计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2020.20190875
57(4):778-790,2020

面向云数据中心多语法日志通用异常检测机制

张圣林[1] 李东闻[1] 孙永谦[1] 孟伟彬[2,3,4] 张宇哲[1] 张玉志[1] 刘 莹[3,4] 裴 丹[2,4]
[1](南开大学软件学院 天津 300350)
[2](清华大学计算机科学与技术系 北京 100084)
[3](清华大学网络科学与网络空间研究院 北京 100084)
[4](北京信息科学与技术国家研究中心 北京 100084)
(zhangsl@nankai.edu.cn)

Unified Anomaly Detection for Syntactically Diverse Logs in Cloud Datacenter

Zhang Shenglin[1], Li Dongwen[1], Sun Yongqian[1], Meng Weibin[2,3,4], Zhang Yuzhe[1], Zhang Yuzhi[1], Liu Ying[3,4], and Pei Dan[2,4]
[1](*College of Software, Nankai University, Tianjin* 300350)
[2](*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)
[3](*Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing* 100084)
[4](*Beijing National Research Center for Information Science and Technology, Beijing* 100084)

Abstract Benefit from the rapid development of natural language processing and machine learning methods, log based automatic anomaly detection is becoming increasingly popular for the software and hardware systems in cloud datacenters. Current unsupervised learning methods, requiring no labelled anomalies, still need to obtain a large number of normal logs and generally suffer from low accuracy. Although current supervised learning methods are accurate, they need much labelling efforts. This is because the syntax of different types of logs generated by different software/hardware systems varies greatly, and thus for each type of logs, supervised methods need sufficient anomaly labels to train its corresponding anomaly detection model. Meanwhile, different types of logs usually have the same or similar semantics when anomalies occur. In this paper, we propose LogMerge, which learns the semantic similarity among different types of logs and then transfers anomaly patterns across these logs. In this way, labelling efforts are reduced significantly. LogMerge employs a word embedding method to construct the vectors of words and templates, and then utilizes a clustering technique to group templates based on semantics, addressing the challenge that different types of logs are different in syntax. In addition, LogMerge combines CNN and LSTM to build an anomaly detection model, which not only effectively extracts the sequential feature of logs, but also minimizes the impact of noises in logs. We have conducted extensive experiments on publicly available datasets, which demonstrates that compared with the current supervised/unsupervised learning methods, LogMerge achieves higher accuracy. Moreover, LogMerge achieves high accuracy when there are few anomaly labels in the target type of logs, which therefore significantly reduces labelling efforts.

收稿日期:2019-12-18;**修回日期**:2020-02-10
基金项目:国家重点研发计划项目(2018YFB0204304)
This work was supported by the National Key Research and Development Plan of China (2018YFB0204304).
通信作者:孙永谦(sunyongqian@nankai.edu.cn)

Key words syslog; anomaly detection; cloud datacenter; word embedding; CNN; LSTM

摘 要 得益于自然语言处理和机器学习方法的快速发展,基于日志对云数据中心软硬件系统进行自动异常检测变得越来越流行.无监督学习方法不需要标记异常日志,但通常存在准确性较低、仍需标注大量正常日志的问题.尽管有监督学习方法的准确性较高,但由于不同软硬件系统产生不同类型的、语法各异的日志,导致有监督学习方法需要为每一类型日志标注足够多的异常日志以训练相应的异常检测模型,这极大地增加了标注异常日志的人力成本.与此同时,不同类型日志在发生异常时往往具有相同或相似的语义.因此,提出了一种跨日志类型的通用异常检测机制——LogMerge.该机制通过学习多语法日志的语义相似性,可实现日志异常模式的跨日志类型迁移,从而大大减少了异常标注开销.LogMerge 采用词嵌入方法先后构建单词和模板的向量,然后使用聚类方法将语义相同或相近的模板聚成一类,解决了不同类型日志语法不同带来的挑战.此外,LogMerge 结合 CNN 与 LSTM 方法构建异常检测模型,既有效提取了日志序列的前后依赖性,又显著降低了日志序列中噪声带来的影响.使用公开日志数据集的实验表明,相比于当前的有监督学习方法和无监督学习方法,LogMerge 取得了更高的准确性.实验还验证了 LogMerge 能够显著减少异常标注工作量——在目标类型日志异常标注较少时,依然能够取得较高的准确性.

关键词 日志;异常检测;云数据中心;词嵌入;CNN;LSTM

中图法分类号 TP391

在信息技术快速发展的背景下,云数据中心作为各行各业的关键基础设施,为我国经济转型升级提供了重要支撑[1],其稳定运行对于保证国家行政、金融、电力、电信、互联网等方面的安全与稳定至关重要[2-4].随着云数据中心所提供服务的急剧膨胀,云数据中心的规模也在快速增长.例如大型云数据中心网络往往部署数万台路由器和交换机,用于连接数十万台服务器[5].此外,云数据中心提供的服务呈多样化、复杂化发展,导致其软硬件系统日趋复杂.规模的增长和复杂性的提高导致云数据中心不可避免发生异常.这些异常不仅影响上层服务的性能,甚至会影响用户体验,产生经济损失.因此,运维人员迫切希望及时发现异常,从而快速规避并修复异常,减少异常带来的损失.

目前已有的云数据中心异常检测方案,主要基于监控指标数据,如端口流量、设备 CPU 使用率、进程内存使用率、丢包率、错误率等量化数据[6-14].然而,这种基于监控指标数据的异常检测方法并不能呈现异常发生的原因.云数据中心持续地产生日志以记录软硬件系统发生的事件,如接口状态变化、配置变更、电源关闭、板卡插入或拔出、运维人员登入登出系统、DDoS 攻击、文件读写等.通过这些日志不仅可以检测云数据中心的异常,而且可以进一步分析异常的根因,从而快速修复异常,降低甚至避免异常带来的损失.大型云数据中心每天产生数千万条日志,因此依赖运维人员人工分析海量的日志是行不通的.

基于日志的自动异常检测已广泛应用于计算机系统领域[15-23]和网络服务领域[24-28].这些方法可以分为无监督学习方法和有监督学习方法.无监督学习方法一般使用聚类[23]或长短期记忆神经网络(long short-term memory, LSTM)[15-16,28]等机器学习算法学习日志的正常模式,从而发现日志的异常行为.然而,这种方法通常存在准确性较低的问题[20],且需要从大量日志中剔除异常日志以获得正常日志.有监督学习方法一般根据异常标注学习日志的异常模式,从而达到异常检测的目的[20,26-27].这类方法通常准确性较高,但由于日志数量庞大且复杂多样,导致标注异常日志耗费大量的人力和物力资源,给运维人员带来了极大的开销.此外,不同类型软硬件系统产生的日志的语法往往是不同的.例如,图 1 展示了从 2 种厂商的交换机上采集的日志.由于不同厂商交换机打印日志的风格迥异,导致 2 种类型日志的语法存在较大差异.由于上述有监督日志异常检测方法均未考虑机器学习模型迁移的问题,导致需要为每一种类型的日志训练一个日志异常检测模型.所以,只有为每一种类型的日志标注足够多的异常日志样本,才能为这一类型日志训练一个准确的异常检测模型.这无疑极大地增加了标注异常日志产生的人力和时间成本.

