

基于云平台日志的故障检测和复杂构件系统即时可靠性度量研究

王博, 华庆一, 舒新峰

引用本文

王博, 华庆一, 舒新峰. 基于云平台日志的故障检测和复杂构件系统即时可靠性度量研究[J]. 计算机科学, 2022, 49(12): 125-135.

WANG Bo, HUA Qing-yi, SHU Xin-feng. [Study on Anomaly Detection and Real-time Reliability Evaluation of Complex Component System Based on Log of Cloud Platform](#) [J]. Computer Science, 2022, 49(12): 125-135.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[车联网中基于航位推算的故障检测方法](#)

Fault Detection Based on Dead Reckoning in VANETs

计算机科学, 2022, 49(12): 319-325. <https://doi.org/10.11896/jsjcx.220200155>

[基于联邦学习的暖通空调系统故障检测与诊断](#)

Fault Detection and Diagnosis of HVAC System Based on Federated Learning

计算机科学, 2022, 49(12): 74-80. <https://doi.org/10.11896/jsjcx.220700280>

[考虑风险规避和资金约束的低碳供应链决策研究](#)

Study on Decision-making for Low-carbon Supply Chain with Capital Constraint and Risk Aversion

计算机科学, 2022, 49(11A): 210900104-6. <https://doi.org/10.11896/jsjcx.210900104>

[一种面向SDN的移动网络可靠性评估算法](#)

SDN Oriented Mobile Network Reliability Evaluation Algorithm

计算机科学, 2022, 49(11A): 211000080-8. <https://doi.org/10.11896/jsjcx.211000080>

[结合深度学习与改进的极限学习机的集成学习胸腺瘤CT图像预测方法](#)

Thymoma CT Image Prediction Method Based on Deep Learning and Improved Extreme Learning Machine Ensemble Learning

计算机科学, 2022, 49(11A): 211200097-6. <https://doi.org/10.11896/jsjcx.211200097>

基于云平台日志的故障检测和复杂构件系统即时可靠性度量研究

王 博^{1,2,3} 华庆一¹ 舒新峰²

1 西北大学信息科学与技术学院 西安 710119

2 西安邮电大学计算机学院 西安 710121

3 西安邮电大学陕西省网络数据智能处理重点实验室 西安 710121

(wangbo@xupt.edu.cn)

摘 要 可靠性、可用性和安全性是软件质量度量的 3 个重要指标,而软件可靠性是软件质量最重要的指标。传统的软件可靠性评估将软件系统看作一个整体或者将软件系统调用结构视为静态结构。现今的软件结构发生了很多的改变,典型的有自主、协同、演进、动态和自适应等特征,已经渗入到当前的复杂网络结构软件系统中,传统的可靠性评估和预测方法已经不能适应当前复杂网络生态环境下的软件系统。在当前“软件定义一切”的高速信息化社会中,海量的信息系统产生了大规模的数据资源。现代信息系统的异构性、并行性、复杂性以及巨大的规模导致了日志资源的多样和复杂,基于系统日志的精准分析和故障预测对构建安全可靠的系统尤为重要。现有文献研究故障预测和软件可靠性的技术颇多,但是较少针对海量日志以及复杂构件进行软件即时可靠性度量。文中在系统分析日志解析、特征提取、故障检测、预测评估到即时可靠性计算的日志处理全过程中,使用集成学习模型对海量系统日志进行分析和故障预测,与传统的机器学习方法进行了比较,提高了故障预测的准确率、召回率和 F1 值;针对预测召回率低的情况,采用召回率修正即时可靠性的评估,较大程度地提高了即时可靠性的精度;根据个体的可靠性,通过基于马尔可夫理论的系统可靠性度量微服务复合构件的可靠性,从而为智能化运维提供精确的数据基础和故障定位依据。

关键词: 日志解析;故障检测;可靠性评估;根因分析;集成学习;复杂构件

中图法分类号 TP311.5

Study on Anomaly Detection and Real-time Reliability Evaluation of Complex Component System Based on Log of Cloud Platform

WANG Bo^{1,2,3}, HUA Qing-yi¹ and SHU Xin-feng²

1 School of Information Science and Technology, Northwestern University, Xi'an 710119, China

2 School of Computer Science, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

3 Shaanxi Key Laboratory of Network Data Intelligent Processing, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

Abstract Reliability, usability and security are three important indicators of software quality measurement, and software reliability is the most important indicator. Software system is regarded as a whole or viewed invocation structure of software as static structure in traditional software reliability evaluation and prediction. Today's software architecture has changed significantly. Typical features such as autonomy, coordination, evolution, dynamic and adaptive have been infiltrated into the current complex network software system. Traditional reliability evaluation and prediction methods cannot adapt to such software architecture or environment. Currently, in the society of high-speed information, "software defines everything". Massive information systems generate large-scale data resources. The diversity and complexity of log resources are the results of heterogeneity, parallelism, complexity and huge scale of modern information systems. Accurate analysis and anomaly prediction based on logs are particularly important for building safe and reliable systems. There are a lot of research on anomaly prediction and software reliability in the existing literatures, but there is little about real-time software reliability measurement for massive logs and complex network component systems. Accordingly, based on the complete procedures of log processing, from its analysis, feature extraction, anomaly detection and prediction evaluation to real-time reliability evaluation, this paper uses ensemble learning model to analyze and pre-

到稿日期:2022-02-17 返修日期:2022-06-05

基金项目:陕西省科技攻关(2016GY-123);陕西省重点研发项目(2020GY-210);河南省工业科学技术研究项目(212102210418);国家自然科学基金(61272286)

This work was supported by the Key Science and Technology Program of Shaanxi Province, China(2016GY-123), Key Research and Development Projects of Shaanxi Province(2020GY-210), Industrial Science and Technology Research Project of Henan Province(212102210418) and National Natural Science Foundation of China(61272286).

通信作者:华庆一(nwuhuaqingyi@163.com)

dict anomaly of the massive system logs. Comparisons with the traditional machine learning methods are made to improve the accuracy, recall rate and F1 value of anomaly prediction. The evaluation result is used to correct the real-time reliability in view of the low predicted recall rate, which greatly improves the accuracy of real-time reliability. According to the individual reliability, the system reliability based on Markov theory is used to measure the reliability of microservice composite components, so as to provide accurate data basis and anomaly location basis for intelligent operation and maintenance.

Keywords Log parsing, Anomaly detection, Reliability evaluation, Root cause analysis, Ensemble learning, Complex components

1 引言

当前,随着“个人计算”向“社会计算”发展以及工业物联网的兴起,海量的服务赋之云端,云计算允许通过网络随时随地且便捷地按需使用可配置的计算资源共享池(如网络、服务器、存储、应用和服务)等模式,深入当前的人类社会中,一切皆服务(X as a Service, XaaS)强调下游对上游按照契约提供服务,隐藏实现细节。云计算的本质就是 XaaS^[1]。其次,随着软件技术的不断发展,软件体系结构发生了巨大的变化^[2]。网络结构软件体系结构从单体结构、垂直结构、面向服务体系结构(Service-Oriented Architecture, SOA)到微服务构架不断演进,软件结构趋于动态、离散和分布,如 SOA 的服务离散和中心化,微服务的去中心化的特征就集中体现了这种趋势^[3]。

在软件系统开发和维护过程中,海量系统运行日志成为了重要的监视系统剖析异常行为的重要资源^[4]。系统日志记录了不同关键点的系统状态和重要事件,异常检测和可靠性预测是构建安全可靠系统的关键^[5]。在当今的大型软件系统中,一个故障可能会影响数百万的用户^[6]。随着日志信息量的不断增长,从宝贵的日志资源中挖掘有效的支持智能化运维的决策信息,通过人工检索已经很难完成,即使有长期运营经验的工程师或者分析专家亦难以系统化地进行分析^[5],而且手工分析效率低下,工作量巨大且精确度差。因此,寻求高效、精准的海量日志数据的自动处理方法^[6-10]、日志异常分析^[11-14]、根因分析和故障预测^[15-18]以及可靠性评估成为了基于系统日志的分析预测技术研究的重点^[19-20]。

随着近年来人工智能的发展,诸如神经网络^[21]、深度学习^[22]、集成学习^[23-25]、主动学习^[26]等技术不断成熟,结合非结构化日志解析技术,使得海量日志信息的高效精准预测分析成为可能,从而为智能化运维^[19,27-28]提供精确的数据基础和故障定位依据。

