

基于日志的异常检测技术综述

张颖君¹, 刘尚奇², 杨牧², 张海霞¹, 黄克振¹

(1. 中国科学院软件研究所可信计算与信息保障实验室, 北京 100190;

2. 北京市公安局网络安全保卫总队, 北京 100029)

摘要: 日志信息是信息系统快速发展中产生的重要信息资源, 通过日志的分析, 可以进行异常检测、故障诊断和性能诊断等。研究基于日志的异常检测技术, 首先对主要使用的基于日志的异常检测框架进行介绍, 然后对日志解析、日志异常检测等关键技术进行详细介绍。最后对当前技术进行总结, 并对未来研究方向给出建议。

关键词: 异常检测; 日志分析; 机器学习

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2020072

Survey on anomaly detection technology based on logs

ZHANG Yingjun¹, LIU Shangqi², YANG Mu², ZHANG Haixia¹, HUANG Kezhen¹

1. Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

2. Network Security Corps of Beijing Municipal Public Security Bureau, Beijing 100029, China

Abstract: Log information has become an important information resource in the rapid development of information systems. Through the analysis of logs, abnormal detection, fault diagnosis and performance diagnosis can be performed. The log-based anomaly detection technology was focused on. Firstly, the currently used log-based anomaly detection framework was introduced, and then the key link technologies such as log analysis and log anomaly detection were focused on. Finally, the current technology was summarized and suggestions for future research directions were given.

Key words: abnormal detection, log analysis, machine learning

1 引言

现代系统在运行过程中往往产生大量日志文

件, 日志文件包含有关系统中事件的信息。日志分析已经在控制台、网络应用、存储系统、并发系统以及一般系统等广泛应用, 日志文件在监控

收稿日期: 2020-07-09; 修回日期: 2020-09-24

通信作者: 张颖君, yjzhang@tca.iscas.ac.cn

基金项目: 公安部科技项目(2018JSYJA08)

Foundation Item: The Science and Technology Project of the Ministry of Public Security (2018JSYJA08)

论文引用格式: 张颖君, 刘尚奇, 杨牧, 等. 基于日志的异常检测技术综述[J]. 网络与信息安全学报, 2020, 6(6): 1-12.

ZHANG Y J, LIU S J, YANG M, et al. Survey on anomaly detection technology based on logs[J]. Chinese Journal of Network and Information Security, 2020, 6(6): 1-12.

网络情况、检查硬件故障、保护软件安全等方面起着重要作用^[1]。通过分析系统日志记录程序运行时的动态信息,可以有效帮助维护人员分析重现错误,进而更正系统错误,提高系统运行时的可靠性^[2]。然而,在系统运行中,每天产生的日志数量庞大。例如,根据对拥有 1 000 名员工设备的企业的保守估计,小型组织预计每天产生超过 100 GB 的日志流量,峰值 EPS(事件数每秒)超过 22 000^[3]。尽管日志具有巨大的价值,但如何有效地分析仍然是一个巨大的挑战。

在日志分析方面面临的主要问题包括:① 分布式软件系统会生成大量的日志信息,甚至达到每小时吉比特字节的数据^[4];② 不同类型设备包含多样的元素,如路由器、交换机、服务器等生成的日志涉及不同内容,包括网络层、协议和服务等,因此,产生的日志具有多样化的特点^[5];③ 开发人员为了方便,即使是统一类型的设备,不同厂商为了灵活地记录系统事件,使用的日志结构也不同,导致日志信息的非结构化^[6]。总之,自动化地分析日志信息具有较大难度。因此,研究人员主要通过日志解析获取日志相关事件模板,完成对日志的自动化分析。日志解析包括很多方法,如正则表达式、代码分析、机器学习和自然语言处理等相关技术,且已经具有较为成熟的工具。

在日志解析基础上,日志分析应用也很广泛,主要包括异常检测^[7]、故障诊断^[8-10]、程序分析与验证^[11]、调查取证^[12]和性能诊断^[13-14]等。异常检测是指在数据中发现与预期行为不符的模式的问题,这些不合格的模式通常被称为不同应用领域的异常情况、不一致的观察结果、意外情况等^[15]。本文主要针对日志信息进行异常检测,一方面,系统越来越先进和复杂,如果希望系统安全有效,

那么异常检测是一项必不可少的任务^[16];另一方面,执行异常检测对于大规模的开发、维护和性能优化具有重要作用。目前,基于日志的异常检测技术主要包括监督学习、无监督学习和深度学习等方法。