Logs generated by the switch of vendor *A*: [SIF pica_sif] Interface te-1/1/11, changed state to down [SIF pica_sif] Interface te-1/1/11, changed state to up [OSPF] Neighbour (addr:X.X.X.X) on vlan20, changed state from Loading to Full [OSPF] Neighbour (addr:X.X.X.X) on vlan20, changed state from Full to Down [SIF] Vlan-interface vlan20, changed [SIF] Vlan-interface vlan20, changed state to up state to down
Logs generated by the switch of vendor *B*: %%10IFNET/3/LINK_UPDOWN(l): GigabitEthernet1/0/10 link status is DOWN. %%10IFNET/3/LINK_UPDOWN(l): GigabitEthernet1/0/10 link status is UP. %%10OSPF/3/OSPF_NBR_CHG(l): OSPF 1 Neighbor (Vlan-interface20) from Loading to Full. %%10OSPF/3/OSPF_NBR_CHG(l): OSPF 1 Neighbor (Vlan-interface20) from Full to Down. %%10IFNET/3/LINK_UPDOWN(l): Vlan-interface20 link status is DOWN. %%10IFNET/3/LINK_UPDOWN(l): Vlan-interface20 link status is UP.

Fig. 1 Examples of anomalous log sequences with different syntax but same semantics generated by two switches of different vendors

图 1 不同厂商交换机产生的语法不同、语义相同的异常日志序列举例

虽然不同类型的日志存在较大语法差异,但是其表示的语义有大量相似之处.例如尽管图 1 所示的 2 种类型日志的异常序列在语法上有较大差异,但其语义是相同的——交换机端口正在发生抖动.在分析了数百个异常日志序列之后,可以得出:不同软硬件系统在发生异常时,其产生的不同类型的日志往往存在相同或相似的语义.如果能够学习到这种"相同或相似之处",就可以把异常检测模型学习到的一种类型日志的异常模式"迁移"到另外一种类型的日志,从而避免为每一种类型的日志标记足够多的异常日志样本.通过这种方式,可以极大地降低人工标注的成本.

因此,本文提出了一种面向多语法日志的通用异常检测机制——LogMerge,基于不同类型日志的异常序列存在语义相似性的特点,结合自然语言处理方法和深度学习方法,学习多语法日志的语义相似性,实现日志异常模式的跨日志类型迁移.在获得了一种日志类型(下文称这种日志类型为源类型)的异常标注之后,LogMerge 学习源类型的异常模式并迁移到另一种日志类型(下文称这种日志类型为目标类型)的日志异常检测中,实现在缺少目标类型日志的异常标注时,保证目标类型的异常检测模型取得较高的准确性.通过这种方式,可以极大地降低人工标注异常的成本,提高运维人员的效率.

LogMerge 在实现跨日志类型的异常模式迁移过程中,遇到了 2 个主要挑战:

1) 不同类型日志语法不同.如图 2 所示,当前的有监督日志异常检测方法通常将各个日志映射到日志模板(模板定义详见 2.2 节)上,并使用模板 ID 来表征各个日志[20,26-27].但是,不同类型日志的语法往往存在较大差异,从而导致它们的日志模板以及模板 ID 是不同的.因此,如果仅仅使用日志模板 ID,无法在不同类型日志存在语法差异的前提下学习语义相似性.

Original log sequences: L1: Interface te-1/1/18, changed state to up L2: Sent xrl got response 211 Reply timed out L3: read error 104 L4: Interface te-1/1/32, changed state to down L5: Interface te-1/1/32, changed state to up L6: Interface te-1/1/32, changed state to down L7: Interface te-1/1/32, changed state to up L8: Vlan-interface vlan22, changed state to down L9: Vlan-interface vlan20, changed state to down
Log templates: T1: Sent * got response * Reply timed out T2: Interface * changed state to up T3: Interface * changed state to down T4: read error * T5: Vlan-interface *, changed state to down
Mapping logs to templates: L1→T2, L2→T1, L3→T4, L4→T3, L5→T2, L6→T3, L7→T2, L8→T5, L9→T5
Template sequences T2, T1, T4, T3, T2, T3, T2, T5, T5

T3 and T5 have the same semantics but corresponding to different templates, and only using template index cannot capture this feature.

Fig. 2 An example of mapping log sequences to template sequences used in previously proposed supervised log anomaly detection methods

图 2 前人提出的有监督日志异常检测方法将日志序列映射到模板序列举例

2) 异常日志序列中存在大量噪声.日志描述了云数据中心软硬件系统上发生的各种事件.这些事件既包括表征系统异常的事件,如受到 DDoS 攻击、电源故障、端口抖动、系统重启等,也包括表征系统正常的事件,如 PING 会话成功、运维人员登录系统、文件读写等.由于日志通常由系统的多个进程或线程生成,因此一个日志序列往往包含多个正常/异常

事件.这就导致一个异常日志序列往往夹杂一个或多个正常日志,为异常日志序列检测带来了巨大的挑战.

LogMerge 采用词嵌入(word embedding)[29]的方法,在保留日志模板语义信息的前提下,尽量避免日志模板语法的干扰.此外,LogMerge 结合一维卷积神经网络(convolutional neural network, CNN)与 LSTM,既利用了日志序列的前后关联性检测异常日志序列,又避免了噪声给异常日志序列检测带来的影响.本文的贡献可总结为 4 个方面:

1) 针对不同类型日志语法不同带来的挑战,LogMerge 创新性地使用 GloVe[29]框架构建模板中单词的词向量并进一步构建每一个模板的模板向量.模板向量不仅充分保留了日志模板的语义信息,而且有效避免了不同类型日志语法不同给模型带来的负面影响.在此基础之上,LogMerge 对模板向量进行聚类,并使用聚类中心向量表征这一类日志模板,从而将语义相近的日志模板有机融合在了一起.

2) 针对异常日志序列存在大量噪声的挑战,LogMerge 前瞻性地融合 CNN 和 LSTM 模型学习异常日志的模式,并提高异常检测的鲁棒性.CNN 根据异常标注,对聚类中心向量构成的序列不断修正,动态地调整日志序列中每个(日志所映射的聚类中心)向量的权重,从而降低甚至消除噪声日志带来的影响,提高异常检测的准确性.LSTM 模型学习异常日志序列的前后关联关系,捕获异常日志序列的序列特征.CNN 与 LSTM 的组合最大限度地降低了噪声对日志异常检测模型的干扰,提高了模型的准确性.据我们所知,这是 CNN 与 LSTM 首次组合应用于日志异常检测领域.

3) 本文使用 HDFS,PageRank,WordCount 这 3 种不同类型日志的公开数据集验证了 LogMerge 的性能.实验表明,LogMerge 的准确性(以 F_1 Score 衡量)优于前人提出的有监督日志异常检测方法和无监督日志异常检测方法.此外,LogMerge 仅仅需要目标日志类型的 20 个异常标注,即可在不同实验数据集上分别取得0.78 和 0.86 的 F_1 Score.

4)为了便于学者了解并熟悉 LogMerge,本文开源了 LogMerge 的源代码①.