可靠性是衡量软件系统质量的重要指标,尤其对于在线服务软件,如金融、电信、医疗、教育、能源、制造、工业物联网和物流等重要行业,没有及时精准的可靠性评估和预测等手段,任何意想不到的错误都有可能中断服务。随着应用软件的快速发展,软件的体系结构、部署环境和测试方法均发生了较大的变化,因此在线可靠性的预测分析也随之发生变化。在线可靠性的预测分析变得困难和复杂,主要原因如下:

(1)软件体系结构发生了巨大变化,从单体、垂直、SOA到微服务的变化,使得传统的可靠性评估方法不能适应软件可靠性的评估。软件的部署环境发生了巨大变化,如微服务的部署依赖于微服务运行框架(Dubbo, SpringCloud等),微服务框架运行于 Docker 的容器, Docker 运行于云虚拟服务器,虚拟服务器构建于基础硬件平台。

(2)功能服务和非功能服务相互依赖,并且服务组件冗余部署,服务在调用路径上是动态的,其中服务发现、服务治理、服务容错、服务网关、服务配置、负载均衡、消息总线、服务跟踪等成熟组件相互组合形成了较大的应用,为可靠性的分析带来了新的挑战。

(3)传统的软件可靠性评估将软件系统看作一个整体或者将软件系统调用结构视为静态结构。目前的软件结构发生了重要的改变,典型的有自主、协同、反应、演进、动态和自适应等特征,已经渗入到当前的复杂网构软件系统中。传统的可靠性评估和预测方法已经不能适应当前复杂网络生态环境下的软件系统。

(4)对于复杂系统未来的可靠性预测成为了当前研究的重点,这将为构建自适应的软件系统提供依据,对提高软件的可靠性具有重要意义。

(5)海量的测试可靠性的测试数据和运行数据为在线可靠性评估和度量带来了新的挑战。

本文在系统分析日志解析、特征提取、故障检测、预测评估和即时可靠性计算的日志处理全过程中,使用集成学习模型对海量系统日志进行分析,并与传统的机器学习方法逻辑回归、PCA-Q 统计预测模型和支持向量机等进行了比较,由实验结果可以发现集成学习模型对海量系统日志异常预测有较高的准确率、召回率和 F1 值;针对预测的召回率低的情况,本文采用召回率修正即时可靠性的计算,较大程度地提高了即时可靠性的精度;根据个体可靠性和基于马尔可夫理论的系统可靠性来度量微服务类型的复合构件的可靠性,从而为智能化运维提供精确的数据基础和故障定位依据。

2 相关工作

日志在软件系统的开发和运维上具有重要的作用和意义^[19],传统的日志解析工作依赖于正则表达式和 grok 模式等技术^[4]来提取事件模板和关键参数。为了实现日志的自动解析工作,很多日志解析工作被学术界和工业界提出,包括频繁集模式^[29]及其扩展^[30]、迭代分区^[31]、层次聚类^[32]、最长公共子序列计算^[33]和解析树^[34],这些方法能够自动从日志数据中学习模式并自动生成公共事件模板。Zhu 等^[4]在日志自动分析工具中详细介绍了 13 种工具的实现。

日志的异常检测是智能化运维的重要环节,近年来针对异常检测提出了很多方法,这些方法被广泛地应用于系统的开发、运维以及数据可视化领域。传统的基于时间序列的异常检测方法主要采用参数估计,包括高斯模型、回归模型和混合参数模型^[35]。目前,随着机器学习的迅速发展,研究者很快将机器学习模型应用到相关研究领域。Du 等采用 LSTM 深度模型进行系统日志异常检测^[36]。Liu 等^[37]提出了一种

基于随机森林分类器的自动异常检测方法。Timčenko 等^[35]利用机器学习中的集成学习方法对网络入侵现象进行检测。Nedelkoski 等^[38]采用多模态的深度学习的系统跟踪数据进行异常检测。基于监督学习的异常检测方法通常需要大量的带有标记的样本,而且样本的分布不均衡^[39]。目前无监督的异常检测大都采用基于统计的分析技术。

基于传统假设的可靠性预测模型是早期经典的软件可靠性增长模型,其中基于非齐次泊松过程和马尔可夫过程的软件可靠性增长模型是软件可靠性增长模型领域中应用最普遍并且研究最广的一类^[40]。1971年, Jelinski 等假设软件测试阶段的失效次数服从马尔可夫过程,提出了 J-M 模型。1979年, Goel 等在 J-M 模型的基础上,提出了基于非齐次泊松过程的 G-O 模型。后期众多文献提出的模型都是以 G-O 模型为基础的改进软件可靠增大模型^[41]。

传统的将软件系统看作黑盒模型的可靠性分析,已经不能适应新型的基于构件的开发模型,结合系统的构架信息及组件可靠性分析系统可靠性的方法,逐渐成为软件系统可靠性分析的主导^[42]。文献[43]在公认的软件缺陷数据集上提取软件失效特征,使用朴素贝叶斯算法在所提特征的基础上建立软件可靠性预测模型,基于朴素贝叶斯的预测模型的平均召回率为 71%,误报率为 25%。文献[44]针对网页代码的缺陷检测和预防,提出了基于最近邻算法的软件可靠性增长模型,实验结果表明,聚类算法在建立软件可靠性增长模型时具有较好的线性判别能力。Jin 等针对 SVM 参数寻优困难的问题,提出了一种改进的分布算法,使用改进的分布式算法来寻找 SVM 模型参数,实验结果证明使用分布式算法寻找 SVM 模型参数能有效提升 SVM 模型的性能^[45]。

监督学习虽然在建立软件可靠性增长模型时具有精度高的优点,但是软件缺陷数据难以收集,训练数据较少时容易导致模型的泛化能力较差。为了解决这一问题,学者尝试使用半监督学习算法建立软件可靠性增长模型^[25,46]。Harikesh 等认为不同素质的软件开发人员开发的软件模块质量不同,应该对不同模块的质量进行分类,因此提出了使用半监督学习算法 k -means 对软件代码模块的质量进行标记^[47]。Seliya 等在软件缺陷数据集上应用 EM 算法建立半监督预测模型,实验结果表明,在软件缺陷标签不足时,EM 算法能够有效提升预测模型的性能^[48-49]。

3 故障预测模型

在进行故障预测时,可采用不同的机器学习算法,如逻辑回归(Logistic Regression, LR)、PCA-Q 统计(Principal Components Analysis, PCA)、支持向量机(Support Vector Machines, SVM)以及集成学习的方法,这里对涉及本文重点和难点的内容加以强调说明。

逻辑回归是假设样本服从伯努利分布,利用极大似然估计和梯度下降求解的分类模型,在分类、预测评估等领域有着广泛的应用。其基本的过程是构造假设函数和损失函数,通过损失函数最小化求目标函数的各个参数,通过求解的目标函数对测试数据进行预测并对预测数据进行评估。下面对前三条涉及的方法的推导过程和公式进行详细说明。

(1) 构造假设函数

逻辑回归的假设函数可以看作是线性回归方程的结果经过 Sigmoid 函数得到的结果,线性回归方程的表达式如下:

$$\theta_0 + \theta_1 x_1 + \dots + \theta_n x_n = \sum_{i=1}^n \theta_i x_i = \theta^T x \quad (1)$$

Sigmoid 函数的表达式如下:

$$g(z) = \frac{1}{1+e^{-z}} \quad (2)$$

将式(1)通过逻辑函数式(2)即获得如下假设函数:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}} \quad (3)$$

函数 $h_\theta(x)$ 表示样本被预测为正例的概率,容易得到样本被预测为正例和负例的概率如下:

$$P(y=1|x;\theta) = h_\theta(x) \quad (4)$$

$$P(y=0|x;\theta) = 1 - h_\theta(x)$$

式(4)可以合并为:

$$P(y|x;\theta) = (h_\theta(x))^y (1-h_\theta(x))^{1-y} \quad (5)$$

(2) 构造损失函数

对预测结果的概率表示式(5)取似然函数,得到如下的似然函数:

$$\begin{aligned} L(\theta) &= \prod_{i=1}^m P(y^{(i)} | x^{(i)}; \theta) \\ &= \prod_{i=1}^m (h_\theta(x^{(i)}))^{y^{(i)}} (1-h_\theta(x^{(i)}))^{1-y^{(i)}} \end{aligned} \quad (6)$$

对其取对数操作得到对数似然函数。

$$\begin{aligned} l(\theta) &= \log L(\theta) \\ &= \sum_{i=1}^m (y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))) \end{aligned} \quad (7)$$

