故障预测方法。首先, 将云服务系统定义为运行在相同的软/硬件环境下, 具有相同输入数据的并行系统。在数据预处理过程中, 对性能计数器中的数据进行标准化, 并选取了一定分位数下的计数器数据信息。最后, 根据具有相同软/硬件环境和输入数据的节点将产生相同的输出这一原则提出了一种统计测试方法用于系统故障的预测。实验表明, 本文提出的基于统计测试的故障预测方法与其它相关算法相比, 具有预测准确性高

看出圆心误差 $\Phi < 1 \text{ mm}$, 角度误差 $\sigma = \gamma_1 - \gamma_2 < 2^\circ$, 因为星轮直径 d 小于 45 mm , 当角度误差在 2° 之内的情况下, 取 $d = 45 \text{ mm}$, $\sigma = 2^\circ$ 折算成圆周上的误差 $\Delta = \frac{\pi \gamma d}{360 \times 2} \leq 1 \text{ mm}$, 符合装配的要求, 所以该视觉检测系统能够达到检测要求。

6 结束语

本文提出了一种单向器星轮自动化装配方案, 研究了基于计算机视觉的星轮位姿检测, 采用 Open CV 视觉库进行星轮的位姿提取, 实现方便, 效率较高。通过实验验证了改视觉检测系统的可靠性。

参考文献:

- [1] 尹仕斌, 任永杰. 机器人视觉测量系统中的工具中心点快速修复技术 [J]. 机器人, 2013, 6 (2): 736-743.
- [2] 吴俊君, 胡国生. 攀爬机器人末端位姿的视觉检测与控制 [J]. 计算机测量与控制, 2013, 21 (7): 1740-1742.
- [3] 程广华. 自动化装配系统及其应用 [J]. 日用电器, 2012,

3 (3): 49-54.

- [4] 伍太宾, 任广升. 单向器主体的实用制造技术 [J]. 农业机械学报, 2005, 12 (2): 159-161.
- [5] 胡小平, 左富勇, 谢珂. 微装配机器人手眼标定方法研究 [J]. 仪器仪表学报, 2012, 7 (3): 1521-1526.
- [6] 郭斌, 徐杜, 蒋永平, 等. 一种轴类零件边缘精确定位方法 [J]. 计算机工程与应用, 2012, 10 (2): 238-241.
- [7] 刘振宇, 赵彬, 邹凤山. 机器视觉技术在工件分拣中的应用 [J]. 计算机应用与软件, 2012, 11 (3): 87-91.
- [8] Chen Y J, Zhang ZH J. Correction of CCD pixel nonuniformity [J]. Opt. Precision Eng, 2004, 12 (2): 216-220.
- [9] Li Z H, Ding Z H L, Yuan F. Global calibration method for multi-vision measurement system with coplanar targets [J]. Opt. Precision Eng., 2008, 16 (3): 467-472.
- [10] Ming J Tsai, Lee H W, Ann N J. Machine vision based path planning for a robotic golf club head welding system [J]. Robotics and Computer-Integrated Manufacturing, 2011, 27 (2): 843-849.

(上接第 1459 页)

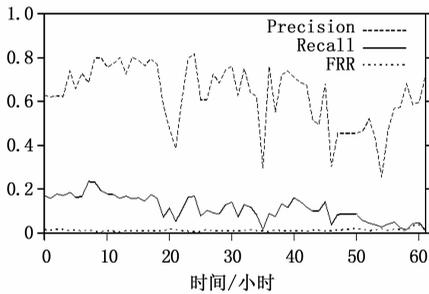


图3 LDA的图形表示

和执行效率快等优点。

参考文献:

- [1] 史佩昌, 王怀民, 尹刚, 等. 云服务传递网络资源动态分配模型 [J]. 计算机学报, 2011, 34 (12): 2305-2318.
- [2] Sun D, Chang G, Miao C, et al. Modelling and Evaluating a High Serviceability Fault Tolerance Strategy in Cloud Computing Environments [J]. Int. J. Secur. Netw., 2012, 7 (2): 196-210.
- [3] Nguyễn T A N, Désidéri J. Resilience for Collaborative Applications on Clouds: Fault-tolerance for Distributed HPC Applications [A]. in ICCSA12 [C]. Berlin, Heidelberg, 2012: 418-433.
- [4] Yilmaz C. Using Hardware Performance Counters for Fault Localization [A]. in VALID'10 [C]. Washington, DC, USA, 2010: 87-92.
- [5] Yadav S K, Kalra P K. Automatic Fault Diagnosis of Internal Combustion Engine Based on Spectrogram and Artificial Neural Network [A]. in ROCOM10 [C]. Stevens Point, Wisconsin, USA, 2010: 101-107.
- [6] 马笑潇, 黄席樾, 柴毅, 等. 基于 SVM 的二叉树多类分类算法

及其在故障诊断中的应用 [J]. 控制与决策, 2003, 18 (3): 272-276

- [7] Ligeza A, Koscielny J. A New Approach to Multiple Fault Diagnosis: A Combination of Diagnostic Matrices, Graphs, Algebraic and Rule-Based Models. The Case of Two-Layer Models [J]. Int. J. Appl. Math. Comput. Sci., 2008, 18 (3): 465-476
- [8] Tan S C, Lim C P, Rao M V C. A Hybrid Neural Network Model for Rule Generation and Its Application to Process Fault Detection and Diagnosis [J]. Eng. Appl. Artif. Intell., 2007, 20 (3): 203-213.
- [9] Wang M, Hu N, Qin G. A Method for Rule Extraction Based on Granular Computing: Application in the Fault Diagnosis of a Helicopter Transmission System [J]. J. Intell. Robotics Syst., 2013, 71 (4): 445-455.
- [10] Wang B, Cao W, Ma L, et al. Fault Diagnosis Approach Based on Qualitative Model of Signed Directed Graph and Reasoning Rules [A]. in FSKD05 [C]. Berlin, Heidelberg, 2005: 339-343.
- [11] Cohen I, Goldszmidt M, Kelly T, et al. Correlating Instrumentation Data to System States: A Building Block for Automated Diagnosis and Control [A]. in OSDI'04 [C]. Berkeley, CA, USA, 2004: 16-16.
- [12] Yin G, Zhang Y, Li Z, et al. Online Fault Diagnosis Method Based on Incremental Support Vector Data Description and Extreme Learning Machine with Incremental Output Structure [J]. Neurocomput., 2014, 12 (8): 224-231.
- [13] Patan K, Witczak M, Korbicz J O Z. Towards Robustness in Neural Network Based Fault Diagnosis [J]. Int. J. Appl. Math. Comput. Sci., 2008, 18 (6): 443-454.
- [14] Sune V, Carrasco J A. A failure-distance based method to bound the reliability of nonrepairable fault-tolerant systems without the knowledge of minimal cuts [J]. Reliability IEEE Transactions on, 2001, 50 (1): 60-74.