1 相关工作

考虑到获取异常日志标注比较困难,学术界已经提出了一系列无监督日志异常检测方法.Lin 等人[23]尝试利用日志间相似性对系统日志进行聚类.但是,简单地按照相似性进行聚类并不能获得日志在时间维度上的前后关联性,因此无法捕获异常日志的序列性特征.Qiu 等人提出了 SyslogDigest 系统[24]理解大型互联网服务提供商(Internet service provider, ISP)中路由器发生的事件.他们首先将非结构化的日志消息映射到消息模板上以解析日志消息,然后基于消息模板的序列生成高层次的路由器事件.Du 等人[16]提出的 DeepLog 模型——一种基于 LSTM 的无监督学习框架,在离线学习阶段基于正常日志序列训练模型,在测试阶段判断新产生日志序列与模型学习到的特征是否相符,如果不相符,就认为该日志序列是异常的.但是,DeepLog 依然需要标注大量的正常日志来保证模型的准确性,且由日志模板编号表征的日志序列并不能实现跨日志类型的迁移.LogAnomaly[28]对 DeepLog 进行了改进,将模板编号映射到模板向量.然而,这种方法依然需要标注大量的正常日志,且未考虑不同类型日志语法不同的问题.总体而言,无监督日志异常检测方法普遍存在准确性较低、需要标注大量正常日志的问题,故无法实现面向多语法日志的通用异常检测.

与无监督异常日志检测方法不同,有监督异常检测方法学习异常日志的模式,并在此基础上检测新生成的日志是否符合异常日志的特征.Lu 等人[20]利用 CNN 来检测大规模系统日志中的异常,基于不同大小的卷积核提取出日志之间的内部关联(下文以 LogCNN 表示这种方法).由于深度学习模型往往需要大量的训练数据,因此该方法需要大量的异常标注数据.对于只有少量标注的日志数据集,这种方法的准确性会显著降低.Kimura 等人[26]和 Zhang 等人[27]依据专家经验从日志序列中提取典型特征,并使用传统机器学习方法(如随机森林)学习日志的异常模式,从而达到检测异常日志的目的.与 LogCNN 相同,这 2 种方法将日志模板 ID 直接输入机器学习模型,无法解决不同类型日志的语法不同带来的多异常标注问题.

考虑到前人提出的无监督异常日志检测方法和有监督异常日志检测方法均无法解决不同类型日志语法不同导致的海量异常日志标注问题,本文提出了 LogMerge 以解决这一问题.

① LogMerge 的源代码链接为 https://github.com/Logs2019/LogMerge.

2 日志与模板

2.1 系统日志

日志是云数据中心软硬件系统中生成的非结构化文本,例如可通过"printf"函数生成.如表 1 所示,一条日志通常具有固定的结构,这一结构包含了时间戳、软硬件系统 ID、消息类型和详细消息这 4 个域.其中,时间戳域表示日志生成的具体时间;软硬件系统 ID 域表示生成日志的软硬件系统的标识;消息类型域描述了日志的概要特征;详细消息域描述了日志的具体事件.一般地,消息类型域和详细消息域的语法随着软硬件系统的类型、生产厂商、型号的变化而变化,没有统一的格式,但通常由固定部分和参数部分组成.固定部分是开发人员预定义的,用于表示某类事件的信息;参数部分在软硬件系统运行过程中依据具体的时序、交互设备等信息动态生成.下文所述的"日志",如无特别说明,指的是一条日志的详细消息域.

在日常运维过程中,运维人员一般基于单条日志消息或多条日志消息组成的日志序列判断云数据中心的软硬件系统是否发生异常.一些单条异常日志消息,例如"Power PowerSupply1 failed",可表征软硬件系统发生了异常.其次,尽管一些日志消息单独出现时并不表征软硬件系统的异常,但是当其以特定的日志序列出现时则表征发生了异常.例如当"Interface te-1/1/8, changed state to up"单独出现时,并不表征软硬件系统发生异常,因为稀疏的端口抖动是正常现象.但是,如果其在短时间内频繁出现,那就表明软硬件系统发生了异常.LogMerge 既关注单条异常日志表征的异常,也关注异常日志序列表征的异常.

Table 1 Examples of Logs

表 1 日志举例

Type of Log	Timestamp	System ID	Information Type	Detailed Information
A	Apr 13 22:48:46 2018	ID *m*	SIF pica_sif	Interface te-1/1/18, changed state to up
A	Jan 5 14:54:07 2017	ID *n*	FINDER xorp_rtrmgr	Sent xrl got response 211 Reply timed out
B	Mar 18 07:40:33 2017	ID *x*	LIBCOMM pica_login	QSFP qe-1/1/52 is plugged in, vendor:WTD, serial number: RD143210010213
B	Apr 1 08:25:19 2015	ID *y*	XifCardManager	modify_l3_entry_ext failed

2.2 日志模板

由于日志消息通常是非结构化的文本,只有进行适当地解析后才能有效地用于(基于机器学习方法的)异常检测.当前,解析设备日志的普遍做法是从日志消息中提取模板.模板通常指日志消息中的固定部分,其能够概括日志所表达的事件,且相似日志消息可以用同一模板表示.如表 1 所示,"Interface * changed state to up"是"Interface te-1/1/18, changed state to up"的模板.相比于原始日志,模板中删除了变量部分——"te-1/1/18",保留了事件的主体部分,即"端口的状态变为开".这一模板不仅可以表示"Interface te-1/1/18, changed state to up"这一日志消息,还可以表示与这一日志消息阐述相同事件的其他日志消息,如"Interface te-1/1/32, changed state to up"等.

本文采用 FT-Tree[29-30] 进行模板提取.FT-tree 是一种扩展的前缀树结构,其基本思想是日志消息中的固定部分通常是频繁出现的单词的最长组合.因此,提取模板等价于从日志中识别出频繁出现单词的最长组合.大量基于生产环境日志的实验表明,FT-tree 支持增量式学习、准确性高,且具有很高的模板匹配效率.

3 LogMerge 框架

为了解决不同类型日志语法不同带来的高异常标注开销,本文提出了 LogMerge 实现跨日志类型的异常检测模型迁移.在获得了源类型的异常标注之后,LogMerge 学习源类型的异常模式并迁移到目标类型的日志异常检测中,实现当目标类型日志的异常标注不足时,保证目标类型的异常检测模型取得较高的准确性.本节将详细介绍 LogMerge 的整体框架以及各个部分的细节信息.

3.1 整体架构

图 3 展示了 LogMerge 的整体架构.LogMerge 分为 2 个部分:离线训练部分和在线检测部分.在离线训练部分,LogMerge 首先从源类型日志和目标类型日志中提取模板,并基于模板中的单词先后构建

词向量和模板向量.在此基础上,对模板向量进行聚类,并使用聚类中心向量代表这一聚类簇中的所有模板.因此,每一个日志都可以映射到一个聚类中心向量上.结合日志序列,可以得到聚类中心向量序列.LogMerge将这一向量序列、源类型日志的大量异常标记、目标类型日志的少量异常标记输入到融合了CNN与LSTM模型的深度神经网络,训练得到一个跨日志类型的通用异常检测模型.在在线检测部分,LogMerge依据上述方法将在线日志序列映射到聚类中心向量序列,并依据训练好的异常检测模型判断该在线日志序列是否异常,如果是异常日志序列,则产生告警.