令:

$$J(\theta) = -\frac{1}{m} l(\theta) \quad (8)$$

得到损失函数的最终形式。

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m (y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))) \quad (9)$$

(3) 通过损失函数最小化求目标函数的各个参数,对损失函数求梯度。

$$\begin{aligned} \frac{\partial}{\partial \theta_j} J(\theta) &= -\frac{1}{m} \sum_{i=1}^m \left(y^{(i)} \frac{1}{h_\theta(x^{(i)})} \frac{\partial}{\partial \theta_j} h_\theta(x^{(i)}) - (1-y^{(i)}) \right. \\ &\quad \left. \frac{1}{1-h_\theta(x^{(i)})} \frac{\partial}{\partial \theta_j} h_\theta(x^{(i)}) \right) \\ &= -\frac{1}{m} \sum_{i=1}^m \left(y^{(i)} \frac{1}{g(\theta^T x^{(i)})} - (1-y^{(i)}) \frac{1}{1-g(\theta^T x^{(i)})} \right) \\ &\quad \frac{\partial}{\partial \theta_j} g(\theta^T x^{(i)}) \\ &= -\frac{1}{m} \sum_{i=1}^m \left(y^{(i)} \frac{1}{g(\theta^T x^{(i)})} - (1-y^{(i)}) \frac{1}{1-g(\theta^T x^{(i)})} \right) \\ &\quad g(\theta^T x^{(i)}) (1-g(\theta^T x^{(i)})) \frac{\partial}{\partial \theta_j} \theta^T x^{(i)} \\ &= -\frac{1}{m} \sum_{i=1}^m (y^{(i)} (1-g(\theta^T x^{(i)})) - (1-y^{(i)}) \\ &\quad g(\theta^T x^{(i)})) x_j^{(i)} \\ &= -\frac{1}{m} \sum_{i=1}^m (y^{(i)} - g(\theta^T x^{(i)})) x_j^{(i)} \end{aligned}$$

$$= \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad (10)$$

其中, θ 为最终更新式, α 为学习率。

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad (11)$$

LR 在确定了模型的形式后, 通过最大似然估计法来实现最小散度, 从而求出模型参数。

PCA 即主成分分析技术, 旨在利用降维的思想, 把多指标转化为少数综合指标。主成分分析经常用于减少数据集的维数, 同时保持数据集对方差贡献最大的特征。

PCA 降维算法的流程如算法 1 所示。

算法 1 PCA 降维算法

输入: n 维样本集 $\mathbf{D} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(m)})$, 降维到维度 n' ($n' < n$)

输出: 降维后的样本集 \mathbf{D}'

1. 对所有的样本进行中心化: $\mathbf{x}^{(i)} = \mathbf{x}^{(i)} - \frac{1}{m} \sum_{j=1}^m \mathbf{x}^{(j)}$;
2. 计算样本的协方差矩阵 \mathbf{XX}^T ;
3. 对矩阵 \mathbf{XX}^T 进行特征值分解;
4. 取出最大的 n' 个特征值对应的特征向量 $(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{n'})$;
5. 将样本集中的样本 $\mathbf{x}^{(i)}$ 转化成新样本 $\mathbf{z}^{(i)} = \mathbf{W}^T \mathbf{x}^{(i)}$;
6. 得到输出样本集 $\mathbf{D}' = (\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(m)})$ 。

在不指定降维后的 n' 的值时, 通过降维得到的主成分比重阈值 t , 对样本集进行降维, 阈值 t 在 $(0, 1]$ 之间, 假如我们的 n 个特征值为 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, 则 n' 为:

$$\frac{\sum_{i=1}^{n'} \lambda_i}{\sum_{i=1}^n \lambda_i} \geq t \quad (12)$$

PCA 方法对日志数据只能起到降维作用, 特征提取并降维的数据采用建立统计量假设检验的方法, 判断日志数据是否背离主元模型。Q 统计量 (即平方预测误差 SPE 统计量) 刻画了某时刻预测值对主元模型的偏离程度。 \tilde{S} 空间中的 SPE 统计量定义为:

$$SPE = \|\tilde{C}_x\|^2 \leq Q_{\alpha}^2 \quad (13)$$

SPE 或者 Q 统计量代表了数据中没有出现主元模型所解释的变化。当 SPE 过大时, 说明系统运行过程中出现了异常的情况, 正常运行时根据正常数据所建立的模型已经不再适用, 控制限的计算是建立在一定的假设的基础上, 当检验水平为 α 时, SPE 控制限的计算式为^[50]:

$$Q_0 = \theta_1 \left[\frac{C_{\alpha} \cdot \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}} \quad (14)$$

其中:

$$\theta_i = \sum_{j=k+1}^n \lambda_j^i, i=1, 2, 3$$

$$h_0 = i - \frac{2\theta_1 \theta_3}{3\theta_2^2} \quad (15)$$

式(14)、式(15)中, λ_i 是 X 的协方差矩阵的特征值, C_{α} 表示正态分布在检验水平临界值 α 下的取值, k 是主元模型中保留的主元个数, n 是全部主元个数。

支持向量机是定义在特征空间的间隔最大的线性分类器; SVM 还包括核技巧, 这使它成为了非线性分类器。SVM 的学习策略就是间隔最大化, 可形式化为求解凸二次规划的问题。SVM 的学习算法就是求解凸二次规划的最优化算法。

支持向量机的基本思想是求解能够正确划分训练数据集并且几何间隔最大的分离超平面。

假设给定特征空间的训练数据集 $\mathbf{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$ 。其中, $\mathbf{x}_i \in \mathbb{R}^n$; $y_i \in \{+1, -1\}$, $i=1, 2, \dots, N$; \mathbf{x}_i 为第 i 个特征向量; y_i 为类标记, 当它等于 +1 时为正则例, 等于 -1 时为负例。再假设训练集为线性可分的。

对于给定的数据集 T 和超平面 $\mathcal{W} \cdot \mathcal{X} + b = 0$, 定义超平面关于样本点 (\mathbf{x}_i, y_i) 的几何间隔为:

$$\gamma_i = y_i \left(\frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot \mathbf{x}_i + \frac{b}{\|\mathbf{w}\|} \right) \quad (16)$$

超平面关于所有样本点的几何间隔的最小值为:

$$\gamma = \min_{i=1, 2, \dots, N} \gamma_i \quad (17)$$

根据以上定义, SVM 模型求解的最大分隔超平面可以表示为以下约束最优化问题:

$$\max_{\mathbf{w}, b} \gamma \quad (18)$$

$$\text{s. t. } y_i \left(\frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot \mathbf{x}_i + \frac{b}{\|\mathbf{w}\|} \right) \geq \gamma, i=1, 2, \dots, N$$

将约束条件两边同除以 γ , 得到:

$$y_i \left(\frac{\mathbf{w}}{\|\mathbf{w}\| \gamma} \cdot \mathbf{x}_i + \frac{b}{\|\mathbf{w}\| \gamma} \right) \geq 1 \quad (19)$$

因为 $\|\mathbf{w}\|$, γ 都是标量, 所以为了表达简便, 令:

$$\mathbf{w} = \frac{\mathbf{w}}{\|\mathbf{w}\| \gamma} \quad (20)$$

$$b = \frac{b}{\|\mathbf{w}\| \gamma}$$

得到如下表达式:

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, i=1, 2, \dots, N \quad (21)$$

又因为最大化 γ 等价于最大化 $\frac{1}{\|\mathbf{w}\|}$, 也就等价于最小化 $\frac{1}{2} \|\mathbf{w}\|^2$ ($1/2$ 是为了后面求导以后形式简洁, 不影响结果), 因此 SVM 模型的求解最大分隔超平面又可以表示为以下约束最优化问题:

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad (22)$$

$$\text{s. t. } y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, i=1, 2, \dots, N$$

这是含有不等式约束的凸二次规划问题, 可以对其使用拉格朗日乘子法得到其对偶问题, 并对该问题进行求解。

在机器学习的有监督学习算法中, 目标是学习出稳定的且在各个方面都表现较好的模型, 但实际情况往往并不是这么理想, 有时只能得到多个有偏好的模型 (弱监督模型在某些方面的表现较好)。集成学习组合的多个弱监督模型, 以期得到更好、更全面的强监督模型。集成学习潜在的思想是即便某弱分类器得到了错误的预测, 其他的弱分类器也可以将错误纠正回来, 以达到减小方差、偏差或改进预测效果的目的。集成学习主要分为 3 类:

