文章编号:1674-2974(2022)04-0089-11 DOI:10.16339/j.cnki.hdxbzkb.2022272

面向不稳定日志的一致性异常检测方法

刘春波[1],梁孟孟[2],侯晶雯[2],顾兆军[1],王志[3†]
(1. 中国民航大学 信息安全测评中心,天津 300300;
2. 中国民航大学 计算机科学与技术学院,天津 300300;
3. 南开大学 网络空间安全学院,天津 300350)

摘 要:系统日志被用作系统异常检测的主要数据源.现有的日志异常检测方法主要利用从历史日志中提取的日志事件数据构建检测模型,即假设日志数据随时间的推移其分布规律具有稳定性.然而,在实践中,日志数据往往包含以前未出现过的事件或序列.这种不稳定性有两种来源:1)日志发生了概念漂移;2)日志处理过程中引入了噪声.为缓解日志中出现的不稳定问题,设计了基于置信度协同多种算法的异常检测模型EBCAD(Ensemble-Based Conformal Anomaly Detection).首先,用统计量p值度量日志之间的不一致性,选择多个合适的集成算法作为不一致性度量函数计算不一致性得分进行协同检测;然后,设计了基于置信度的更新机制来缓解日志不稳定问题,将新日志的不一致性得分添加到已有得分集,更新日志异常检测的经验;最后,根据协同检测得到的置信度与预设置信水平大小来判断不稳定日志是否异常.实验结果表明,在HDFS日志数据集中,当不稳定数据注入率从5%增加到20%时,EBCAD模型的F_1值仅从0.996降低到0.985;在BGL_100K日志数据集中,当不稳定数据注入率从5%增加到20%时,EBCAD的F_1值仅从0.71降低到0.613.证明EBCAD在不稳定日志中可以有效检测到异常.

关键词:异常检测;日志分析;不稳定日志;置信度;不一致性度量;更新
中图分类号:TP391 **文献标志码**:A

Conformal Anomaly Detection Method for Unstable Logs

LIU Chunbo[1], LIANG Mengmeng[2], HOU Jingwen[2], GU Zhaojun[1], WANG Zhi[3†]
(1. Information Security Evaluation Center, Civil Aviation University of China, Tianjin 300300, China;
2. College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China;
3. College of Cyber Science, Nankai University, Tianjin 300350, China)

Abstract: System logs are used as the primary data source for system anomaly detection. Existing log anomaly detection methods mainly use log event data extracted from historical logs to build detection models, that is, the distribution of log data is assumed to be stable over time. However, in practice, log data often contains events or se-

* **收稿日期**:2021-10-01
基金项目:国家自然科学基金资助项目(61872202, 61601467), National Natural Science Foundation of China(61872202, 61601467);民航安全能力建设项目(PESA2019073, PESA2019074, PESA2020100), Civil Aviation Safety Capacity Building Foundation of China(PESA2019073, PESA2019074, PESA2020100);中国科学院重点部署项目(KFZD-SW-440), Key Research Program of the Chinese Academy of Sciences(KFZD-SW-440);天津市自然科学基金项目(19JCYBJC15500), Natural Science Foundation of Tianjin(19JCYBJC15500)
作者简介:刘春波(1976—),男,天津人,中国民航大学讲师
† 通信联系人,E-mail: zwang@nankai.edu.cn

quences that have not occurred before. The instability comes from two sources: 1) conceptual drift occurs in logs; 2) noise is introduced during log processing. In order to alleviate the problem of instability in logs, an anomaly detection model called Ensemble-Based Conformal Anomaly Detection (EBCAD) based on confidence degree and multiple algorithms is designed. Firstly, the p-value statistics are used to measure the non-conformity between logs, and multiple appropriate ensemble algorithms are selected as the non-conformity measure functions to calculate the non-conformal scores for collaborative detection. Then, an update mechanism based on confidence is designed to alleviate the problem of log instability. By adding scores of new logs into existing sets, the experiences of log anomaly detection are updated. Finally, according to the confidence degree and the preset significance level obtained by collaborative detection, the unstable log is judged to be abnormal. The experimental results show that when the unstable data injection rate increases from 5% to 20% in HDFS log data set, the F_1-score of EBCAD model only decreases from 0.996 to 0.985. In the BGL_100K log data set, when the unstable data injection rate increases from 5% to 20%, the F_1-score of EBCAD decreases only from 0.71 to 0.613. This proves that EBCAD can effectively detect anomalies in unstable logs.

Key words: anomaly detection; log analysis; unstable log; confidence; non-conformity measure; update

日志数据详细记录了系统的所有状态和行为,特别是在大规模分布式系统和工业物联网(IIoT)中.管理员可以从大规模日志数据中发现和识别系统异常[1],跟踪系统行为[2],防御恶意攻击[3].基于日志的异常检测[4-6]将帮助管理员快速定位和解决事故问题,构建安全可信的系统.但传统方法往往使用关键字搜索或正则表达式匹配,这往往需要领域知识,人工检查效率低.因此,高效、实时的基于日志的异常检测算法具有一定的理论和实际应用价值.

但是,概念漂移或模型老化[7]往往会出现在动态变化和不稳定的环境中.有些系统需要7×24小时运行来支持广泛的智能应用程序和在线服务.各种各样的运行程序将生成比以前更复杂和可变的日志.如果以往日志训练的模型、超平面或分类器等概念不随日志数据的分布而变化,这些算法就不能正确识别异常事件.因此,基于日志的异常检测算法面临着不稳定问题,表现为在系统剧烈变化环境下的精度降低现象.根据Zhang等人[8]的分析结果,日志数据不稳定的原因主要来自两个方面:1)日志语句受软件升级和外部环境影响而自然发生的变化,即概念漂移;2)日志数据在采集和解析过程中所引入的噪声.