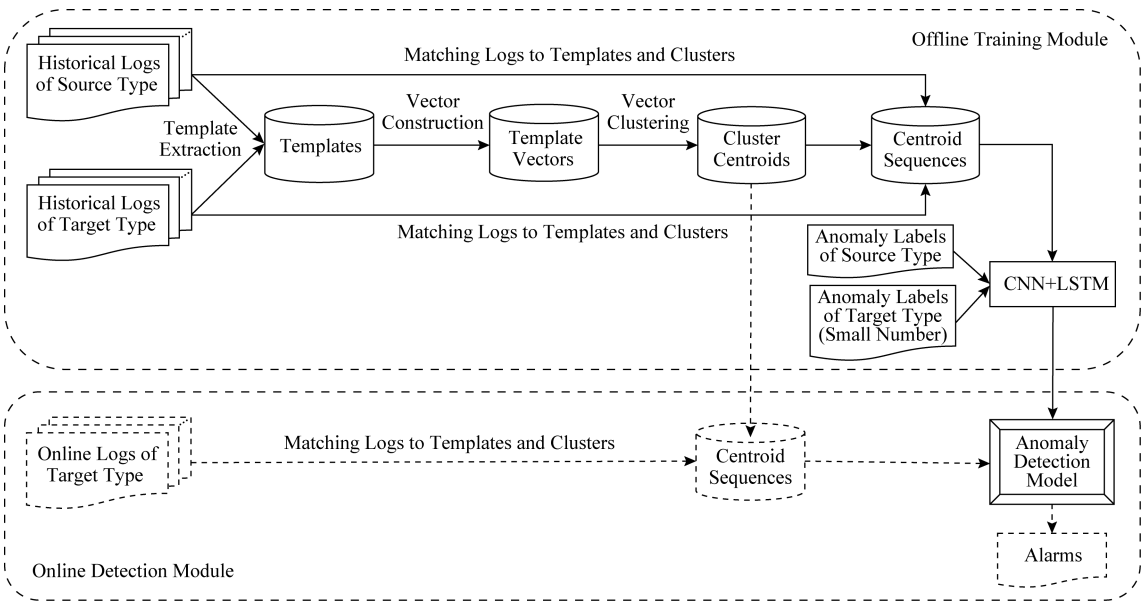


Fig. 3 Architecture of LogMerge

图3 LogMerge架构

3.2 构建模板向量并聚类

尽管不同软硬件系统产生的不同类型日志的语法存在较大差异,但是它们在发生异常时往往表现出相同或相近的语义.如2.2节所述,日志模板能够概括日志所表达的事件.为了保留日志模板的语义信息而降低甚至消除语法给跨日志类型异常检测带来的影响,LogMerge首先利用全部模板(模板中单词按照日志语序正序排列)作为训练数据获取模板单词的词向量,并在此基础上构建模板向量.

GloVe[29]利用全局矩阵分解和局部上下文窗口兼顾较长范围的全局信息和短距离的单词上下文信息,从而充分保留了模板中每个单词的语义信息.其模型目标函数为

$$J=\sum_{i,j=1}^{V}f(X_{ij})(\boldsymbol{w}_i^{\mathrm{T}}\tilde{\boldsymbol{w}}_j+b_i+\tilde{b}_j-\log X_{ij})^2,$$

其中,$\boldsymbol{w}_i$, b_i 分别表示单词 i 的词向量表示和偏置量;$\tilde{\boldsymbol{w}}_j$,$\tilde{b}_j$ 分别表示单词 j 的上下文词向量表示和上下文偏置量;X_{ij}表示单词 j 在单词 i 上下文出现的次数($\boldsymbol{X}$ 表示全局矩阵);f 是加权函数,定义为

$$f(x)=\begin{cases}(x/x_{\max})^{\alpha}, & x<x_{\max},\\ 1, & \text{otherwise.}\end{cases}$$

通过使用GloVe可以获得模板中每个单词的词向量,LogMerge将模板中单词的词向量进行加权求和,最终获得模板的模板向量,加权求和公式为

$$t_i=\frac{1}{n}\sum_{j=1}^{n}w_i^j\ (i=1,2,\cdots),$$

其中,t_i 为模板向量第 i 维的值,w_i^j 为模板中第 j 个单词第 i 维的值,n 为该模板中单词的数量.通过这种方式,模板的语义信息被保留了下来.由于模板向量与单词的顺序无关,因此模板向量排除了日志语法的影响.

为了将源类型日志的模板和目标类型日志的模板进行有机融合,LogMerge将语义相同或相似的模板进行了合并.由于模板向量反映了模板的语义信息,LogMerge基于模板向量对模板进行了合并.具体而言,LogMerge首先计算不同模板向量之间的欧氏距离;然后采用 K-means 算法[31]对模板向量进行聚类;最后,LogMerge使用每个聚类簇的中心向量来表征这一聚类簇中所有的模板向量.

为了便于读者理解,图4展示了从原始日志映射到聚类中心向量的过程.可以看到,任一日志序列都可以映射到一个聚类中心向量序列.

Original log sequences:
L1: Interface te-1/1/18, changed state to up
L2: Sent xrl got response 211 Reply timed out
L3: read error 104
L4: Interface te-1/1/32, changed state to down
L5: Interface te-1/1/32, changed state to up
L6: Interface te-1/1/32, changed state to down
L7: Interface te-1/1/32, changed state to up
L8: Vlan-interface vlan22, changed state to down
L9: Vlan-interface vlan20, changed state to down

Log template vectors:
T1: Sent xrl got response Reply timed out
[0.0066, 0.0093, ⋯, −0.0039, 0.0382]
T2: Interface te changed state to up
[−0.0377, 0.0075, ⋯, 0.0505, −0.0061]
T3: Interface te changed state to down
[−0.0406, 0.0102, ⋯, 0.0316, −0.0491]
T4: read error
[−0.0910, 0.0471, ⋯, −0.0074, 0.0084]
T5: Vlan-interface changed state to down
[−0.0396, 0.0092, ⋯, 0.0416, −0.0391]

Log cluster centroid vectors:
C1: [0.1044, −0.0854, ⋯, −0.0689, 0.0298]
C2: [0.1010, −0.0561, ⋯, −0.0414, 0.0086]
C3: [0.0001, −0.0399, ⋯, −0.0682, −0.0080]

Mapping logs to cluster centroid vectors:
L1→T2, L2→T1, L3→T4, L4→T3, L5→T2, L6→T3, L7→T2, L8→T5, L9→T5
T1→C1, T2→C2, T3→C2, T4→C3, T5→C2

Fig. 4 An example of mapping a syslog sequence to its cluster centroid vector sequence

图 4 原始日志映射到聚类中心向量举例

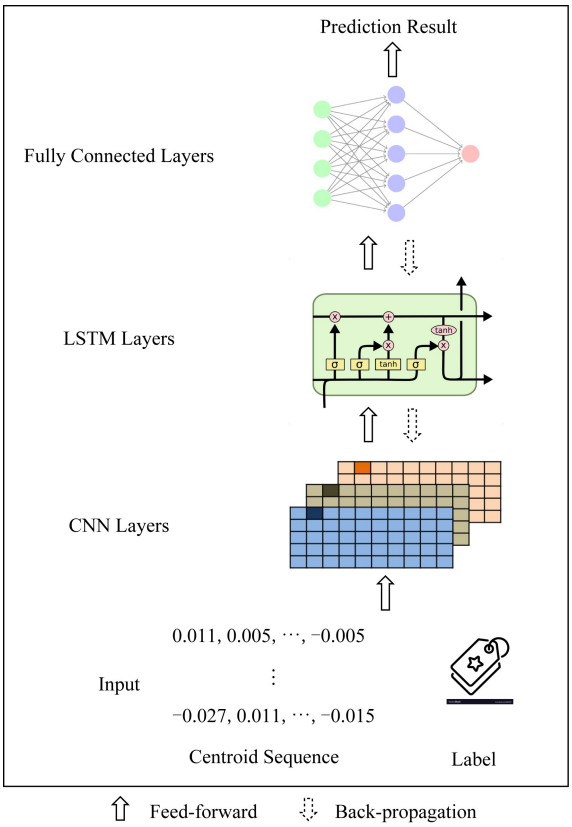


Fig. 5 Anomaly detection model's structure(the output of the lower layer is the input of the higher layer)