(1) 序列集成方法 boosting。参与训练的基础学习器按照串行方式生成。将基分类器层层叠加, 每一层在训练时, 对前一层基分类器分错的样本给予更高的权重, 在测试时根据各层分类器的结果加权得到最终结果, 这样就利用了基础学习器之间的依赖关系。通过对之前训练中错误标记的样本赋予较高的权重, 可以提高整体的预测效果。

(2)并行集成方法 bagging。在训练过程中参与训练的基学习器之间无强依赖,可以进行并行训练(如 Random Forest)。为了使基分类器之间相互独立,可以将训练集分为若干子集(训练样本较少时子集之间可能有重叠)。在最终决策时,每个个体单独做出判断,再通过投票的方式作出最后的集体决策。并行方法的原理是利用基础学习器之间的独立性,通过平均可以显著减少错误。

(3)模型融合集成方法 Stacking。Stacking 方法指训练模型用于组合其他各个模型。首先训练多个不同的模型,然后把训练的各个模型的输出作为输入来训练模型,以得到最终的输出。理论上,Stacking 只要采用合适的模型组合策略,就可以表示上文提到的两种集成学习方法,实际中通常使用 logistic 回归作为组合策略。如图 1 所示,先在整个训练数据集上通过 Bootstrap 抽样得到各个训练集合,从而得到一系列分类模型,然后将输出用于训练第二层分类器。

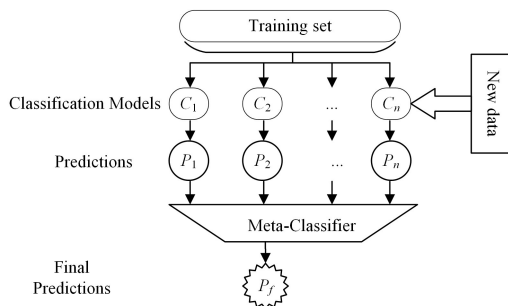


图 1 集成学习模型融合 Stacking

Fig. 1 Model fusion Stacking of ensemble learning

集成学习法的特点如下:聚集多种分类方法,以提高分类的准确率;集成学习法由训练数据构建一组基分类器,然后通过每个基分类器的预测进行投票来进行分类;通常集成分类器的分类性能优于单个分类器。

4 复杂系统即时可靠性度量

首先对原始日志数据进行初步处理,把无结构的数据通过规则变换变成半结构的日志数据,在此采用日志提取工具^[33-34](drain 或者 spell);其次将初步处理的数据读取后分成样本集合训练集,对数据进行特征提取并进行归一化后,进行故障预测并对预测结果进行评估;最后完成系统即时可靠性的计算。根据原始日志数据的故障预测以及复杂组合构件即时可靠性计算的整体过程如图 2 所示。

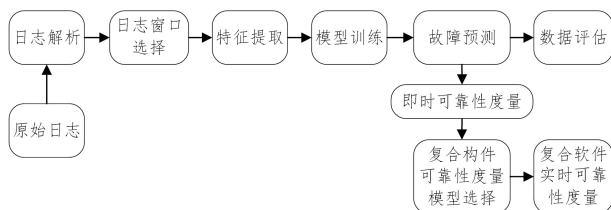


图 2 原始日志数据的故障预测以及复杂组合构件即时可靠性计算的整体过程

Fig. 2 Overall process of anomaly prediction of original log data and real-time reliability calculation of complex components

4.1 数据处理和特征提取

把原始日志数据通过 drain 或者 spell 等工具提炼成半结

构的数据和模板,如图 3 所示。

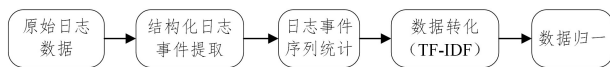


图 3 日志数据处理和特征提取

Fig. 3 Log data processing and feature extraction

日志是由固定部分和可变部分组成的纯文本,这些部分在不同的事件中可能会有所不同。例如,对于给出的日志“Receiving block from src:/10.251.127.191:43606 dest:/10.251.127.191:50010”和“Receiving block from src:/10.251.127.232:45808 dest:/10.251.127.232:50127”,“Receiving block from src:…dest:”被认为是不变的部分,其余的部分被认为是可变部分。开发人员在源代码中预先定义了常量部分,并且变量部分通常是动态生成的,在异常检测中无法得到很好的利用。日志解析的目的是将常量部分与变量部分分开,形成成熟的日志事件,如图 4 所示。

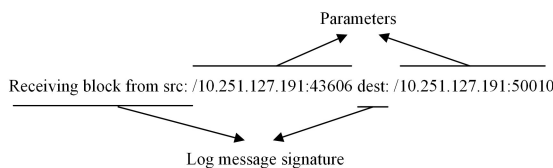


图 4 日志信息的不变部分和可变部分

Fig. 4 Invariant and variable parts of log information

该步骤的主要目的是从日志事件中提取有价值的特征,这些特征可以被输入异常检测模型。特征提取的输入是日志解析步骤中生成的日志事件,输出是事件计数矩阵。为了提取特征,首先需要将日志数据分为不同的组,其中每个组代表一个日志序列,因此窗口被应用于将日志数据集划分成有限块。日志解析中使用 3 种不同类型的窗口:固定窗口、滑动窗口和会话窗口。固定窗口和滑动窗口都基于时间戳,时间戳记录每个日志的发生时间。每个固定窗口都有其大小,表示时间跨度或持续时间;滑动窗口由两个属性组成,即窗口大小和步长;会话窗口基于标识符而不是时间戳,标识符用于在一些日志数据中标记不同的执行路径。

首先对日志数据进行特征提取,然后选择不同的预测算法进行故障预测,并对预测结果进行评估和度量。数据处理和特征提取的算法如算法 2 所示。

算法 2 数据处理和特征提取算法

输入:原始日志数据文件,异常标记文件

输出:特征提取矩阵

1. 从原始日志文件中获取结构化日志,并用拼写方法标记每个日志的事件;
2. 加载结构化日志文件,通常以 1:1 的比例获得训练集和测试集;
3. 统计训练集和数据集中每个数据块 id 对应的事件数;
4. 将每个数据块 id 对应的不同事件数记录为表的新列,即新数据集的行为数据块 id,新表的列为不同的事件;
5. 数据补全:空数据用零填充;
6. 使用 tf-idf 转换训练集和测试集,然后使用零均值或 Sigmoid 标准化测试数据集;
7. 返回特征提取的训练数据和测试数据。

根据日志提取的序列数据统计出事件调用序列次数后,采用 TF-IDF 对日志序列调用次数进行特征提取,TF-IDF 用

于评估字词在文件或语料库中的重要程度。字词的重要性随着其在文件中出现的次数成正比增加,但同时会随着其在语料库中出现的频率成反比下降。TF-IDF的主要思想是,如果某个词或短语在文章中出现的频率 TF 高,并且在其他文章中很少出现,则认为此词或者短语具有很好的类别区分能力,适合用于分类。

4.2 故障预测模型

对系统日志数据进行特征提取并归一化后,在此采用不同的机器学习算法进行故障预测,如逻辑回归(LR)、PCA-Q 统计、支持向量机(SVM)以及集成学习的方法。第 3 节已对这些方法进行了详细的描述,下文将对本实验中采用的特殊方法和参数进行说明。

在所有上述预测模型中,在训练和预测之前,对日志数据进行分析,特征提取和归一化都相同,即不同的预测模型所采用的训练数据和测试数据完全一致。

LR,SVM 和 PCA-Q 统计预测模型在第 3 节和文献[5]中有详细说明,在此不再赘述。集成学习 Boosting 预测模型采用了 Sklearn 中的 AdaBoosting 分类模型,设置弱分类器为 200 个决策树分类,学习率 learning-rate 取 0.8;集成学习 Bagging 预测模型采用了 Sklearn 中的 Bagging 分类模型,基分类器仍旧采用决策树分类器模型,基分类器的个数设置为 500,采用有放回的抽样方式;集成学习 Stacking 预测模型采用了 Sklearn 中的 Stacking 分类模型,基分类器分别采用支持向量机、决策树、梯度提升机(Gradient Boosting Machine, GBM),目标分类器采用逻辑回归分类器。

在后续的仿真实验中,分别采用了 2kB,100kB 和 600kB 的日志数据集进行实验,通过实验结果发现,在样本量较少的情况下,传统的方法表现较好,但是在 600kB 的数据集上,集成学习的 3 种方法在准确率、召回率和 F1 值方面都有大幅度的提高。