尽管许多算法在实际环境中取得了良好的性能,如支持向量机[9](Support Vector Machine,SVM)、朴素贝叶斯[10](Naive Bayes,NB)、决策树[11](Decision Tree,DT)、随机森林[12](Random Forest,RF)等.但在处理不稳定日志时仍面临许多挑战,为应对不稳定问题,它们经常使用周期性的再训练,但这往往需要领域知识且效率低下,并不能解决问题.这些算法以粗粒度检测异常,而不考虑同类型日志数据的差异或日志数据之间的不一致性;另外日志训练出来的模型、超平面、分类器等概念不随日志数据的分布而变化,缺乏动态更新以前经验的机制.

为减少日志不稳定问题对基于日志的异常检测算法的干扰,本文设计了一个基于置信度的动态日志多算法协同异常检测模型EBCAD(Ensemble-Based Conformal Anomaly Detection).它使用多种学习算法进行协同检测.首先,设计了集成多种算法的不一致性度量模块,计算统计值p值;其次,构建了基于置信水平的预测集,通过动态调整置信水平来标记一个可信标签;然后,将包含标签、不一致性得分和待检测日志置信度的结果反馈到包含相应标签的得分集中,作为已有的经验来计算后续检测中的p值;最后,根据协同检测得到的置信度与预设显著性水平大小来判断不稳定日志的新样本是否异常.

本文的结构如下:首先介绍了日志中的不稳定问题,其次对模型EBCAD进行了详细介绍,然后展示了实验结果并对其进行了分析,最后对论文进行了总结.

1　问题描述

1.1　日志不稳定问题

1.1.1　日志的概念漂移

随着软件升级和外部环境变化,一些旧的日志事件将会发生部分变化或者不复存在.图 1 给出了微软在线服务系统日志事件的变化[8].

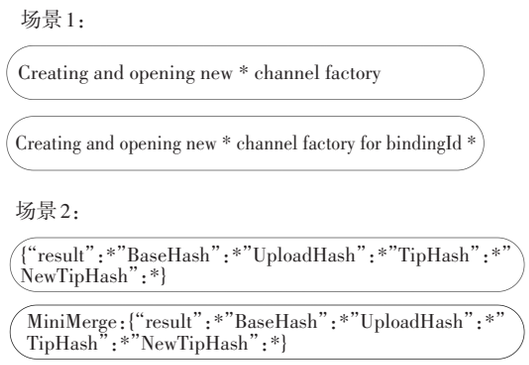


图 1　微软在线服务系统日志事件的变化[8]

Fig.1　Evolving log events in Microsoft online service system[8]

场景 1:原始日志事件后面增加“for bindingId”作为补充说明.

场景 2:原始日志事件前面添加关键字“MiniMerge”,说明程序执行阶段.

1.1.2　日志处理过程中产生的噪声

在日志数据的采集过程中,不可避免地会引入一定程度的噪声,例如由于网络错误、系统吞吐量有限、存储不足等问题导致日志丢失、重复或混乱现象发生而引入噪声.另一种重要的噪声来源是日志解析.不准确的日志解析会影响异常检测模型[13]的性能.He 等人[14]、Zhu 等人[15]对 SLCT[16]、IPLoM[17]、LKE[18]、LogSig[19]等常用的日志解析器进行了评估,指出现有的日志解析器如 LogSig 存在处理某些数据集时不够准确的问题.He 等人[14]还指出解析过程中 4% 的错误甚至会导致异常检测的性能下降一个数量级,这是因为异常检测分类器对一些关键事件较为敏感,但日志解析器对这些关键事件进行了错误解析.图 2 显示了微软在线服务系统日志解析过程中出现错误的案例[8].

场景 1:与真实日志事件相比,日志解析器遗漏了关键字“resize”.

场景 2:日志解析器错误地将参数“rtc”作为日志事件的关键字.

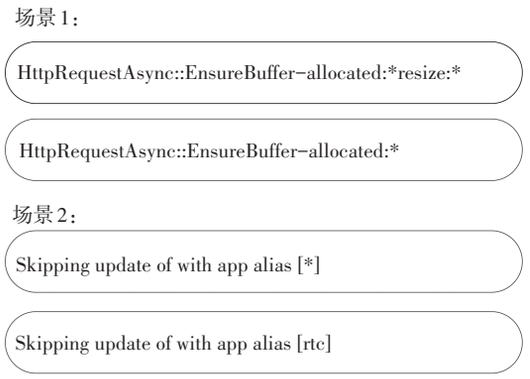


图 2　微软在线服务器 X 解析错误日志[8]

Fig.2　Mis-parsing logs in Microsoft online server X

1.2　现存算法面临挑战

动态日志随着时间的推移而变化,仅仅基于以前的检测模型很难做出准确的决策.这就要求检测模型可以根据动态日志进行动态调整.然而,以往基于日志的异常检测算法(SVM、LR、DT 等)的决策往往仅分析新日志是正常的还是异常的,而不考虑算法是如何确定选择的,忽略了新日志和已有日志的不一致性.例如支持向量机通常忽略了测试对象距离超平面的距离,仅仅根据在超平面的哪一侧做决策.统计评估则利用这个距离检测待测对象的不一致性并用来判断是否属于别的类.此外,统计评估还告诉我们,与同一类中的旧日志相比,已提交的新日志是如何属于某个类的,以及与同一事件中的其他日志相比,新日志是属于正常事件还是异常事件.然而,基于概率的传统度量,如准确率、召回率和 F_1 不能评估分类器的决策,只能表明一个新日志属于正常事件或异常事件的可能性有多大.

幸运的是,Vovk 等人[20]基于统计评估的一致性预测(Conformal Predictor, CP)理论,可以利用以前的经验来确定新的预测的精确置信水平.因此,我们将使用这种方法将新日志和历史日志联系起来,并根据数据分布的变化更新以前的经验.在一致性预测[21]中,我们可以得到一个统计量 p 值,它可以用来计算置信度,并指导算法进行决策或评估.它还可以输出具有可信度和置信度的二分类预测,或给出固定置信度下的预测集.通过将之前的数据引入决策中,可以有效减少数据分布变化引起的数据集不稳定问题.