图 5 异常检测模型结构(下一层的输出为上一层输入)

3.3 基于 CNN 和 LSTM 的异常检测

LogMerge 结合 CNN 与 LSTM 以实现准确的异常检测.CNN 基于异常日志标记动态地调整每个聚类中心向量序列中每个日志所映射的向量的权重,从而降低日志序列中噪声的影响.LSTM 学习异常日志序列的前后关联性特征,从而判断在线日志序列是否异常.LogMerge 异常检测模块的整体架构如图 5 所示.

由于 CNN[32] 通常被用于捕获数据的局部特征信息,因此 CNN 能够在文本数据中发现短距离的词与词之间的特征.LogMerge 利用 CNN 从聚类中心向量的单维度中捕获显著的特征.由于这些特征与整体输入数据的其他特征没有高度相关性,因此 LogMerge 采用一维 CNN 模型.通过这种方式,LogMerge 获得的聚类中心向量序列被不断修正.

由于 LogMerge 针对异常日志序列进行异常检测,而日志序列间具有顺序依赖性,因此 LogMerge 采用 LSTM[33] 捕捉异常日志序列的模式.RNN(recurrent neural network)是一种单元之间直接连接的神经网络,当前单元的内部状态依赖于当前输入和前一刻状态.LSTM[33] 在 RNN 的基础上解决了长期依赖性问题,在序列更长的数据上有更好的表现.

由于卷积核的多样性,CNN 能够充分提取局部的特征,LSTM 能够充分考虑日志序列的顺序依赖特征.最后,LogMerge 采用 Dense 层将前面提取的特征经过非线性变化得到最终的异常检测模型.

LogMerge 作为一种深度学习机制,空间复杂性大于传统机器学习方法(基于 PCA 的方法[21] 和基于逻辑回归的方法),且小于基于 CNN 的模型[20];时间复杂性大于传统机器学习方法(基于 PCA 的方法[21] 和基于逻辑回归的方法),且小于基于 CNN 的模型[20].

总而言之,LogMerge 先利用 GloVe 算法获取模板的向量表示,再使用 K-means 算法对模板向量进行聚类并得到每一个聚类中心的向量表示,最后将数据输入由 CNN 和 LSTM 组合而成的异常检测模型进行模型参数学习和异常检测.LogMerge 首次结合了词嵌入方法、聚类方法、CNN 和 LSTM,有效解决了不同类型日志语法不同以及日志中存在大量噪声的问题,并有效降低了异常检测的标注开销.

4 实验及分析

本文基于公开数据集验证了 LogMerge 的性能.首先,将 LogMerge 与当前的有监督学习方法和无监督学习方法进行对比.其次,验证了 LogMerge 的词嵌入与聚类对 LogMerge 的重要性.最后,评估了目标类型日志需要标记的异常数量.

4.1 数据集介绍

如表 2 所示,为了准确、全面地评估 LogMerge,本文使用了 3 个公开数据集进行实验(所有数据均来自公开数据集,本文未对数据集进行修改):

1) WordCount 数据集[22].WordCount 是一种基于 Hadoop 架构的应用,时间跨度约 50 h.

2) PageRank 数据集[22].与 WordCount 类似,PageRank 也是一种基于 Hadoop 架构的应用,这一数据集的时间跨度也是 50 h.

3) HDFS 数据集[16].涵盖了从 200 多个 Amazon EC2 节点收集 HDFS 日志,时间跨度约 38.7 h.

由表 2 可以看出,在 3 种数据集中 WordCount 数据集所包含的异常日志序列最少.此外,WordCount 数据集所包含的异常日志序列模态各异,噪声较多,是理想的测试数据集.因此,本文将 WordCount 数据集作为目标类型日志集,并将 PageRank 和 HDFS 分别作为源类型日志集,以验证 LogMerge 的性能.

Table 2 The Detailed Information of Three Datasets

表 2 3 种数据集的统计信息

Dataset	The Number of Total Sequences	The Number of Anomalous Sequences
PageRank	70 115	40 445
WordCount	26 024	1 690
HDFS	1 873 780	59 438

4.2 构建聚类中心向量序列

LogMerge 将一个日志序列映射到聚类中心向量序列,并将聚类中心向量序列输入 CNN 和 LSTM 结合而成的深度神经网络.本文利用滑动窗口方法获得日志序列.滑动窗口方法有 2 个参数:窗口大小和步长.窗口大小表示一个日志序列中包含的日志数量,步长表示获取下一个日志序列跨过的日志条目数.图 6 以窗口大小为 3、步长为 2 为例展示了聚类中心向量序列的获取,其中实线表示获取的第 1 个日志序列所映射的聚类中心向量序列,虚线表示获取的第 2 个日志序列所映射的聚类中心向量序列.

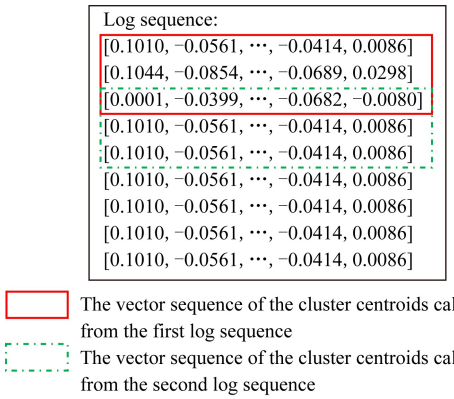


Fig. 6 Constructing cluster centroid vector sequences

图 6 构建聚类中心向量序列举例

4.3 对比方法介绍

当前学术界已经提出了一系列有监督学习方法和无监督学习方法来检测异常日志.本文分别从有监督学习方法和无监督学习方法中各选择 2 种方法(一种基于传统机器学习方法,一种基于深度学习方法),并将其与 LogMerge 进行对比.以下是本文所选取的 4 种异常检测方法的介绍:

1) 无监督学习方法

① 基于主成分分析的方法(PCA)[21].一种基于日志模板计数矩阵的异常检测方法.

② LogAnomaly[28].一种基于 LSTM 的无监督日志异常检测机制,通过学习日志的正常模式进行异常检测.

2) 有监督学习方法

① 基于逻辑回归的模型.逻辑回归是一种广泛应用于分类的统计模型,首先计算日志异常或正常的概率,然后将概率较大的状态作为日志的检测值.

② 基于 CNN 的模型[20].利用 CNN 捕获异常特征,然后基于全连接网络进行异常检测.

对于 LogMerge 和上述 4 种检测方法,本文使用 HDFS 和 PageRank 的全部数据集以及包含前 20 个异常日志序列的 WordCount 数据集作为训练集,并使用 WordCount 的剩余数据集作为测试集.本文将在 4.7 节论述这一数据划分的依据.