本文重点对基于日志的异常检测框架、日志解析相关技术、日志异常检测相关技术进行详细介绍。

2 基于日志的异常检测框架介绍

由于现有的日志类型繁多,本文主要针对正常日志进行分析(异常日志的发现和本文不进行处理),挖掘系统或设备运行中的异常行为。日志的异常检测框架如图 1 所示,主要包括 4 个部分:日志收集与预处理、日志解析、特征提取和异常检测^[7]。

首先,进行日志收集,获取原始数据并进行预处理。日志收集^[17]主要指从设备中获取相关日志记录,包括系统的运行状态、时间信息等,这些信息用来记录系统运行过程中的重要信息,是日志分析的前提。在收集完日志数据后,进行预处理,主要是根据具体需求,将日志数据中的无效信息进行剔除,包括重复信息、无用信息等。日志信息中的事件序列如图 2 所示。

其次,对处理后的数据进行解析,分析日志结构与信息,获取日志针对每个事件的模板。日志解析是从无结构的日志中提取相应的事件模板,每个模板由多个指定参数构成,作为后续特征提取的基础。由于日志数量庞大,当前主要采用固定时间窗口、滑动时间窗口、会话窗口等为日志分割的方式^[7],将庞大的日志划分为多个部分进行检测,或者通过采样的方式进行检测。在数据采样过程中,采样方式、样本尺寸与检测正

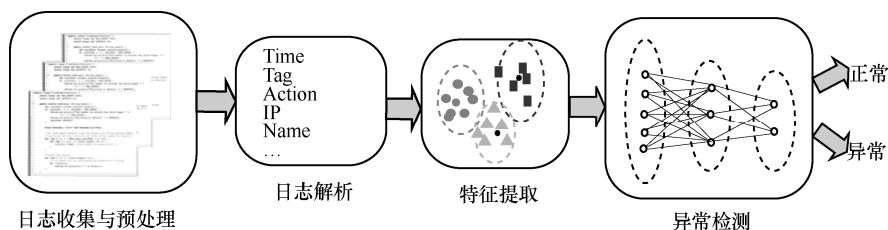


图 1 基于日志的异常检测框架
Figure 1 Log-based anomaly detection framework

```

2018-03-12 17:00:14,833 INFO [main] server.ZooKeeperServer: Server environment:host.name=localhost
2018-03-12 17:00:14,834 INFO [main] server.ZooKeeperServer: Server environment:user.home=/home/hadoop
2018-03-12 17:00:14,854 INFO [main] server.NIOServerCnxnFactory: binding to port 0.0.0.0/0.0.0.0:2181
2018-03-12 17:00:17,882 INFO [NIOServerCnxn.Factory:0.0.0.0/0.0.0.0:2181] server.NIOServerCnxnFactory:
Accepted socket connection from /192.168.31.154:38221
2018-03-12 17:00:17,894 INFO [NIOServerCnxn.Factory:0.0.0.0/0.0.0.0:2181] server.ZooKeeperServer: Client
attempting to establish new session at /192.168.31.154:38221
2018-03-12 17:00:17,944 INFO [SyncThread:0] server.ZooKeeperServer: Established session
0x1621970549a0000 with negotiated timeout 90000 for client /192.168.31.154:38221

```

图2 Hadoop 日志信息样例
Figure 2 Hadoop log information sample

确率关系密切,因此,在该方向进一步探索,对提高所提方案的实用性有重要推动作用。本文通过设定固定的时间范围作为窗口,对窗口内日志内容进行解析,提炼其中的事件模板,将日志切分成一组日志事件序列 $\text{Log}=\{e_1, e_2, \dots, e_n\}$,进而从提取信息中选择合适的变量和分组来表示相关内容,并进行数字化向量表示,构建成特征向量,方便后续进行机器学习。例如,对一段时间窗口内的信息组成一个日志序列(图2中的内容为一个时间窗口),根据日志解析后的事件模板进行分析(如表1所示)。

为了描述清晰,这里将不同服务器相关操作构建不同的特征向量,方便后续挖掘服务器中的异常操作。实际解析中,可以将不同服务器的操作进行关联分析,获得更多的异常信息。例如,针对服务器“ZooKeeperServer”构建特征向量 $\langle 2, 1, 1, \dots \rangle$,表示事件1“server environment”发生2次,“client attempting to establish new session”“established session”各发生1次。最终,可以由不同时间段形成的特征向量构建成特征矩阵。