4.3 即时可靠性计算

针对云平台上某种服务在某个事件窗口内的日志信息,获取其故障或者异常信息,根据该时间段上的故障或者异常可以度量其可靠性,而且可以根据系统当前所产生的数据来评估系统的即时可靠性。假定在某个时间段 $t = t_n - t_n > 0$,服务 w 在 t 时间累积调用 C_n 次,其中 n 次调用失效,在 t 时刻调用 C_k 次, k 次失效,则失效率可以记为:

$$F(w) = \frac{n-k}{C_n - C_k} \quad (23)$$

根据失效率,可以求得可靠性:

$$R(w) = 1 - F(w) = 1 - \frac{n-k}{C_n - C_k} \quad (24)$$

本文将其简化为:

$$R_c(w) = 1 - \frac{n_{te}}{n_{ta}} \quad (25)$$

其中, n_{te} 是给定时间段的系统服务调用预测故障事件次数, n_{ta} 是系统服务调用所有的调用事件次数。

由于系统存在误差,用 α 修正该公式,得:

$$\hat{R}_c(w) = 1 - \frac{\alpha * n_{te}}{n_{ta}} \quad (26)$$

其中, α 取召回率的倒数作为修正,取得了与真实结果较相近的评估值,并在实验仿真中得到了验证。即时可靠性的计算

与评估,尤其在可视化的展示中,为智能化运维提供了较好的决策依据。

4.4 复杂构件的可靠性度量

基于状态的系统可靠性评估方法使用软件控制流图来描述系统的结构,将系统的执行过程看作一个状态转移过程^[51],即组件某一时刻的状态仅与上一时刻的状态有关,而与其他状态无关。利用随机过程的理论对系统可靠性进行分析,典型的是 Markov 过程理论,文献[52]对该类方法做了较详细的总结。基于该理论的方法主要有时间离散的马尔可夫模型^[53-54],即将应用程序建模成时间离散的马尔可夫链(Discrete Time Markov Chain,DTMC),软件每次运行都对应一个终止状态;时间连续马尔可夫模型^[55],即将应用程序建模成时间连续马尔可夫链(Continuous Time Markov Chain,CTMC)。表 1 列出了基于马尔可夫模型的不同结构类型的可靠性的计算式。

先考虑每个构件都未备份的情况,通常只要某一构件出现问题,任务就无法完成。假设任务由 C_1 开始,涉及 C_1, C_2, \dots, C_k ,在任务执行过程中各构件之间会进行必要的通信。假设系统中每个构件的可靠性是独立的,下一个要执行任务的构件只与上一个向其发出任务请求的构件有关,因此完成任务的整个过程可以被看作一个马尔可夫过程,其状态转移矩阵为:

$$M = \begin{pmatrix} 0 & P_{12}r_2 & P_{13}r_3 & \cdots & P_{1(n-1)}r_{n-1} & P_{1n}r_n \\ P_{21}r_1 & 0 & P_{23}r_3 & \cdots & P_{2(n-1)}r_{n-1} & P_{2n}r_n \\ P_{31}r_1 & P_{32}r_2 & 0 & \cdots & P_{3(n-1)}r_{n-1} & P_{3n}r_n \\ P_{41}r_1 & P_{42}r_2 & P_{43}r_3 & \cdots & P_{4(n-1)}r_{n-1} & P_{4n}r_n \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ P_{n1}r_1 & P_{n2}r_2 & P_{n3}r_3 & \cdots & P_{n(n-1)}r_{n-1} & 0 \end{pmatrix} \quad (27)$$

其中, P_{ij} 为状态 i 到状态 j 的转移概率, r_i 为系统处于状态 i 时的可靠性。

表 1 基于马尔可夫模型的不同结构类型的可靠性计算公式^[56]

Table 1 Reliability calculation formulas of different structural types based on Markov model^[56]

分类	结构类型	模型分析方法	公式
DTMC_1	吸收态	组合	$R = S(1, n)r_n$
DTMC_2	吸收态	层次	$R = \prod_{i=1}^n r_i^{v_i}$
DTMC_3	吸收态	层次	$R = \prod_{i=1}^n e^{-\lambda_i V_i t_i}$
DTMC_4	吸收态	层次	$R = \prod_{i=1}^n e^{-\int_0^{t_i} \lambda_i(\tau) d\tau}$
DTMC_5	不可约	组合	
DTMC_6	不可约	层次	$R = \sum_{i=1}^n \pi_i r_i$
DTMC_7	不可约	层次	$\lambda = \sum_{i=1}^n \pi_i \lambda_i, R = e^{-\lambda t}$
CTMC_1	吸收态	组合	
CTMC_2	吸收态	层次	$R = \prod_{i=1}^n e^{-\lambda_i L_i(t)}$
CTMC_3	吸收态	层次	$R = \prod_{i=1}^n r_i^{v_i}$
CTMC_4	吸收态	层次	$R = \prod_{i=1}^n e^{-\int_0^{L_i(t)} \lambda_i(\tau) d\tau}$
CTMC_5	不可约	组合	
CTMC_6	不可约	层次	$\lambda = \sum_{i=1}^n \pi_i \lambda_i, R = e^{-\lambda t}$
CTMC_7	不可约	层次	$R = \sum_{i=1}^n \pi_i r_i$

前文给出了在运行剖面下的单系统可靠度量度的最基本的方法,对于复杂的有备用构件的系统,自主构件的系统(包括集中式系统和联邦式系统的计算)则无能为力。图5给出了有备用构件的系统。

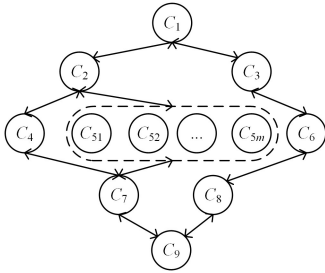


图5 有备用构件的系统

Fig.5 Construction association of system with spare components

图5中, $C_{52} \dots C_{5m}$ 是 C_{51} 的备份,针对这种情况给出可靠性计算方法。假设 k 个构件 $C_1, C_2, \dots, C_a, \dots, C_k$ 的可靠性关系为串行,其中 C_a 有 m 个备份 $C'_{1'}, \dots, C'_{m'}$,在执行任务时 $C_a, C'_{1'}, \dots, C'_{m'}$ 的可靠性关系为并行,只要其中一个构件能成功运行,该节点就不会失效。在这种情况下,进行可靠性评测时,可以将 $C_a, C'_{1'}, \dots, C'_{m'}$ 看作一个整体,并记为 $C_b = \{C_a, C'_{1'}, \dots, C'_{m'}\}$,将 $C_1, C_2, \dots, C_b, \dots, C_k$ 看成串行关系,用 r_b 表示 C_b 的可靠度, r'_i 代表 C'_i 的可靠度, r_b 的可靠度的计算式如下:

$$r_b = 1 - \prod_{i=1}^m (1 - r'_i)(1 - r_a) \quad (28)$$

然后,用传统方法来计算系统的可靠度。在去中心的微服务的网构软件的体系结构中存在大量的备用服务,在计算单个服务可靠性的基础上,再计算复杂环境下的微服务构架可靠性。

集中式结构如图6所示,系统有一个协调构件之间交互的构件,这里称之为管理构件或主控构件,其他构件称为普通构件。图6中, C_r 为主控构件, C_i 是普通构件,主控构件是其他构件之间交互的中介,任何两个普通构件之间进行通信时都要通过主控构件。

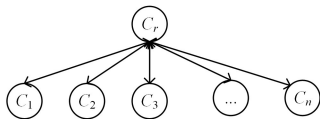


图6 集中式复合结构

Fig.6 Centralized composite structure

如果不考虑构件之间的通信可靠性,则可采用表1中的分析模型对系统进行可靠性建模,然后将主控构件的可靠性考虑进来,再用对应的公式对系统的可靠性进行求解。下文以一种分析模型为例来说明系统可靠度的评测方法。假设系统可靠性模型为DTMC_6,主控构件的可靠度为 r_c ,普通构件 C 的可靠度为 r ,在系统中被使用的概率为 π_i ,则无备用构件时系统可靠度的计算式为:

$$R = r_c \sum_{i=1}^n \pi_i r_i \quad (29)$$

由于基于自主构件的软件系统中的构件通常分布在网络的各个结点上,构件之间通过网络通信,互相协作来完成指定

的任务。因此,构件之间的通信可靠性也是影响系统可靠性的重要因素。在进行系统可靠性评估时,应充分考虑通信的可靠性。系统可用DTMC_1模型^[57]建模,并构建如式(27)所示的状态转移矩阵。式(31)为状态转移矩阵中 P_{ij} 的计算式。

$$P_{ij} = f_{ij} \times \tau_c \times r_c \times \tau_j \quad (30)$$

其中, f_{ij} 为构件 C_i 到构件 C_j 的转移概率, τ_c 为普通构件 C_i 到主控构件 C 的通信可靠度, τ_j 为主控构件 C 到普通构件 C_j 的通信可靠度。使用式(31)计算系统的可靠度^[57]。

$$R = S(1, n) \times r_n = (-1)^{n+1} \frac{|E|}{|I - M|} \times r_n \quad (31)$$