4.4 评价标准

异常检测可以被转化为二分类问题,其常用的评价指标为精度(*Precision*)、召回率(*Recall*)以及精度与召回率的调和平均(F_1 Score).计算公式为

$$Precision=\frac{TP}{TP+FP},$$

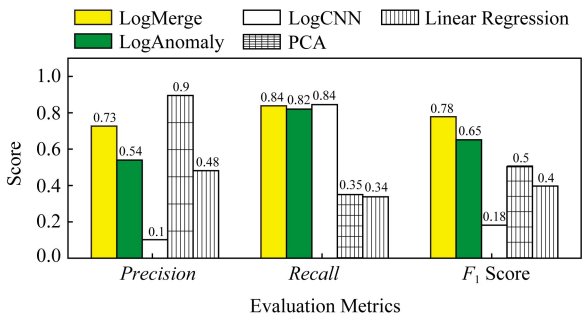
$$Recall=\frac{TP}{TP+FN},$$

$$F_1=\frac{2\times Precision\times Recall}{Precision+Recall},$$

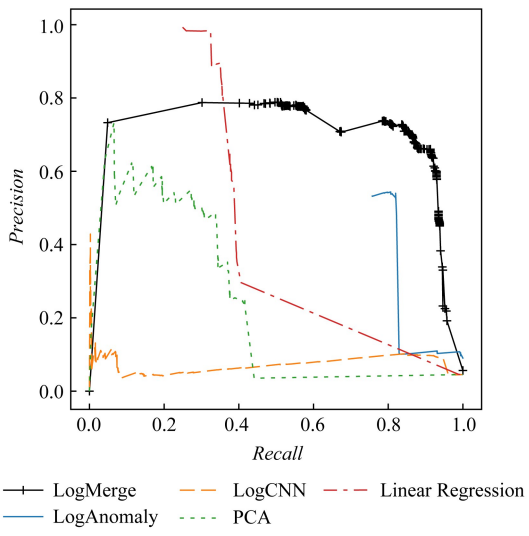
其中,TP(true positive)表示成功检测出的异常日志序列数量,FP(false positive)表示正常日志序列被异常检测模型判断为异常的数量,FN(false negative)表示异常日志序列数量被异常检测模型判断为正常的数量.

4.5 异常检测模型整体性能评价

图 7 和图 8 分别展示以 HDFS 数据集和 PageRank 数据集作为源类型日志集的实验结果.请注意,本文依据最优实验结果确定各种方法的参数.例如,对于 LogMerge 来说,以 HDFS 作为源类型日志集时聚类数量为 250,以 PageRank 作为源类型日志集时聚类数量为 170.图 7(a)和图 8(a)分别展示了在 2 组数据集上各种方法取得的最好 F_1 Score.LogMerge 在 2 组数据集上均取得了最高的准确性,其 F_1 Score 分别为 0.78 和 0.86.基于 CNN 的模型在 HDFS 数据集上也取得了 0.84 的召回率,但是其精度只有 0.1,因此产生了大量的误报,从而给运维人员带来了较大的(处理误报警)开销.基于线性回归的方法取得了比 LogMerge 更高的精度(在 HDFS 和 PageRank 数据集上分别较 LogMerge 提高了 0.17 和 0.08).但是,这种方法的召回率显著低于 LogMerge(其在 2 种数据集上的召回率均为 0.35),即大量的异常被遗漏了,这可能会给云数据中心软硬件系统的稳定性带来极大的挑战.通过调整不同方法的阈值,可以得到这些方法的 PRC(precision recall curve)曲线,如图 7(b)和图 8(b)所示.通过 PRC 曲线可以看出相比于上述 4 种方法,LogMerge 的鲁棒性最好,在大部分情况下都能取得最好的 F_1 Score.



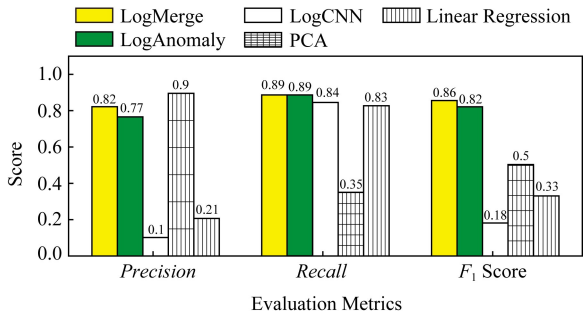
(a) Comparison of best F_1 Score



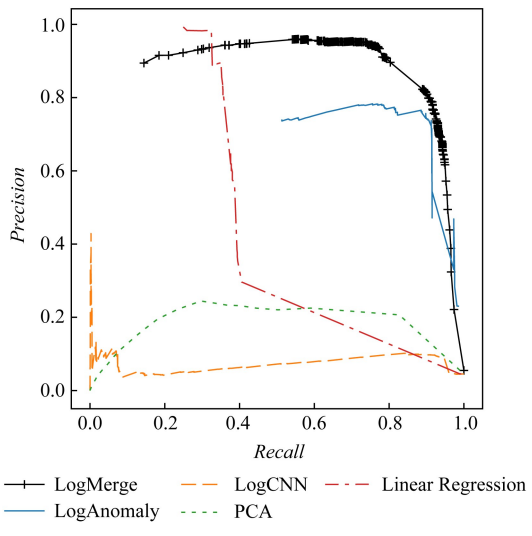
(b) Comparison of PRCs

Fig. 7 Comparison of results when HDFS is applied as the source type

图 7 以 HDFS 作为源类型日志集的对比实验结果



(a) Comparison of best F_1 Score



(b) Comparison of PRCs

Fig. 8 Comparison of results when PageRank is applied as the source type

图 8 以 PageRank 作为源类型日志集的对比实验结果

4.6 词嵌入与聚类评价

LogMerge 采用词嵌入方法先后构建单词和模板的向量,并使用聚类方法将语义相同的模板进行

合并.词嵌入与聚类的组合有效解决了不同类型日志语法不同的挑战,是 LogMerge 的核心部分.为了验证词嵌入与聚类这一组合的性能,本文尝试将 LogMerge 的这一组合去掉,并将其与 LogMerge 进行性能对比.图 9 和图 10 分别展示了使用 HDFS 数据集和 PageRank 数据集作为源类型日志集的上述 2 种方法的最优 F_1 Score 对比结果.在 HDFS 数据集上,去掉组合的 LogMerge 取得了更高的召回率,但是其精度很低,导致产生了很多误报,这对运维人员来说是不可接受的.在 PageRank 数据集上,完整的 LogMerge 比去掉组合的 LogMerge 取得了更好的精度和召回率.