2　模型框架

在本节中,我们将详细描述 EBCAD 的每个步骤.如图 3 所示,EBCAD 主要包括三个步骤:数据预

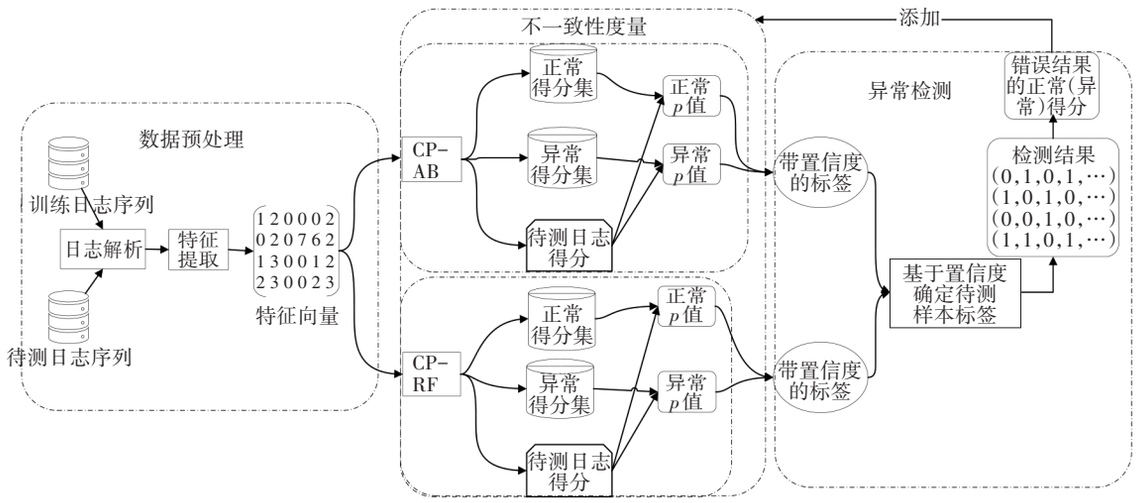


图3 EBCAD模型组成

Fig.3 Composition of EBCAD model

处理、不一致性度量、异常检测.我们首先将非结构化原始日志预处理为结构化特征矩阵.在置信度的指导下,我们选择多个集成算法来组成不一致性度量模块,这里选择的集成算法分别是AdaBoost和RF,分别记为CP-AB和CP-RF.最后,我们调整显著性水平,将检测到的日志序列标记为一个标签同时附加置信度,并反馈给相应的类作为后续检测的经验.

2.1 数据预处理

日志数据预处理的详细过程见图4.

2.1.1 日志数据

现代大型系统通过日志来记录系统的运行情况,每条日志包含无结构化的数据,例如时间戳、日志优先级、系统组件和日志序列.通常,日志消息用一组字段记录特定的系统事件.从Amazon EC2平台的日志中提取了8行日志(图4),为了便于表示,这里省略了一些字段.

2.1.2 日志解析

日志是无结构的,包含自由文本.日志解析的目的就是提取一组事件模板,从而可以结构化原始日志.每一条日志消息被解析为一个事件模板(常量部分)和一些具体的参数(变量部分),正如图4所展示的那样,第一条日志消息被解析为事件模板"Event 1":PacketResponder * for blocking * terminating.对于自动化日志解析器的有效性,可以用解析准确率[22]来评估,即成功解析的日志事件在全部日志消息中所占比例.解析后,每个日志消息都有一个事件模板,该事件模板对应于同一模板的一组消息.当且仅当日志消息的事件模板与真实数据对应的日志消息

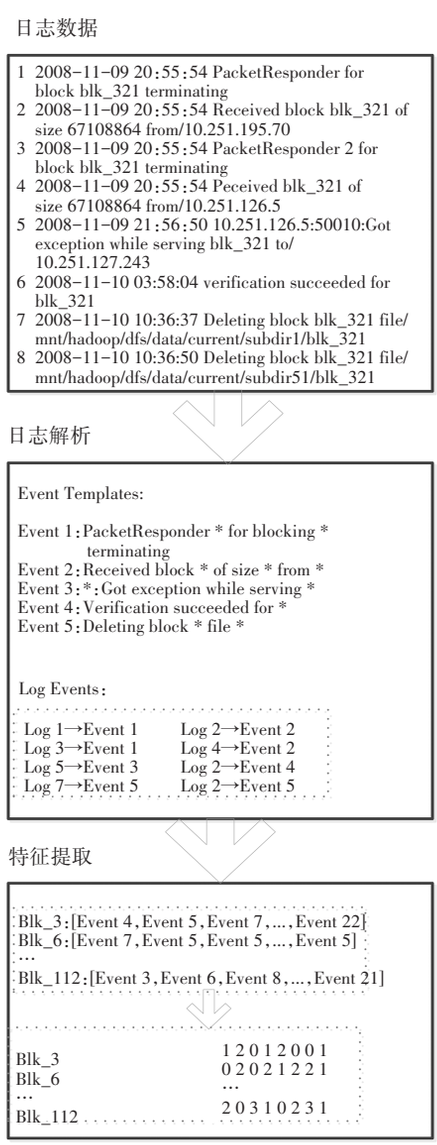


图4 日志数据预处理

Fig.4 Log data preprocessing

组相同时,日志消息被认为正确解析.例如,一个日志序列是[E1,E2,E3],如果经过日志解析后变为[E1,E3,E2],那么解析正确率即为1/3.经过比较,本文采用Drain作为日志解析器,因为在13种常见的日志解析器中,Drain的解析准确率最高[15].尽管Drain也会引入不同程度的噪声,但是本文提出的EBCAD模型可以缓解因解析器不够准确带来的日志不稳定问题.

2.1.3　特征向量提取

这个步骤的主要目的是从日志事件中提取有价值的特征,这些特征可以提供给异常检测模型.特征提取的输入为日志解析步骤中生成的日志事件,输出为事件计数矩阵.在每个日志序列中,将每个日志事件的发生次数计数到日志解析器中,以形成事件计数向量.例如,如果事件计数向量为[0,0,2,3,0,0,0],则意味着在这个日志序列中,事件3发生了两次,事件4发生了三次.最后,构造全部的事件计数向量构成事件计数矩阵$\boldsymbol{A}$,其中序列A_{ij}记录了事件j在第i个日志序列中发生的次数.