最后,进行异常检测过程,主要是使用机器学习的方法(如聚类分析、决策树、深度学习等)对特征向量进行学习,产生一个异常检测模型,通过对输入特征矩阵机器学习模型进行训练,从而生成一个异常检测模型,并使用该模型对新的

日志进行检测。例如,在将日志信息标记为正常、异常后,根据提取的特征进行聚类分析。首先对各个特征赋予不同权重;然后将特征向量进行归一化处理;之后定义特征向量之间的相似性(即距离大小);最后通过定义阈值,对日志间的相似性进行判断,定义是否为异常。针对异常检测结果,进一步进行人工分析和确认,结果可以作为样本加入数据集中,对模型进行修正。

综上所述,基于日志的异常检测技术中,各个环节都不可或缺。但是,由于日志获取部分,不同系统、不同用户有不同实现方式。而特征提取及表示主要为异常检测做准备,与异常检测使用的模型及方法直接相关,因此,本文对特征提取与表示合并入异常检测部分进行介绍。

3 日志解析相关技术

目前,日志解析技术在工业界取得了较大进展,已开发完成多个日志解析工具,并进行应用,日志解析器跨越分布式系统、超级计算机、操作的数据集系统、移动系统、服务器应用程序和独立软件^[18]。较为典型的包括 Splunk^[19]、Logentries^[20]、Logz.io^[21]等。但大多针对专业领域进行应用,本文重点对学术研究中相关技术进展进行介绍。

在自动化解析日志过程中,传统的方式主要

表1 日志模板示例
Table 1 Log template samples

Eid	Name	Message	Frequency	Parameters
1	server.ZooKeeperServer	Server environment	2	host.name=localhost, user.home=/home/hadoop
2	server.NIOServerCnxnFactory	binding to port	1	0.0.0.0/0.0.0.0:2181
3	server.NIOServerCnxnFactory	Accepted socket connection	1	192.168.31.154:38221
4	server.ZooKeeperServer	Client attempting to establish new session	1	192.168.31.154:38221
5	server.ZooKeeperServer	Established session	1	0x1621970549a0000
...	...	...	...	...

通过正则表达式来提取事件模板和主要参数^[22]。但是,手动分析并写正则表达式规则是很耗时的,而且日志更新速度非常快,导致正则表达式的效率不高。因此,很多研究主要使用代码分析、机器学习、自然语言处理等方法进行日志模板的提取,方便对日志信息进行解析。

3.1 基于代码分析的日志解析技术

在代码分析方面,主要采用静态分析的方法从源代码中提取事件模板。其中,文献[23]在日志解析过程中主要分为静态源代码解析和运行时日志解析两个步骤。静态源代码分析步骤将程序源(以及记录器类的名称)作为输入,首先生成源代码的抽象语法树(AST);然后枚举所有类中的 toString 调用,并查看这些调用中的字符串格式语句,以推断出消息模板中的变量类型,并将此类型信息替换为部分模板,递归地执行此操作,直到所有模板仅插值原始类型为止。文献[24]主要将执行日志解析为事件序列,并通过构造事件的后缀数组来生成所有可能子序列的有序列表。实验证明,该方法具有较高的效率。但在实际中,源代码并不容易获取。文献[25]提出了一种新的基于概率后缀树的统计方法,使用源代码分析对日志信息之间的逻辑关系进行解析,然后进行准确的分割以提取完整的执行路径,之后通过定义异常路径索引进行异常检测,该方法可以有效进行自动异常检测。但是,大多数情况获取代码并不容易。因此,研究人员展开了利用机器学习相关技术进行日志解析的研究。

3.2 基于机器学习的日志解析技术

在机器学习方面,日志解析技术主要包括基于聚类的数据挖掘方法、基于自然语言处理的解析方法等。日志是由一系列系统或应用中发生的事件进行记录,一个记录通常代表一个事件发生的相关具体内容。参照 RFC^[26],日志记录通常包含但不限于事件的时间、源头、标签、信息等。在日志解析过程中,一般需要对日志记录进行分析,去除无关部分(如空格等),然后通过机器学习相关方法对日志内容进行学习。