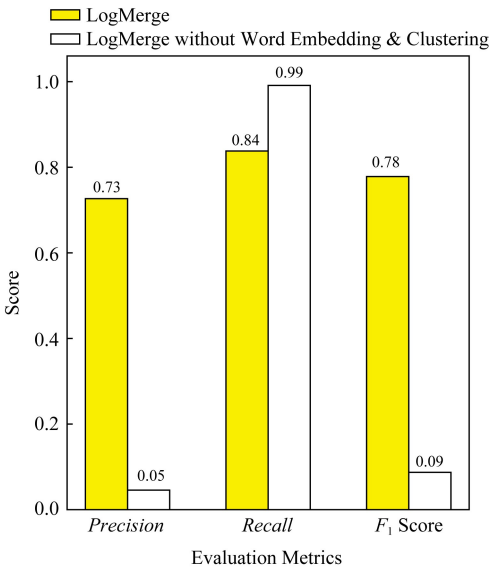


Fig. 9 Evaluation results on word embedding and clustering when HDFS is applied as the source type

图 9 以 HDFS 作为源类型日志集的词嵌入和聚类评价实验结果

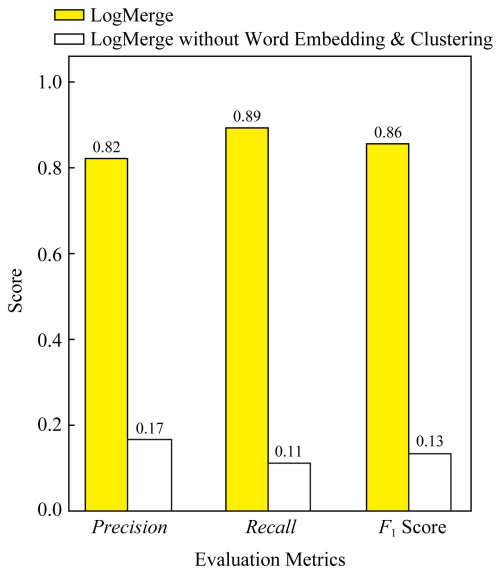


Fig. 10 Evaluation results on word embedding and clustering when PageRank is applied as the source type

图 10 以 PageRank 作为源类型日志集的词嵌入和聚类评价实验结果

4.7 目标类型日志异常标注数量评估

为了评估对目标类型日志进行异常标注数量,本文逐步增大训练集中目标类型日志的异常标注数量,并观察 LogMerge 的准确性变化,如图 11 所示.如果使用 HDFS 数据集作为源类型日志集,当训练集中目标类型日志的异常标注数量超过 20 时,

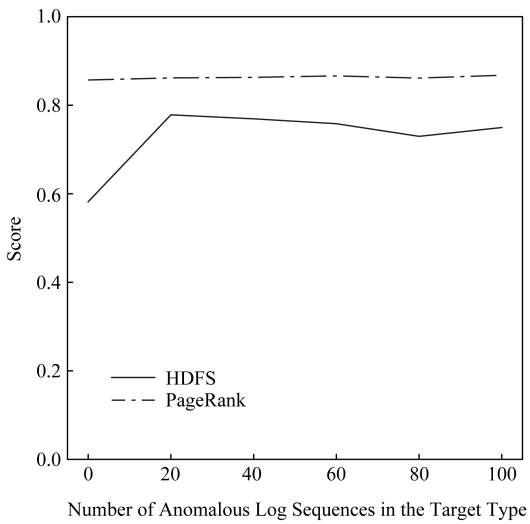


Fig. 11 Evaluation results on the number of anomalous labels in the target type

图 11 对目标类型日志异常标注数量评估的实验结果

LogMerge 的准确性趋于稳定.对于 PageRank 数据集来说,即使这一数量等于 0,LogMerge 也能取得稳定的准确性.这一实验结果表明,在获得了一个源类型日志集的异常标注后,LogMerge 仅仅需要目标类型日志集的少量异常标注,即可取得稳定的准确性,从而极大地降低了为目标类型日志集标注异常带来的人力开销.

5 总 结

日志作为一种反映系统状态和事件的数据为检测云数据中心软硬件系统异常提供了重要支撑.本文提出了一种面向多语法日志的通用异常检测机制——LogMerge,融合词嵌入方法和聚类方法保留日志的语义信息并降低语法各异给异常检测带来的影响,结合 CNN 与 LSTM 方法排除噪声的干扰并提取日志序列的前后依赖性特征.基于公开数据集的实验表明,LogMerge 在准确性方面优于当前的有监督学习方法和无监督学习方法,并极大地降低了异常标注的开销.在获得更多类型的日志数据后,未来将在更广泛数据集上验证 LogMerge 的性能.

参 考 文 献

[1] China Academy of Information and Communications Technology, Open Data Center Committee. Data center white paper (2018) [R]. Beijing: China Academy of Information and Communications Technology, 2018 (in Chinese) (中国信息通信研究院,开放数据中心委员会.数据中心白皮书(2018)[R].北京:中国信息通信研究院,2018)

[2] Yi Gang, Yang Dan, Wang Mowei, et al. The ACM multimedia 2019 live video streaming grand challenge [C] // Proc of the 27th ACM Int Conf on Multimedia. New York: ACM, 2019: 2622-2626

[3] Wang Mowei, Cui Yong, Xiao Shihan, et al. Neural network meets DCN: Traffic-driven topology adaptation with deep learning [J]. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 2018, 2(2): 1-25

[4] Wang Mowei, Cui Yong, Wang Xin, et al. Machine learning for networking: Workflow, advances and opportunities [J]. IEEE Network, 2017, 32(2): 92-99

[5] Guo Chuanxiong, Yuan Lihua, Xiang Dong, et al. Pingmesh: A large-scale system for data center network latency measurement and analysis [C] //Proc of the 2015 ACM Conf on Special Interest Group on Data Communication. New York: ACM, 2015: 139-152

[6] Liu Dapeng, Zhao Youjian, Xu Haowen, et al. Opprentice: Towards practical and automatic anomaly detection through machine learning [C] //Proc of the 2015 Internet Measurement Conf. New York : ACM, 2015: 211-224

[7] Zhang Shenglin, Liu Ying, Pei Dan, et al. Rapid and robust impact assessment of software changes in large Internet-based services [C] //Proc of the 11th ACM Conf on Emerging Networking Experiments and Technologies. New York: ACM, 2015: 1-13

[8] Mahimkar A, Ge Zihui, Wang Jia, et al. Rapid detection of maintenance induced changes in service performance [C] // Proc of the 7th Conf on Emerging Networking Experiments and Technologies. New York: ACM, 2011: 1-12

[9] Zhang Shenglin, Liu Ying, Pei Dan, et al. Funnel: Assessing software changes in Web-based services [J]. IEEE Transactions on Services Computing, 2016, 11(1): 34-48

[10] Yan He, Flavel A, Ge Zihui, et al. Argus: End-to-end service anomaly detection and localization from an ISP's point of view [C] //Proc of 2012 IEEE INFOCOM. Piscataway, NJ: IEEE, 2012: 2756-2760

[11] Mahimkar A A, Song H H, Ge Zihui, et al. Detecting the performance impact of upgrades in large operational networks [C] //Proc of the ACM SIGCOMM 2010 Conf. New York: ACM, 2010: 303-314

[12] Xu Haowen, Chen Wenxiao, Zhao Nengwen, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in Web applications [C] //Proc of the 2018 World Wide Web Conf. New York: ACM, 2018: 187-196

[13] Bu Jaihao, Liu Ying, Zhang Shenglin, et al. Rapid deployment of anomaly detection models for large number of emerging KPI streams [C] //Proc of 2018 IEEE 37th Int Performance Computing and Communications Conf (IPCCC). Piscataway, NJ: IEEE, 2018: 1-8

[14] Ma Minghua, Zhang Shenglin, Pei Dan, et al. Robust and rapid adaption for concept drift in software system anomaly detection [C] // Proc of 2018 IEEE 29th Int Symp on Software Reliability Engineering (ISSRE). Piscataway, NJ: IEEE, 2018: 13-24