其中, r_n 为第 n 个状态时的可靠度, M 为式(27)所示的系统的状态转移矩阵, $|E|$ 为状态转移矩阵 M 中去掉第 n 行和第 1 列后的剩余矩阵的行列式, I 为 $n \times n$ 的单位矩阵。

5 仿真实验

为了使故障预测以及可靠性评估方法更加通用,本文对文献[4]中提供的日志文件的数据进行解析和预测。由于数据量巨大,为了方便比较几种数据,实验仿真中截取了20万条连续的日志数据,数据使用了公开的HDFS日志数据集,并且被相关领域专家标记为正常或者异常。经过处理后的数据随机排列后被分为训练数据和测试数据,训练数据和测试数据的比例在算法中略有调整,仿真实验采用了文献[5]提供的传统的机器学习方法(LR,PCA-Q统计方法,SVM以及集成学习模型(Boosting, Bagging 和 Stacking))对特征提取的数据分别进行了故障预测和评估,最后对系统进行了即时可靠性的度量。

5.1 实验环境

仿真实验的环境如下:处理器为Intel Xeon platinum 8260 M CPU;64核;内存为40GB;程序运行环境为Python3.7, Ubuntu18.04, Cuda 10.2, Pytorch 1.6.0。

5.2 评估方法

在机器学习、数据挖掘和推荐系统完成建模之后,需要对模型的效果进行评价。目前常采用的评价指标有准确率(Accuracy)、召回率(Recall)、F值(F-Measure)等,准确率是最常见的评价指标,准确率越高则分类器越好。在不平衡分布数据集的应用中,召回率和F值在评价方面是较能客观反映分类器的指标。

(1)混淆矩阵

- 1) True Positive(真正, TP):将正类预测为正类数。
 - 2) True Negative(真负, TN):将负类预测为负类数。
 - 3) False Positive(假正, FP):将负类预测为正类数。
 - 4) False Negative(假负, FN):将正类预测为负类数。
- 混淆矩阵如表2所列。

表2 混淆矩阵

Table 2 Confusion matrix

		预测类别		
		YES	NO	总计
实际类别	YES	TP	FN	P
	NO	FP	TN	N
	总计	P'	N'	P+N

(2) 准确率 (Accuracy)

准确率的计算式为:

$$ACC = \frac{TP + FN}{TP + TN + FP + FN} \quad (32)$$

准确率是直观的评价指标, 准确率高并不代表该算法就好。在正负样本不平衡的情况下, 准确率评价指标有很大的缺陷, 单纯靠准确率来评价算法模型不够全面。

(3) 召回率 (Recall)

召回率是覆盖面的度量, 度量有多少个正例被分为正例, 召回率的计算式如下:

$$RECALL = \frac{TP}{TP + FN} \quad (33)$$

(4) 综合评价指标 (F-Measure)

P 和 R 指标有时会出现矛盾的情况, 这样就需要综合考虑它们, 最常见的方法就是 F-Measure (又称为 F-Score), F-Measure 是 Precision 和 Recall 的加权调和平均。

$$F\text{-Measure} = \frac{(\alpha^2 + 1)P * R}{\alpha^2(P + R)} \quad (34)$$

当参数 $\alpha=1$ 时, 就是最常见的 F1, 即:

$$F1 = \frac{2 * P * R}{P + R} \quad (35)$$

可知, F1 综合了 P 和 R 的结果, 当 F1 较高时说明实验方法比较有效。

5.3 预测评估

表 3 列出了对测试集上 6 种方法的准确率、召回率和 F1

指标的评估统计结果。从 6 种方法的准确率、召回率和 F1 统计结果来看, 集成学习的预测结果在准确率、召回率和 F1 值方面相比文献[5]采用的几种方法有所提升。其中 Bagging 方法的准确率最高, Boosting 在召回率和 F1 值方面表现最好, PCA-Q 统计在预测结果上并不理想。由表 3 可知, 虽然集成学习的方法相比传统方法的召回率有所提升, 但它们的召回率并不很高, 这是由正负样本数据不平衡造成的。

表 3 600K 数据集上 6 种方法的准确性、召回率和 F1 指标的评估数据

Table 3 Metrics for 6 methods of Accuracy, Recall and F1 on 600K data set

方法	Accuracy	Recall	F1
LR ^[2]	0.8369	0.9994	0.9109
PCA-Q ^[2]	0.8336	0.1525	0.2579
SVM ^[2]	0.8366	0.9976	0.9100
Adaboosting	0.9995	0.9951	0.9973
Bagging	0.9997	0.9909	0.9954
Stacking	0.9999	0.9901	0.9950

5.4 即时可靠性

采用 4.3 节中的即时可靠性的评估方法, 对系统 t 时刻的即时可靠性进行了度量, 并通过召回率进行了修正, 修正后的即时可靠性误差有较大幅度地减小, 结果如表 4 所列。其次, 在故障预测阶段, 集成学习算法的评估效果也在系统即时可靠性的结果中得到了验证。

表 4 基于 600K 测试集的云平台实时可靠性预测

Table 4 Cloud platform real-time reliability prediction on 600K testing set

Prediction Model	Actual reliability	Prediction reliability	Recall correction	Prediction error/%	Correction error/%
LR ^[2]	0.9607	0.9826	0.9599	2.28	0.06
PCA-Q ^[2]	0.9607	0.9851	0.9591	2.54	0.14
SVM ^[2]	0.9607	0.9826	0.9524	2.28	0.83
Adaboosting	0.9607	0.9806	0.9601	2.07	0.04
Bagging	0.9607	0.9809	0.9602	2.10	0.03
Stacking	0.9607	0.9809	0.9601	2.10	0.03

5.5 系统平台提供的基础服务

丝路物联云大数据分析平台是面向企业级用户的大数据分析应用平台。该平台基于丝路物联云 PaaS 平台, 支持应用多租户化, 通过 SaaS 超市为不同用户提供在线即租即用等良好体验, 集数据接入、数据挖掘、数据可视化于一体, 有效实现了数据价值的发掘和利用。

5.6 复杂可靠性度量实例

已经实施的丝路物联云提供了一个基于 Web 界面的运维管理系统, 通过该 Web 系统, 云平台管理员以及用户可以管理不同的云平台资源以及服务。以其中的可视化展示模块为范例, 对复杂构件系统的可靠性进行度量。该平台可视化系统中包含的主要服务如图 7 所示, RC₁ 和 RC₂ 是 eureka 服务注册中心, 所有服务都在服务中心注册并被调用, Start 和 End 服务是整个可视化系统的集中入口和出口, DBO₁ 和 DBO₂ 是功能相同持久层系统, 为了提高可靠性, 提供了两个功能相同的持久层, VIZ 是可视化展示服务, ERR 是异常处理页面。

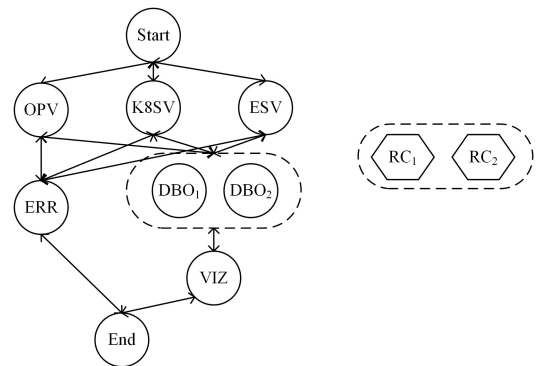


图 7 可视化展示平台微服务调用结构图

Fig. 7 Structure diagram of microservice invocation of visual display platform

对丝路物联云的平台和能源可视化展示系统的运行日志系统进行了跟踪, 选取了访问最频繁的 2021 年 4 月 16 日的日志消息, 该日志包含会话 id、时间、消息级别和日志消息, 采用第 4 节和第 5 节中的方法对各微服务的子系统的可靠性进行预测度量, 并采用召回率进行修正, 得到了系统当前时刻

较为准确的系统即时可靠性,采集的数据的可靠性度量结果如表5所列。

表5 丝路物联网的平台和能源可视化展示系统微服务及时可靠性
Table 5 Timely reliability of microservices of Silk Road IOT cloud platform and energy visualization display system