2.2　一致性预测

2.2.1　不一致性度量

不一致性度量模块是一个接口和算法独立的模块,在显著性水平的指导下,使用适当的算法作为不一致性度量,计算不一致性分数.通过一致性预测,我们将得到一个统计量p值.与概率不同的是,p值是基于具有相同标签的所有分数计算的.p值越大,则新日志与具有相同标签的旧日志的一致性越高.许多先前得分较低的日志将证明新的日志更接近这个类.因此,我们可以检测新的日志消息是否异常,并获得一个有置信度的标签.

不一致性度量模块作为EBCAD的核心,可以利用以往的经验来确定新的预测的精确置信水平,并考虑以往数据与检测数据之间的关系,而不考虑算法的内部细节和实现.我们首先选择多种算法的评分函数作为不一致性度量来计算不一致性得分.评分函数记为:

$$\alpha_{l^*} = A_D\left(L,\ l^*\right) \tag{1}$$

式中:l^*代表新的日志消息,D代表数据集中的训练集,L代表具有相同标签的一组日志.

我们使用公式(1)来计算所有日志的不一致性得分.对于日志训练数据集,每种算法将得到两个不一致性得分集:正常得分集和异常得分集.对于日志检测序列,也将通过每种算法获得一个检测日志得分.然后,根据公式(2)和公式(3)计算p值.将训练序列集合记为K,日志解析算法产生的日志事件模板集合记为T,新日志的p值是K中至少与T中其他日志不相同的日志的比例.新日志的p值的计算公式记为:

$$\forall i \in K, \alpha_i = A_D\left(T \setminus l_i,\ l_i\right) \tag{2}$$

$$p_{l^*}^T = \frac{\#\left\{i = 1, \cdots, n | \alpha_i >= \alpha_{l^*}\right\}}{|K|} \tag{3}$$

为了减少算法的分类错误,选择多个合适的算法组成这个模块,根据真实的标签将训练日志事件分为正常日志事件和异常日志事件,然后我们得到两个得分集和两个p值.最后,为缓解日志中出现的不稳定问题,我们将检测错误的测试日志序列的p值对应的得分作为后续检验的已有经验,将其添加到对应的正常得分集或异常得分集中去,作为反馈机制.这将避免算法在不稳定环境中持续造成错误决策.

2.2.2　点预测的置信度和可信度

一致性预测框架提供了两个关键的指标:置信度和可信度.如上所述,测试对象x_n尝试每个可能的标签$y_c \in y$(y={正常,异常})作为x_n的标签,然后计算p值.最后选择具有最大的p值对应标签作为检测对象x_n的标签.

检测标签将由置信度(confidence)和可信度(credibility)来衡量.置信度定义为1减去第二大的p值,而可信度定义为最大的p值.直观地说,高置信度表明,检测标签的所有其他候选者都是不可能的.低可信度意味着最大的p值很小,所以测试实例不是训练集产生的分布.注意,如果数据集是独立同分布的,可信度不会很低.

四种可能的结果如下:1)高可信度-高置信度.这是最好的情况,该算法能够正确识别一个样本对应一个类,并且只对应一个类.2)高可信度-低置信度.测试样本很相似,属于两个或两个以上的类.3)低可信度-高置信度.算法不能准确将测试样本与数据集现存类别的任何一个类别联系起来.4)低可信度-低置信度.算法给测试样本检测一个标签,但它似乎与另一个标签更相似.

2.3　异常检测

我们将使用p值来计算可信度,即对应标签的p值,置信度就是1减去第二大的p值,伪代码如算法1所示.

算法1

输入:训练集$\left\{\left(x_1, y_1\right), \cdots, \left(x_{n-1}, y_{n-1}\right)\right\}$,测试实

例x_n和一个给定的显著性水平ε.

输出:测试实例x_n的标签以及置信度.

1)用训练集$\{(x_1,y_1),\cdots,(x_{n-1},y_{n-1})\}$训练集成分类器AdaBoost和随机森林.

2)分别计算训练集中正常样本与异常样本的$\{(x_1,y_1),\cdots,(x_l,y_l)\}$中的不一致性得分$\alpha_1,\alpha_2,\cdots,\alpha_{n-1}$.注意:$\alpha_i, i=1,\cdots,n-1$是步骤1)中集成学习后分类器的不一致性度量函数结果.

3)初始化检测集合Γ^{ε}为空.

4)对每一个测试日志序列X,分别计算正常和异常的p值,即p_{N}和p_{A}.

5)比较p_{N}和p_{A}的大小,可信度取二者中较大的,置信度为1减去二者中较小的.如果置信度大于给定置信水平,待测样本的标签即为p值较大值对应的标签,否则,待测样本的标签即为p值较小值对应的标签.

6)将待测样本的标签以及置信度添加到检测集合Γ^{ε}中.

7)根据真实标签将检测错误样本对应的不一致性得分添加到对应的不一致性得分集合中,作为后续检测经验.

8)判断待测样本序列是否检测完毕,如果没有,转到步骤4);如果完毕,则输出检测集合Γ^{ε}.

3 实验及分析

3.1 实验设计

3.1.1 数据集

HDFS数据集:HDFS数据集是一种常用的基于日志的异常检测基准[23-24].它是通过在超过200个Amazon的EC2节点上运行基于Hadoop的MapReduce作业生成的,并由Hadoop领域专家进行标记.29个日志事件共产生24 396 061条日志消息.这些日志消息根据其block_id形成不同的日志序列,其中约2.9%表示系统异常.我们从原始HDFS数据集随机收集6 000个正常日志序列和6 000个异常日志序列作为训练集.

BGL_100K数据集:BGL数据包含100 000条日志信息,由劳伦斯利弗莫尔国家实验室(LLNL)[25]的BlueGene/L超级计算机系统记录.与HDFS数据不同,BGL日志没有记录每个作业执行的标识符.因此,我们必须使用固定窗口或滑动窗口将日志切片为日志序列,然后提取相应的事件计数向量.但是窗口的数量取决于所选的窗口大小和步长.在BGL_100K数据中,有2 613条日志消息被标记为失败,如果一个日志序列中存在任何失败日志,则该日志序列被标记为异常.