[15] Zhang Ke, Xu Jianwu, Min Martin Renqiang, et al. Automated IT system failure prediction: A deep learning approach [C] //Proc of 2016 IEEE Int Conf on Big Data. Piscataway, NJ: IEEE, 2016: 1291-1300

[16] Du Min, Li Feifei, Zheng Guineng, et al. DeepLog: Anomaly detection and diagnosis from system logs through deep learning [C] //Proc of the 2017 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2017: 1285-1298

[17] Du Min, Li Feifei. Spell: Online streaming parsing of large unstructured system logs [J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 31(11): 2213-2227

[18] Vaarandi R, Blumbergs B, Kont M. An unsupervised framework for detecting anomalous messages from syslog log files [C] //Proc of 2018 IEEE/IFIP Network Operations and Management Symp (NOMS 2018). Piscataway, NJ: IEEE, 2018: 1-6

[19] Astekin M, Zengin H, Sözer H. DILAF: A framework for distributed analysis of large - scale system logs for anomaly detection [J]. Software: Practice and Experience, 2019, 49 (2): 153-170

[20] Lu Siyang, Wei Xiang, Li Yandong, et al. Detecting anomaly in big data system logs using convolutional neural network [C] //Proc of 2018 IEEE 16th Int Conf on Dependable, Autonomic and Secure Computing, the 16th Int Conf on Pervasive Intelligence and Computing, the 4th Int Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech). Piscataway, NJ: IEEE, 2018: 151-158

[21] Xu Wei, Huang Ling, Fox A, et al. Detecting large-scale system problems by mining console logs [C] //Proc of the ACM SIGOPS 22nd Symp on Operating Systems Principles. New York: ACM, 2009: 117-132

[22] Xu Wei, Huang Ling, Fox A, et al. Mining console logs for large-scale system problem detection [C] //Proc of the 3rd Conf on Tackling Computer Systems Problems with Machine Learning Techniques. San Diego, CA: USENIX Association, 2008: 1-4

[23] Lin Qingwei, Zhang Hongyu, Lou Jianguang, et al. Log clustering based problem identification for online service systems [C] //Proc of the 38th Int Conf on Software Engineering Companion. New York: ACM, 2016: 102-111

[24] Qiu Tongqing, Ge Zihui, Pei Dan, et al. What happened in my network: Mining network events from router syslogs [C] //Proc of the 10th ACM SIGCOMM Conf on Internet Measurement. New York: ACM, 2010: 472-484

[25] Kimura T, Ishibashi K, Mori T, et al. Spatio-temporal factorization of log data for understanding network events [C] //Proc of 2014 IEEE Conf on Computer Communications (IEEE INFOCOM). Piscataway, NJ: IEEE, 2014: 610-618

[26] Kimura T, Watanabe A, Toyono T, et al. Proactive failure detection learning generation patterns of large-scale network logs [C] //Proc of the 11th Int Conf on Network and Service Management (CNSM). Piscataway, NJ: IEEE, 2015: 8-14

[27] Zhang Shenglin, Liu Ying, Meng Weibin, et al. Prefix: Switch failure prediction in datacenter networks [J]. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 2018, 2(1): 1-29

[28] Meng Weibin, Liu Ying, Zhu Yichen, et al. LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs [C] //Proc of the 28th Int Joint Conf on Artificial Intelligence (IJCAI 2019). San Francisco, CA: Morgan Kaufmann, 2019: 4739-4745

[29] Pennington J, Socher R, Manning C. GloVe: Global vectors for word representation [C] //Proc of the 2014 Conf on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, 2014: 1532-1543

[30] Zhang Shenglin, Meng Weibin, Bu Jiahao, et al. Syslog processing for switch failure diagnosis and prediction in datacenter networks [C] //Proc of 2017 IEEE/ACM 25th Int Symp on Quality of Service (IWQoS). Piscataway, NJ: IEEE, 2017: 1-10

[31] MacQueen J. Some methods for classification and analysis of multivariate observations [C] //Proc of the 5th Berkeley Symp on Mathematical Statistics and Probability. Oakland, CA: University of California Press, 1967: 281-297

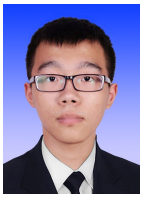
[32] Bouvrie J. Notes on convolutional neural networks [R/OL]. Cambridge, Massachusetts: Center for Biological and Computational Learning, 2006: 38-44 [2019-01-15]. http://cogprints.org/5869/1/cnn_tutorial.pdf

[33] Kawakami K. Supervised sequence labelling with recurrent neural networks [D]. Pittsburgh, Pennsylvania: Carnegie Mellon University, 2008

Zhang Shenglin, born in 1989. Received his BS degree in network engineering from the School of Computer Science and Technology, Xidian University, Xi'an, China, in 2012, and his PhD degree in computer science from Tsinghua University, Beijing, China, in 2017. Assistant professor with the College of Software, Nankai University, Tianjin, China. Member of CCF. His main research interests include failure detection, diagnosis and prediction in data center networks.

Li Dongwen, born in 1997. Received her BS degree in software engineering from Nankai University, Tianjin, China. Master candidate in the College of Software at Nankai University, Tianjin, China. Her current research interests include anomaly detection, deep learning and NLP.

Sun Yongqian, born in 1988. Rreceived his BS degree in statistical specialty from Northwestern Polytechnical University, Xi'an, China, in 2012, and his PhD degree in computer science from Tsinghua University, Beijing, China, in 2018. Assistant professor with the College of Software, Nankai University, Tianjin, China. Member of CCF. His main research interests include anomaly detection, root cause localization, and high performance switching in datacenter.



Meng Weibin, born in 1994. Received his BS degree in software engineering from Jilin University, Changchun, China, in 2016. PhD candidate in the Department of Computer Science and Technology and the Institute for Network Sciences and Cyberspace at Tsinghua University, Beijing, China. His main research interests include anomaly detection, syslog analysis, and failure prediction in datacenter networks.

Zhang Yuzhe, born in 1998. Undergraduate in the College of Software at Nankai University, Tianjin, China. His main research interests include anomaly detection, deep learning and NLP.

Zhang Yuzhi, born in 1964. Received his BS degree and MS degree in computer science from the Department of Computer Science and Technology, Tsinghua University in 1985 and 1987, respectively, and his PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences in 1991. Distinguished professor. His main research interests include deep learning and other aspects in artificial intelligence.

Liu Ying, born in 1973. Received her BS degree in information engineering, MS degree in computer science and PhD degree in applied mathematics from Xidian University in 1995, 1998 and 2001, respectively. She made postdoctoral research in the Department of Computer Science and Technology, Tsinghua University during 2001—2003. Associate professor in the Institute for Network Sciences and Cyberspace, Tsinghua University. Member of CCF. Her main research interests include multicast routing, network architecture, and router design and implementation.

Pei Dan, born in 1973. Received his BE and MS degree in computer science from the Department of Computer Science and Technology, Tsinghua University in 1997 and 2000, respectively, and his PhD degree in computer science from the Computer Science Department, University of California, Los Angeles (UCLA) in 2005. Associate professor in the Department of Computer Science and Technology, Tsinghua University. IEEE senior member and ACM senior member. His main research interests include network and service management in general.