编号	模块名称	功能说明	可靠性度量	组合可靠度
1	Start	登录功能	0.9918	0.9918
2	OPV	Openstack 可视化	0.9886	0.9886
3	K8SV	K8S 可视化	0.9955	0.9955
4	ESV	能源系统可视化	0.9893	0.9893
5-1	DBO1	持久层 1	0.9885	
5-2	DBO2	持久层 2	0.9969	0.9999
6	ERR	出错处理	0.9875	0.9875
7	VIZ	可视化展示	0.9906	0.9906
8	End	程序出口	0.9964	0.9964
9-1	RC1	注册中心 1	0.9898	
9-2	RC2	注册中心 2	0.9941	0.9998

根据部署在实验室的集群管理系统的通信可靠度参数和实际通信可靠度综合度量出构件和构件之间、服务和注册中心以及注册中心到服务之间的通信可靠度,其中 f_{ij} 为构件 C_i 到构件 C_j 的转移概率, t_{ic} 为构件 C_i 到主控构件 C 的通信可靠度, t_{cj} 为主控构件 C 到构件 C_j 的通信可靠度,如表6所列。

表6 复杂构件之间的可靠度

Table 6 Reliability on complex components

	通信可靠度	说明
t_{ic}	0.9589	服务和注册中心通信可靠度
t_{cj}	0.9726	注册中心到服务之间的通信可靠度
r_c	0.9998	注册中心可靠度(见表5)

复合系统可靠度计算过程如下:

(1)首先计算服务间的转移概率 F ,如表7所列。

表7 服务间调用转移概率矩阵 F

Table 7 Transfer probability matrix F between services

	1	2	3	4	5	6	7	8
1	0	0.333	0.333	0.333	0	0	0	0
2	0	0	0	0	0.98	0.02	0	0
3	0	0	0	0	0.98	0.02	0	0
4	0	0	0	0	0.98	0.02	0	0
5	0	0	0	0	0	0	1	0
6	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0

结合表2计算 P 矩阵,其中上行服务到注册中心的通信可靠度 t_{ic} 和下行注册中心到服务的通信可靠度 t_{cr} 各自相同,因此 P 矩阵为:

$$P = F * r_c * t_{ic} * t_{cr}$$

(2)然后计算矩阵 M ,单个服务软件的系统可靠性记为 R 。

$$R = [0.997, 0.992, 0.992, 0.992, 0.993, 0.999, 0.996, 0.995]$$

记矩阵 $M = P * R$

(3)根据矩阵 M 计算系统可靠度,如算法3所示。

算法3 复合系统可靠度求解算法

输入:服务转移概率矩阵 F ,通信可靠度 t ,服务组件可靠度 R

输出:复合系统可靠度 SR

1. 求得矩阵转移概率 $P = F * r_c * t_{ic} * t_{cr}$

2. 求得状态转移矩阵 $M = P * R$

3. 根据式(31), $SR = S(1, n) * r_n = (-1)^{n+1} \frac{|E|}{|I-M|} * r_n$,求得复合系统可靠度

4. 返回复合系统可靠度 SR

(4)实验结果

采用上述数据和算法,计算出当前的系统服务可靠度为0.9667。根据实际使用情况,能较准确地反映系统的可靠度。变人工统计为自动统计预测评估较大地减轻了人工负担,同时为系统的评估和进一步的可靠性运维提供了数据基础。

结束语 本文从日志原始数据初步处理到特征提取,然后进行故障检测和复杂软件系统可靠性计算,为系统的可靠性运维提供了决策依据,主要工作如下:

(1)本文分析了日志解析、特征提取、故障检测、预测评估到即时可靠性计算的日志处理全过程,采用逻辑回归、PCA-Q 统计方法、支持向量机等传统的机器学习方法,以及集成学习的3类方法(Boosting, Bagging 和 Stacking)对处理的日志特征数据进行故障预测,提高了日志异常预测的准确率、召回率和 F1 值;

(2)在异常检测后进行系统的即时可靠性的度量,并在召回率较低的情况下对即时可靠性的度量进行了修正,较大幅度地提高了即时可靠性度量的准确度;

(3)根据个体的可靠性和基于马尔可夫系统可靠性理论来度量微服务类型的复合构件系统的可靠性,为系统的评估和进一步的可靠性运维奠定了数据基础。

本文的重点在于异常检测和即时可靠性的度量和预测,限于篇幅,基于系统日志的组合服务的异常根因分析和即时可靠性计算尚未展开,这两个问题将是未来的研究重点。

(1)软件系统具有离散型、分布性等特征,系统的组合和服务的组合存在多样性,对于组合服务故障的根本原因(Root Cause Analysis, RCA)需要进一步的分析。

(2)组合型服务复杂多样,构成系统的服务的组合可能是并行、联邦等其他的冗余组合结构,对此类系统即时可靠性的计算及评估,也是随后的研究重点。

参考文献

- [1] HUANG H, ZHANG H, SHAO D. Practical Impacts of Automation Tools in Support of DevOps in China[J]. Journal of Software, 2019, 30(10): 3056-3070.
- [2] SUN C A, JIN M Z, LIU C. Overviews on software Architecture [J]. Journal of Software, 2002, 13(7): 1228-1237.
- [3] SUN W X, ZHAI Y L, BAO T H, et al. A Microservices Oriented Edge Computing Framework for LVC Simulation in the IoT Era[C] // Proceedings of the 11th International Conference on Computer Modeling and Simulation (ICCMS 2019). Association for Computing Machinery, New York, USA, 2019: 190-195.
- [4] ZHU J, HE S, LIU J, et al. Tools and Benchmarks for Automated Log Parsing[C] // 2019 IEEE/ACM 41st International Conference on Software Engineering. Software Engineering in Practice (ICSE-SEIP). 2019: 121-130.
- [5] HE S, ZHU J, HE P, et al. Experience Report: System Log Analysis for Anomaly Detection[C] // 2016 IEEE 27th Interna-