合成日志数据集:为了显示EBCAD处理不稳定日志数据方法的有效性,我们基于原始HDFS和BGL_100K数据集创建了不稳定的测试数据集.我们主要模拟一种日志不稳定性,如图5所示,根据Zhang等人[8]在实证研究中获得的经验,合成不稳定的日志数据.合成日志数据可以反映真实日志的不稳定特征.

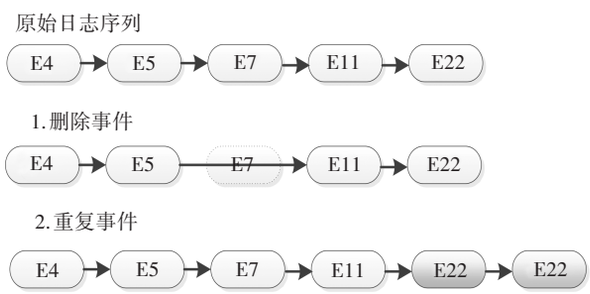


图5 合成日志序列

Fig.5 Synthetic log sequences

不稳定日志序列:在日志演变或收集过程中,日志序列可能会发生变化.为了模拟不稳定的日志序列,从原始日志序列中随机移除一些不重要的日志事件(它们不影响相应的异常状态标签).随机选择一个不重要的日志事件,并在一个日志序列中重复几次,将这些合成的不稳定日志序列按一定比例注入原始日志数据中.

为合成具有不稳定性的数据集,随机从原始HDFS数据集中收集51 000个日志序列,其中包含50 000条正常和1 000条异常,异常百分比是2%,接近原始HDFS数据集的异常比例.随机从原始BGL数据集中收集100 000个日志序列,其中包含97 387条正常和2 613条异常,异常百分比也是2%左右.将不稳定的日志序列注入其中,并创建一个测试集,表1和表2分别总结了合成的不稳定HDFS和BGL_100K数据集情况.

表1 合成不稳定HDFS数据集

Tab.1 Synthetic unstable HDFS dataset

数据集	是否稳定	正常样本数	异常样本数	总计
训练集	是	6 000	6 000	12 000
测试集	否	50 000	1 000	51 000

表 2　合成不稳定 BGL_100K 数据集

Tab.2　Synthetic unstable BGL_100K dataset

数据集	是否稳定	正常样本数	异常样本数	总计
训练集	是	58 432	1 568	60 000
测试集	否	38 955	1 045	40 000

实验在下列平台进行:Intel(R) Core(TM) i7-6700 CPU @3.40GHz 3.41 GHz, 16.0 GB RAM, Windows操作系统.

3.1.2　评估指标

实验评价采用基于混淆矩阵的二级评价指标:准确率(Accuracy)、精度(Precision)、召回率(Recall)和调和平均数(F_1)值.这些指标被用来评价算法异常检测的有效性.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{4}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

如上所示,Precision 表示所报告的异常中正确的百分比,Recall 表示所检测到的真实异常的百分比,F_1表示 Precision 和 Recall 的调和平均值.

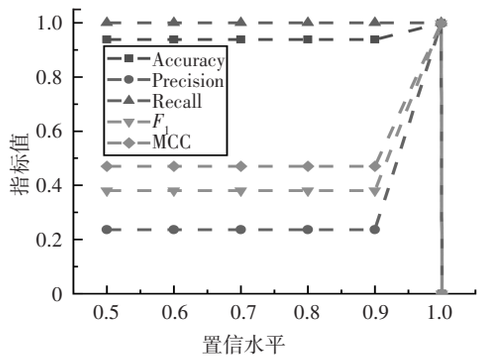
为了评估混淆矩阵中的真阳性(TP)、真阴性(TN)、假阳性(FP)和假阴性(FN)指标,采用马修斯相关系数(MCC).

$$\text{MCC} = \frac{\text{TP} \times \text{TN} + \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})} \times \sqrt{(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{8}$$

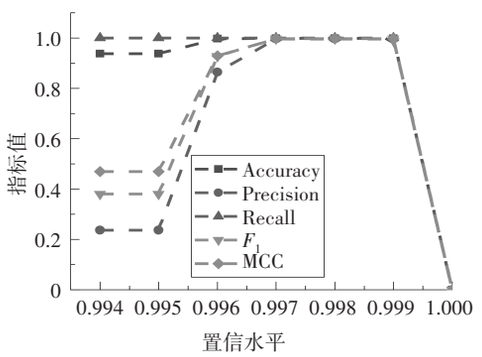
3.2　最佳置信水平的确定

本节我们探究不同置信水平下,EBCAD 各指标的性能(图 6、图 7),从而确定最优的置信水平来继续接下来的实验.在图 6(a)中,F_1、MCC 等随着置信水平的增加而缓慢提高,在置信水平为 0.995~1 之间取得最大值,图 6(b)详细描述了置信水平在 0.995~1 之间时,F_1、MCC 等指标在 HDFS 数据集上的变化过程.可以观察到,当置信水平为 0.997 的时候,F_1、MCC 等指标的值达到最大.所以,对 HDFS 日志数据集我们将置信水平设置为 0.997.