在基于聚类的日志解析方法中,主要通过计算对日志进行聚类,进而进行数据挖掘,形成事件模板,完成对日志的解析。其中,日志解析工

具 SLCT^[27]通过 3 个步骤进行工作:首先,从数据中识别出频繁出现的单词(比支持阈值出现频率更高的单词)或 1 个项集;然后,提取数据集中每一行中出现的这些项的组;最后,将出现比支持值更频繁出现的候选簇作为数据集中的簇。LogCluster^[28]是对 SLCT 的改进,实现对数据的聚类进而挖掘频繁发生的线性模式。IPLoM^[29]是一种迭代分区日志挖掘算法,用来挖掘事件日志中的事件类型。通过三步分层分区过程,IPLoM 将日志数据分区到各自的群集中。与其他类似算法不同,IPLoM 基于启发式的方法,主要对于日志中每个词的位置进行计数,反复出现的文字会被选取,并组成事件候选,最后选出合适的词形成日志事件。文献[30]使用统计聚类方法从非结构化日志消息中提取主要模板,然后建立一个统计模型 LTF,捕获日志消息的时空模式,方便对大量网络操作进行有效分析,发现其中的异常操作。LKE^[31]通过将日志自由格式的文本转换为日志键,然后对日志键进行聚类。通过将每个日志消息对应到一系列日志键,可以将日志消息转化为日志键序列。当收到新的日志消息时,首先根据规则提取日志消息中的日志键,然后选择与原始日志键最近距离的键,进而进行异常判断。LogSig^[32]首先将每个日志消息转换为一组单词对,以对单词及其位置信息进行编码;然后基于单词对为每个日志消息计算一个潜在值,以确定该日志消息可能属于哪个群集。经过多次迭代后,可以对日志消息进行集群。最后,在每个群集中,利用日志消息来生成日志模板。文献[33]提出了一种称为 POP 的并行日志解析方法,该方法可以准确有效地解析大规模日志数据。为了提高日志解析的准确性,本文使用迭代分区规则来生成候选事件,并使用分层聚类来改进事件类型。为了提高处理大规模数据的效率,本文将 POP 设计为线性时间复杂度的日志大小,并在大型数据处理平台 Spark 的基础上进一步优化其计算。上述方法大多采用聚类和数据挖掘相关算法进行日志解析,但是结果大多基于经验值,一般参数选取过程比较困难。文献[34, 35]通过将安全日志表示为图数据结构,日志依据其相似性进行自动分组,达到较好的聚类效果,然后自动化地提取模板,

是一种自动化的方法实现日志自动解析。此外，为了提高日志解析的自动化，减少人的参与，一些研究人员提出将日志作为文本进行自然语言处理，根据不同单词进行解析的方法。

在自然语言处理方面，Logram^[36]是一种自动日志解析方法，利用自然语言处理中的 *n*-gram 字典来实现有效的日志解析。文献[37]使用自然语言处理对日志进行解析。首先，将文本划分为单词进行标记；然后，对单个单词进行词干分析、同义词替换、停用词删除等；之后，将单词列表中的句子转化为矢量表示形式；在运行分类之前进行哈希，之后运用两种无监督聚类算法 bisecting k-means 和 Latent Dirichlet Allocation 进行分类。文献[38]提出了一种基于经典自然语言处理技术的实现日志分析。首先，使用快速的 *n*-gram 语言模型对消息的类型进行预过滤，其中考虑了语法和词汇的复杂性。然后，使用 LDA 从大量未标记消息中构建语义空间。最后，使用标准的前馈多层感知器来实现 3 种分类，并以最少的人工监督。NLog^[39]是一种基于自然语言处理的分布式数据分析系统，用于日志解析。首先，通过自然语言处理（NLP）从日志消息中获取关键信息并确定由不同语句生成的日志中的相同对象；之后，使用 keyed message 存储结构来存储解析的日志；最后，实现 NLog 并将其应用于分布式数据分析框架 Spark 和 MapReduce。

3.3 其他技术

此外，Fastlogsim^[40]通过对文本相似度进行计算，合并类似模板，实现快速日志解析。Spell^[22,41]

利用最长的公共子序列算法（LCS）以流方式解析日志，实现在线系统日志检测。MoLFI^[42]将日志解析建模为多目标优化问题，并使用 NSGA-II 进化算法解决。Drain^[43]应用固定深度的树结构来表示日志消息，并有效地提取常用模板。文献[44]主要是对给定两个或多个系统的执行日志，输出它们差异的模型。文献[45]构建 CD-LDA 方法实现日志分析。首先，通过变更点检测算法识别网络中新事件开始或现有事件发生的时刻，然后通过 LDA 算法将消息分类为事件。LogHound^[27]主要使用频繁项集挖掘算法从事件日志中发现频繁模式。