- tional Symposium on Software Reliability Engineering(ISSRE). 2016;207-218.
- [6] VERVAET A. MoniLog: An Automated Log-Based Anomaly Detection System for Cloud Computing Infrastructures[C] // 2021 IEEE 37th International Conference on Data Engineering (ICDE). 2021;2739-2743.
- [7] MALLIKARJUN B C, ANNAPOORNESHWARI K, MADHAN Y M, et al. Intelligent Automated Text Processing System—An NLP Based Approach[C] // 2020 5th International Conference on Communication and Electronics Systems (ICES). 2020; 1026-1030.
- [8] RAND J, MIRANSKY A. On Automatic Parsing of Log Records[C] // 2021 IEEE/ACM 43rd International Conference on Software Engineering: New Ideas and Emerging Results(ICSE-NIER). 2021;41-45.
- [9] XIAO T, QUAN Z, WANG Z J, et al. LPV: A Log Parser Based on Vectorization for Offline and Online Log Parsing[C] // 2020 IEEE International Conference on Data Mining (ICDM). 2020; 1346-1351.
- [10] YADAV R B, KUMAR P S, DHAVAL S V. A Survey on Log Anomaly Detection using Deep Learning[C] // 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). 2020; 1215-1220.
- [11] HAN S B, WU Q H, ZHANG H, et al. Log-Based Anomaly Detection With Robust Feature Extraction and Online Learning [J]. IEEE Transactions on Information Forensics and Security, 2021, 16; 2300-2311.
- [12] MARLAITHONG T, BARROSO V C, PHUNCHONGHARN P. A Hyperparameter Tuning Approach for an Online Log Parser[C] // 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). 2021; 1036-1040.
- [13] WON H, KIM Y. Performance Analysis of Machine Learning Based Fault Detection for Cloud Infrastructure[C] // 2021 International Conference on Information Networking (ICOIN). 2021; 877-880.
- [14] YANG L, CHEN J J, WANG Z, et al. Semi-Supervised Log-Based Anomaly Detection via Probabilistic Label Estimation [C] // 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE). 2021; 1448-1460.
- [15] HAN Y, MA Y, WANG J, WANG J. Research on ensemble model of anomaly detection based on autoencoder[C] // 2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS). 2020; 414-417.
- [16] PROVOTAR O I, LINDER Y M, VERES M M. Unsupervised Anomaly Detection in Time Series Using LSTM-Based Autoencoders[C] // 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT). 2019; 513-517.
- [17] PU G, WANG L J, SHEN J, et al. A Hybrid Unsupervised Clustering-Based Anomaly Detection Method[J]. Tsinghua Science and Technology, 2021, 26(2); 146-153.
- [18] YILMAZ S F, KOZAT S S. Robust Anomaly Detection via Sequential Ensemble Learning[C] // 2020 28th Signal Processing and Communications Applications Conference (SIU). 2020; 1-4.
- [19] DANG Y, LIN Q, HUANG P. AIOps: Real-World Challenges and Research Innovations[C] // IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). 2019; 4-5.
- [20] HE P, ZHU J, HE S, et al. Towards Automated Log Parsing for Large-Scale Log Data Analysis[J]. IEEE Transactions on Dependable and Secure Computing, 2018, 15(6); 931-944.
- [21] ZHANG Y, TIČO P, LEONARDIS A, et al. A Survey on Neural Network Interpretability[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2021, 5(5); 726-742.
- [22] LATHA R S, SREEKANTH G R R, SUGANTHE R C, et al. A survey on the applications of Deep Neural Networks[C] // 2021 International Conference on Computer Communication and Informatics (ICCCD). 2021; 1-3.
- [23] RINCY T N, GUPTA R. Ensemble Learning Techniques and its Efficiency in Machine Learning: A Survey [C] // 2nd International Conference on Data, Engineering and Applications (IDEA). 2020; 1-6.
- [24] SRAVANTHI N, VENKAT M L, HARSHINI S, et al. An Ensemble Approach to Predict Weather Forecast using Machine Learning[C] // 2020 International Conference on Smart Electronics and Communication (ICOSEC). 2020; 436-440.
- [25] WANG K, LIU X, ZHAO J, et al. Application Research of Ensemble Learning Frameworks[C] // 2020 Chinese Automation Congress (CAC). 2020; 5767-5772.
- [26] TANG X, ASTLE Y S, FREEMAN C. Deep Anomaly Detection with Ensemble-Based Active Learning[C] // 2020 IEEE International Conference on Big Data (Big Data). 2020; 1663-1670.
- [27] BECKER S, SCHMIDT F A, GULENKO, et al. Towards AIOps in Edge Computing Environments[C] // 2020 IEEE International Conference on Big Data (Big Data). 2020; 3470-3475.
- [28] ANNA L, SHELLY G, ELLIOT K K. AIOps for a Cloud Object Storage Service[C] // 2019 IEEE International Congress on Big Data (Big Data Congress). 2019; 165-169.
- [29] VAARANDI R. A data clustering algorithm for mining patterns from event logs[C] // Proceedings of the 3rd IEEE Workshop on IP Operations & Management (IPOM 2003) (IEEE Cat. No. 03EX764). 2003; 119-126.
- [30] VAARANDI R, PIHEL GAS M. LogCluster—A data clustering and pattern mining algorithm for event logs[C] // 2015 11th International Conference on Network and Service Management (CNSM). 2015; 1-7.
- [31] ADETOKUNBO A O, MAKANJU A. Nur Zincir-Heywood, and Evangelos E. Miliotis. Clustering event logs using iterative partitioning[C] // Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09). Association for Computing Machinery, New York, USA, 2009; 1255-1264.
- [32] FU Q, LOU J, WANG Y, et al. Execution Anomaly Detection in Distributed Systems through Unstructured Log Analysis[C] // 2009 Ninth IEEE International Conference on Data Mining. 2009; 149-158.
- [33] DU M, LI F. Spell: Streaming Parsing of System Event Logs

- [C]//2016 IEEE 16th International Conference on Data Mining (ICDM). 2016:859-864.
- [34] HE P, ZHU J, ZHENG Z, et al. Drain: An Online Log Parsing Approach with Fixed Depth Tree[C]//2017 IEEE International Conference on Web Services(ICWS). 2017:33-40.
- [35] TIMČENKO V, GAJIN S. Ensemble classifiers for supervised anomaly based network intrusion detection[C]//2017 13th IEEE International Conference on Intelligent Computer Communication and Processing(ICCP). 2017:13-19.
- [36] DU M, LI F F, ZHENG G N, et al. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security(CCS '17). Association for Computing Machinery, New York, USA, 2017:1285-1298.
- [37] LIU D P, ZHAO Y J, XU H W, et al. Opprentice: Towards Practical and Automatic Anomaly Detection Through Machine Learning[C]//Proceedings of the 2015 Internet Measurement Conference(IMC '15). Association for Computing Machinery, New York, USA, 2015:211-224.
- [38] NEDELKOSKI S, CARDOSO J, KAO O. Anomaly Detection from System Tracing Data Using Multimodal Deep Learning [C]//2019 IEEE 12th International Conference on Cloud Computing(CLOUD). 2019:179-186.
- [39] DROMARD J, ROUDIÈRE GOWEZARSKI P. Online and Scalable Unsupervised Network Anomaly Detection Method[J]. IEEE Transactions on Network and Service Management, 2017, 14(1):34-47.
- [40] YOO T H. The Infinite NHPP Software Reliability Model based on Monotonic Intensity Function [J/OL]. https://www.researchgate.net/publication/282984295_The_Infinite_NHPP_Software_Reliability_Model_based_on_Monotonic_Intensity_Function.
- [41] JELINSKI Z, MORANDA P. Software Reliability Research [C]//Statistical Computer Performance Evaluation. 1972:465-484.
- [42] YI J, LUO X, AO J X, et al. Software fault classification prediction model based on Markov chain[J]. Journal of University of Chinese Academy of Sciences, 2013, 30(4):562-567.
- [43] ZHANG H, ZHANG X. Data Mining Static Code Attributes to Learn Defect Predictors[J]. IEEE Transactions on Software Engineering, 2007, 33(9):635-637.
- [44] MUHAMMED M Ö, UNAL C, AHMET Z. A novel defect prediction method for web pages using k-means++[J]. Expert Systems with Applications, 2015, 42(19):6496-6506.
- [45] JIN C, JIN S W. Software reliability prediction model based on support vector regression with improved estimation of distribution algorithms[J]. Applied Soft Computing, 2014, 15:113-120.
- [46] OKAMURA H, DOHI T. A Novel Framework of Software Reliability Evaluation with Software Reliability Growth Models and Software Metrics[C]//IEEE 15th International Symposium on High-Assurance Systems Engineering. 2014:97-104.
- [47] HARIKESH B Y, DILIP K Y. A fuzzy logic based approach for phase-wise software defects prediction using software metrics [J]. Information and Software Technology, 2015, 63(C):44-57.
- [48] SELIYA N, KHOSHGOFTAAR T M. Software Quality Analysis of Unlabeled Program Modules With Semisupervised Clustering[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2007, 37(2):201-211.
- [49] SELIYA N, KHOSHGOFTAAR T M. Software quality estimation with limited fault data: a semi-supervised learning perspective[J]. Software Quality Journal, 2007, 15(3):327-344.
- [50] XU W, HUANG L, FOX A, et al. Detecting large-scale system problems by mining console logs[C]//Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles (SOSP '09). Association for Computing Machinery, New York, USA, 2009:117-132.
- [51] KUBAT P. Assessing reliability of modular software[J]. Operations Research Letters, 1989, 8(1):35-41.
- [52] GOKHALE S S, TRIVEDI K S. Analytical Models for Architecture-Based Software Reliability Prediction: A Unification Framework[J]. IEEE Transactions on Reliability, 2006, 55(4):578-590.
- [53] LI B X, SU Z Y, ZHOU Y, et al. A user-oriented Web service reliability model[C]//IEEE International Conference on Systems, Man and Cybernetics. 2008:3612-3617.
- [54] WEI Y, SHEN X. Heterogeneous Architecture-Based Software Reliability Estimation: Case Study [C] // Third International Conference on Convergence and Hybrid Information Technology. 2008:286-290.
- [55] GOKHALE S S, MICHAELLI R S. A simulation approach to structure-based software reliability analysis[J]. IEEE Transactions on Software Engineering, 2005, 31(8):643-656.
- [56] GUO Y. Research on Reliability Evaluation for Component-based Software System[D]. Harbin: Harbin Institute of Technology, 2013.
- [57] WANG W L, WU Y, CHEN M H. An architecture-based software reliability model[C]//Proceedings 1999 Pacific Rim International Symposium on Dependable Computing. 1999:143-150.



WANG Bo, born in 1976, Ph.D, lecturer, is a member of China Computer Federation. His main research interests include system reliability, software engineering and human-computer interaction.



HUA Qing-yi, born in 1956, Ph.D, professor. His main research interests include human-computer interaction, recommender systems, and user interface engineering.