在图 7(a)中,当置信水平在 0.5~0.9 之间时,F_1、MCC 等指标值随着置信水平的增加而增加;当置信



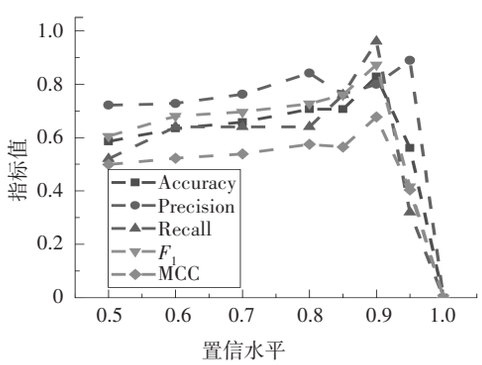
(a)HDFS数据集EBCAD各项指标的整体变化



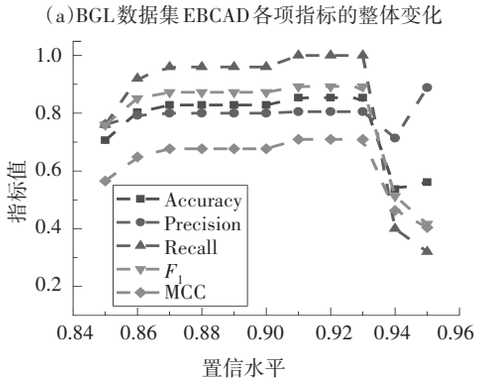
(b)HDFS数据集EBCAD各项指标的局部变化

图 6　HDFS 数据集上 EBCAD 指标性能随置信水平变化

Fig.6　EBCAD's performance on HDFS dataset varies with confidence level



(a)BGL数据集EBCAD各项指标的整体变化



(b)BGL数据集EBCAD各项指标的局部变化

图 7　BGL 数据集上 EBCAD 指标性能随置信水平变化

Fig.7　EBCAD's performance on BGL dataset varies with confidence level

水平在0.85~1之间时,F_1、MCC等的指标值达到最大.图7(b)详细描述了置信水平在0.85~1之间时,F_1、MCC等指标在BGL_100K数据集上的变化过程.可以观察到,当置信水平为0.92时,F_1、MCC等指标的值达到最大,所以,在BGL_100K数据集上,我们将置信水平设置为0.92.

3.3 不稳定日志数据集实验

3.3.1 EBCAD的有效性

我们分别在原始HDFS和BGL_100K上训练EBCAD,然后把训练好的模型在合成的测试集(注入率,即不稳定日志序列注入的比例,分别为5%、10%、15%、20%)上测试.我们将EBCAD与传统的单一决策分类器(LR、SVM、NB、CP)以及Zhang等人[8]提出的LogRobust(简写为LogR)进行了对比,实验结果如表3和表4所示.在这里CP的底层算法有四种,分别对应LR、DT、SVM、NB,我们只取达到最好效果的一种,具体达到的效果在3.3.2节中有所体现.在表3和表4中,可以看到EBCAD表现最好.即使在较高的不稳定比例20%的情况下,EBCAD依然能保持比较高的准确率.例如,当注入率为20%时,即20%的原始日志序列被替换为人工合成的日志序列,EBCAD依然能保持稳健性.原因在于,一方面,在HDFS日志数据集上,EBCAD是在置信水平为0.997下(即预测错误的概率不超过0.3%)的多模型协同检测;在BGL_100K数据集上,EBCAD是在置信水平为0.92下的多模型协同检测.一方面,相比于传统的单一决策,EBCAD的决策过程是采用多个单一决策分类器数次迭代后产生的不一致性得分,这使得度量日志不一致性的统计量p值更精确,从而使异常检测

表3 HDFS不稳定日志序列实验结果

Tab.3 Experimental results on HDFS unstable log sequences

注入率	分类器	Accuracy	Precision	Recall	F_1	MCC
5%	LR	0.999	0.94	0.997	0.967	0.967
	SVM	0.996	0.849	0.994	0.916	0.917
	NB	0.996	0.849	0.998	0.918	0.919
	LogR	0.996	0.990	0.930	0.960	0.965
	CP	0.998	0.946	1	0.972	0.972
	EBCAD	1	0.992	1	0.996	0.996
10%	LR	0.998	0.94	0.984	0.961	0.961
	SVM	0.996	0.848	0.989	0.913	0.914
	NB	0.996	0.849	0.993	0.915	0.916
	LogR	0.997	0.940	0.990	0.961	0.964
	CP	0.998	0.946	1	0.972	0.972
	EBCAD	0.999	0.996	0.992	0.994	0.994
15%	LR	0.998	0.94	0.984	0.961	0.961
	SVM	0.996	0.848	0.989	0.913	0.914
	NB	0.996	0.849	0.993	0.915	0.916
	LogR	0.998	0.980	0.910	0.940	0.960
	CP	0.998	0.945	0.992	0.968	0.968
	EBCAD	1	0.992	0.992	0.992	0.992
20%	LR	0.998	0.939	0.983	0.960	0.960
	SVM	0.993	0.846	0.987	0.911	0.909
	NB	0.995	0.854	0.979	0.912	0.910
	LogR	0.998	0.920	0.970	0.950	0.960
	CP	0.998	0.946	0.996	0.97	0.97
	EBCAD	0.999	0.975	0.996	0.985	0.985

表4 BGL_100K不稳定日志序列实验结果

Tab.4 Experimental results on BGL_100K unstable log sequences

注入率	分类器	Accuracy	Precision	Recall	F_1	MCC
5%	LR	0.463	1	0.12	0.214	0.25
	SVM	0.667	0.833	0.156	0.263	0.32
	NB	0.756	0.857	0.72	0.783	0.622
	LogR	0.488	0.833	0.2	0.323	0.328
	CP	0.756	0.857	0.72	0.783	0.622
	EBCAD	**0.854**	0.806	**1**	**0.893**	**0.71**
10%	LR	0.631	0.571	0.125	0.205	0.296
	SVM	0.667	0.833	0.156	0.263	0.32
	NB	0.655	0.714	0.156	0.758	0.677
	LogR	0.488	0.833	0.2	0.323	0.328
	CP	0.683	0.8	0.64	0.711	0.556
	EBCAD	**0.854**	0.806	**1**	**0.893**	**0.71**
15%	LR	0.631	0.571	0.125	0.205	0.296
	SVM	0.667	0.833	0.156	0.263	0.32
	NB	0.655	0.714	0.156	0.758	0.677
	LogR	0.463	1	0.12	0.214	0.225
	CP	0.756	0.857	0.72	0.783	0.622
	EBCAD	**0.829**	**0.909**	**0.8**	**0.851**	**0.709**
20%	LR	0.631	0.571	0.125	0.205	0.296
	SVM	0.655	0.714	0.156	0.256	0.321
	NB	0.69	0.568	0.781	0.658	0.56
	LogR	0.414	1	0.04	0.077	0.126
	CP	0.683	0.75	0.72	0.735	0.55
	EBCAD	**0.756**	**0.8**	**0.8**	**0.8**	**0.613**

结果更准确,也因此缓解了单一决策分类器容易出现各种错误的问题.另一方面,EBCAD能够将检测错误的不稳定日志得分添加到之前的得分集合中,更新日志异常检测的经验,从而更好地适应不稳定环境.然而,传统的单一分类器对以前未见过的日志序列无法重新学习,不能适应日志动态变化的环境,因此难以得到满意的结果.