在日志解析评估方面，文献[46]提出了基于 *p* 值的内部质量评估多种日志解析算法，通过评估，可以对日志解析器的质量有深入了解。文献[47]对多个日志解析器进行评估，并对成功案例和经验进行介绍。文献[48]提出一个质量模型用来评估自动化日志提取技术，该模型包含 7 个相关参数；并利用该模型对现有技术进行综合评估。本文对上述典型方法进行了对比（如表 2 所示），主要从类别、主要算法、准确性、模式、使用范围、时间复杂度、查找效率等方面进行比较。各方法测评过程使用的测试环境、设备类型等不同，因此不能简单地直接将实验结果进行对比，本文通过转换分析等方法，将结果采用分级方式进行描述，用“+++”表示较高等级，“++”表示中等级，“+”表示较低等级。例如，CFG^[25]中主要采用代码分析方法，涉及的算法是 AST、CFG。该方法采用离线模式，准确性一般，构建时间复杂度为

表 2 日志自动化解析典型方法对比
Table 2 Comparison of typical methods for automatic log analysis

名称	类别	使用算法	模式	准确性	时间复杂度	效率
CFG ^[25]	代码分析	AST,CFG	离线	++	$O(n^3)$	+++
PCA ^[26]	代码分析	AST	在线	+++	$O(n)$	+++
CLSTR ^[29]	机器学习	IPLoM	离线	++	$O(n)$	+++
LKE ^[31]	机器学习	Clustering	离线	++	$O(n^2)$	+
Logram ^[36]	自然语言处理	<i>n</i> -gram	在线	+++	$O(n)$	+++
NLog ^[39]	自然语言处理	POS	离线	++	$O(n)$	+++
Spell ^[22]	经典算法	LCS	在线	+++	$O(n)$	+++
Drain ^[43]	经典算法	解析树	在线	+++	$O(n)$	+++

$O(n^3)$ ，查找效率较高。

综上所述，当前日志解析的技术已经较为成熟，主要通过定义规则、代码分析和机器学习等方法，实现为不同日志结构的快速解析，去除不相关内容，将不同事件的关键字段进行提取与存储，方便后续分析。自然语言处理方面，考虑到性能等影响，目前只是运用简单的相关技术进行日志解析，后续需要将更为复杂的主题划分、同义词替换等相关技术引入日志分析中，使日志分析后具有更强的可解释性，方便操作和理解，并进行实际应用。

4 基于日志的异常检测方法

当前基于日志的异常检测方法已经有很多研究，由于现有系统的规模和复杂性不断增加，手动检查日志以发现异常变得不可行。因此，针对日志信息进行自动异常检测的需求越来越迫切，并在相关领域提出许多方法，包括基于规则^[49-50]、基于工作流^[51]、基于时间序列分析^[52]和基于图^[53]的异常等检测方法。机器学习相关技术的迅速崛起，触发了其广泛应用，本文重点对当前研究热点、基于机器学习的日志异常检测方法进行介绍，主要包括监督学习、无监督学习和现阶段常用的深度学习等方法。

4.1 监督学习

监督学习被定义为一种从标记的训练数据中得出模型的机器学习方法。带标签的训练数据（通过标签指示正常或异常状态）是监督异常检测的先决条件，训练数据标签越多，模型越精确。在基于日志的异常检测方面使用的机器学习方法较为宽泛，典型的方法如下。

AClog^[54]是一种多日志相关分析系统构建攻击链的方法。首先，使用不同的日志来形成系统的步骤，它们是攻击前的事件序列。然后，利用快速线性 SVM 和最长公共子序列来发现攻击前的常规步骤。最后，通过相应的日志序列来识别攻击前的步骤。LogClass^[55]是一个数据驱动的框架对设备日志进行异常检测和分类。LogClass 是预处理设备日志，并产生词描述方法 bag-of-words 矢量，然后通过 PU 学习方法和 SVM 来训练异常检测和分类模型，最后进行在线异常

检测。文献[56]提出了一种使用 K-原型聚类 and k-NN 分类算法的集成方法（IM）从大量日志中执行异常检测。首先，通过分析系统日志的特征，并根据会话信息提取 10 个特征，有效地表征用户行为。其次，基于这些提取的特征，K-原型聚类应用算法将数据集划分为不同的群集。然后，明显的正常事件通常会滤除高度相干的簇，而其他簇则被视为异常候选进行进一步分析。最后，设计了两个基于距离的新功能来测量本地和全局这些异常候选者的异常程度。文献[57]是一种使用数据挖掘和统计学习方法自动监视，并从控制台日志中检测异常的分析方法。首先，使用频繁模式挖掘和分布估计技术以捕获主导模式（频繁序列和持续时间）。然后，使用主成分分析法异常检测技术来识别实际问题。文献[58]通过从日志中挖掘模板和模板序列来构建控制流图（CFG）。CFG 式系统状态基线代表正常执行流，通过对比 CFG 相似性发掘异常行为。