3.3.2　集成学习的有效性

为探究集成学习对一致性异常检测的影响,图8详细记录了在不稳定比例不同的数据集中,无集成学习与有集成学习条件下一致性异常检测的结果.

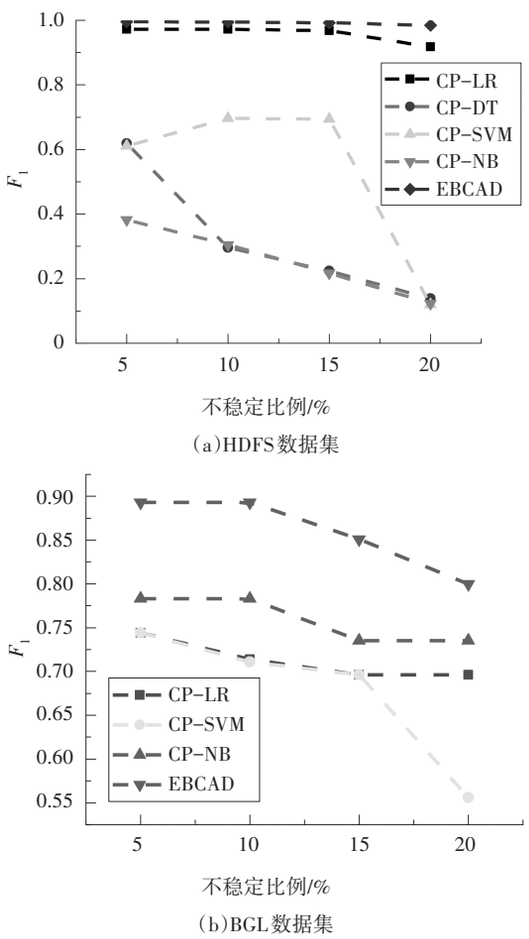


图8　不稳定比例不同的数据集中F_1的变化

Fig.8　Varing values of F_1 on datasets with different instability ratios

从图8(a)中我们看到,在不稳定比例分别为5%、10%、15%、20%的情况下,无集成学习的一致性预测中,CP-LR的F_1值最高,但EBCAD的F_1值高于CP-LR.从图8(b)中看到,所有分类器的F_1值随着不稳定比例数据的提高而呈下降趋势.在无集成学习的一致性预测中,CP-NB的F_1值最高,但EBCAD的F_1值平均高出CP-NB 10%左右,达到了比较好的效果.

3.4　稳定日志数据集实验

HDFS数据集是在从未修改的Hadoop系统[26]中收集的,BGL数据来自劳伦斯利弗莫尔国家实验室(LLNL)[25] BlueGene/L超级计算机系统记录,所以它们在源代码中没有日志序列的变化.另外,HDFS数据集和BGL_100K数据集的所有日志事件都是直接从源代码中识别出来的,可以排除解析错误等处理噪声的影响,原始的HDFS日志和BGL_100K都是稳定的数据集.我们应用EBCAD在原始的HDFS数据集和BGL_100K上,依然采用表1所示的训练集.将原始HDFS数据集中剩余的日志序列作为测试集,测试集共包含51 000条日志消息,其中1 000条表示异常行为.将原始BGL_100K数据集中剩余的日志序列作为测试集,测试集共包含40 000条日志消息,其中1 045条表示异常行为.

在HDFS数据集上的实验结果如图9所示,可以看到相比传统分类器LR、DT、SVM、NB,EBCAD的F_1和MCC值都更高.传统分类器的Recall虽然都很高,但是它们的F1和MCC值没有EBCAD高.

在BGL_100K数据集上的实验结果如图10所示,可以看到在传统分类器中表现最好的是NB.与NB相比,EBCAD的F_1和MCC值明显要高.

实验结果表明,EBCAD不仅可以有效地应用于不稳定的日志数据集,而且可以有效地应用于稳定的日志数据集.

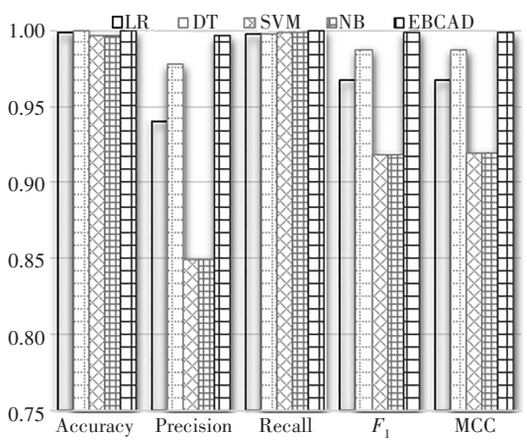


图9　HDFS稳定数据集上的结果

Fig.9　Results on stable HDFS dataset

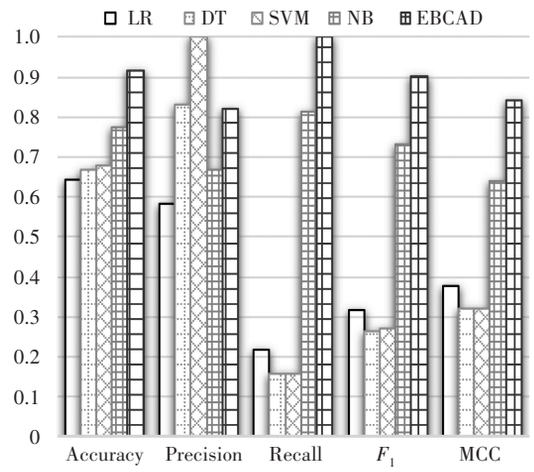


图 10 BGL稳定数据集上的结果

Fig.10 Results on stable BGL dataset

4 结 论

多年来,人们提出了许多基于日志数据的异常检测方法来自动识别大规模软件系统中的异常[26-28].然而,现有的方法无法处理日志数据的不稳定性.为应对不稳定问题,传统方法经常使用周期性的再训练方法,但这往往需要领域知识且效率低下,并不能真正解决问题.传统算法以粗粒度检测异常,而不考虑同类型日志数据的差异或日志数据之间的不一致性.为此,本文提出了一种新的应对不稳定日志的异常检测方法——EBCAD.它将新的日志和训练日志数据集联系起来,并将其作为以前的经验用于不稳定日志中的决策.EBCAD选择多个合适的算法,根据可信度共同做出决策,而不是仅依赖于单个算法做出决策.该方法在HDFS和BGL_100K两个日志数据集上得到了良好的结果,并且在准确率、召回率、F_1和MCC等指标上都取得了更好的性能,验证了EBCAD算法的有效性.

参考文献

[1] MAKANJU A, ZINCIR-HEYWOOD A N, MILIOS E E. Fast entropy based alert detection in super computer logs[C]//2010 International Conference on Dependable Systems and Networks Workshops (DSN-W). Chicago: IEEE, 2010: 52-58.

[2] OPREA A, LI Z, YEN T F, *et al*. Detection of early-stage enterprise infection by mining large-scale log data[C]//2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. Rio de Janeiro, Brazil: IEEE, 2015: 45-56.

[3] 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. New York: ACM Press, 2009: 117-132.

[4] AHMED M, NASER MAHMOOD A, HU J K. A survey of network anomaly detection techniques[J]. Journal of Network and Computer Applications, 2016, 60: 19-31.

[5] LIU C B, REN Y T, LIANG M M, *et al*. Detecting overlapping data in system logs based on ensemble learning method[J]. Wireless Communications and Mobile Computing, 2020, 2020: 1-8.

[6] 顾兆军,任怡彤,刘春波,等. 基于一致性预测算法的内网日志检测模型[J]. 信息网络安全, 2020, 20(3): 45-50.
GU Z J, REN Y T, LIU C B, *et al*. Intranet log anomaly detection model based on conformal prediction[J]. Netinfo Security, 2020, 20(3): 45-50. (In Chinese)

[7] JORDANEY R, SHARAD K, DASH S K, *et al*. Transcend: detecting concept drift in malware classification models [C]//26th {USENIX} Security Symposium. 2017: 625-642.

[8] ZHANG X, XU Y, LIN Q W, *et al*. Robust log-based anomaly detection on unstable log data[C]//Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. New York: ACM, 2019: 807-817.

[9] JOACHIMS T. Making large-scale SVM learning practical[R]. 1998.DOI: 10.1162/153244302760200704.

[10] RISH I. An empirical study of the naive Bayes classifier[C]//Proceedings of IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence. Seattle, USA: Morgan Kaufmann, 2001: 41-46.

[11] SAFAVIAN S R, LANDGREBE D. A survey of decision tree classifier methodology[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1991, 21(3): 660-674.

[12] RAZAVI B, BEHZAD R. RF microelectronics [M]. New York: Prentice Hall, 2012.

[13] XIE X S, WANG Z, XIAO X H, *et al*. A confidence-guided evaluation for log parsers inner quality[J]. Mobile Networks and Applications, 2021, 26(4): 1638-1649.

[14] HE P J, ZHU J M, HE S L, *et al*. An evaluation study on log parsing and its use in log mining[C]//2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). Toulouse, France: IEEE, 2016: 654-661.

[15] ZHU J M, HE S L, LIU J Y, *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). Montreal, QC, Canada: IEEE, 2019: 121-130.

[16] 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). Kansas City, MO, USA: IEEE, 2003: 119-126.

[17] MAKANJU A, ZINCIR-HEYWOOD A N, MILIOS E E. A lightweight algorithm for message type extraction in system application logs[J]. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(11): 1921-1936.

[18] FU Q, LOU J G, WANG Y, *et al*. Execution anomaly detection in distributed systems through unstructured log analysis [C]//2009 Ninth IEEE International Conference on Data Mining. Miami Beach, FL, USA: IEEE, 2009: 149-158.

[19] TANG L, LI T, PERNG C S. LogSig: generating system events from raw textual logs[C]//Proceedings of the 20th ACM International Conference on Information and Knowledge Management - CIKM '11. New York: ACM Press, 2011: 785-794.

[20] VOVK V, FEDOROVA V, NOURETDINOV I, *et al*. Criteria of efficiency for conformal prediction[C]//Conformal and Probabilistic Prediction with Applications. 2016: 23-39. DOI: 10.1007/978-3-319-33395-3_2.

[21] REN Y T, GU Z J, WANG Z, *et al*. System log detection model based on conformal prediction[J]. Electronics, 2020, 9(2): 232.

[22] DU M, LI F F. Spell: streaming parsing of system event logs[C]// 2016 IEEE 16th International Conference on Data Mining (ICDM). Barcelona, Spain: IEEE, 2016: 859-864.

[23] BREIER J, BRANIŠOVÁ J. Anomaly detection from log files using data mining techniques[C]//Information Science and Applications, 2015: 449-457. DOI: 10.1007/978-3-662-46578-3_53.

[24] XU W. System problem detection by mining console logs [D]. Berkeley: University of California at Berkeley, 2010.

[25] OLINER A, STEARLEY J. What supercomputers say: a study of five system logs[C]//37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07). Edinburgh, UK: IEEE, 2007: 575-584.

[26] 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. New York: ACM Press, 2009: 117-132.

[27] 贾统,李影,吴中海. 基于日志数据的分布式软件系统故障诊断综述[J]. 软件学报,2020,31(7):1997-2018.
JIA T, LI Y, WU Z H. Survey of state-of-the-art log-based failure diagnosis[J]. Journal of Software, 2020, 31(7): 1997-2018. (In Chinese)

[28] LIU C B, PAN L L, GU Z J, *et al*. Valid probabilistic anomaly detection models for system logs[J]. Wireless Communications and Mobile Computing, 2020, 2020: 1-12.