由于监督学习需要前期进行人工标记，工作量大，因此，自动化程度相对不高。更多的研究集中到自动化程度更高的无监督学习。

4.2 无监督学习

无监督学习是指没有任何标记的训练数据情况下得到模型的机器学习方法。常见的无监督方法包括各种聚类方法^[22]、关联规则挖掘、基于主成分分析等，具体如下。

（1）聚类方法

LogCluster^[59]是一种将日志聚类来进行日志问题识别的方法。首先将每个日志序列通过一个向量表示，然后计算两个日志序列之间的相似度值，并应用聚集层次聚类技术将相似的日志序列分组为聚类。Log2vec^[60]是一种基于异构图嵌入的威胁检测方法。首先，根据日志条目之间的关系将其转换为异构图；然后使用一个改进的图嵌入方法，可以自动对每个日志描述为一个低维的向量；最后，通过检测算法，可以将恶意日志和良性日志进行聚类，识别出恶意行为。文献[61]提出一种基于 syslog 的系统日志文件中检测异常日志消息的框架，并使用 LogCluster 算法进行模式检测。

（2）关联规则挖掘

文献[62]提出了一种非结构化日志异常检测

的新型分析技术。首先,使用日志解析器将非结构化日志转换为结构化日志。然后,根据日志参数之间的关系,结构化日志消息被进一步分组以记录消息组。之后,程序不变式将自动从日志消息组中挖掘,挖掘的不变量可以揭示内在的线性特征程序工作流程,自动检测日志中的异常。文献[63]提出了一种基于学习的方法进行云服务异常检测,通过利用来自多个服务的日志来细粒度地发现深度云服务异常的原因。首先,通过最长公共子序列(LCS)方法将日志解析为三元组。然后,使用时间关联分析(TCA)将相似事件聚类,找出操作的关键事件。最后,根据标记进行异常检测。LogLens^[64]是实时日志分析系统,采用基于无监督的机器学习技术,以发现应用程序日志中的模式,然后利用这些模式以及实时日志解析,设计高级日志分析应用程序。

(3) 基于主成分分析(PCA)

文献[23]首先通过系统执行时的特征构造特征矩阵,然后使用PCA^[65]与信息检索中的术语加权技术^[66]相结合,构造异常检测方法。DILAF^[65]是一个大型系统日志的分布式分析异常检测方法。DILAF采用了批量同步并行计算(BSP)模型来并行化机器学习过程。BSP算法将工作负载分配到各个参与流程之间,每个流程均执行本地计算。当一个进程完成其任务并到达障碍时,它将等待,直到所有其他进程达到相同的同步障碍。本文使用BSP模型实现了PCA算法进行异常检测。文献[67]是对DILAF的扩展,通过增量分析、分布式和无监督的日志分析框架,更早检测到异常的离线分析方法。文献[68]针对异常检测中两种无监督机器学习算法PCA和Kmeans算法进行评估。

(4) 其他方法

文献[69]提出了使用递归神经网络(RNN)语言模型对系统日志进行异常检测。通过对系统日志中事件的正态分布进行建模,异常检测方法可以发现隐藏在这些日志中的复杂关系。文献[70]使用自然语言处理技术,将日志作为常规文本进行分析。首先,应用基于Google的word2vec算法的嵌入技术,将日志文件的单词映射到高维度空间,然后使用标准分类器进行异常检测。文

献[71]提出了一种利用混合解决方案——卷积自动编码器-可变自动编码器(CAE-VAE)结构来离散事件序列,并利用重构误差进行异常检测。但该方法不能在连续事件上进行分析。

总之,无监督学习具有较高的自动化,节省了人工成本,但是检测中大多使用聚类、关联分析等方法对事件进行分析,事件间自动化关联程度不高,因此,更多的研究使用深度学习方法,将事件进行联合分析,自动化发现潜在安全威胁。

4.3 深度学习

近年来,基于深度学习的异常检测算法变得越来越流行,并且在多种任务集中深度学习的应用超越了传统方法^[72],因此这里单独对其进行介绍。

CausalConvLSTM^[73]是一种新型混合模型,它利用卷积神经网络(CNN)的优势,可以有效地提取并行方式中的空间特征,并通过长短期记忆(LSTM)网络来捕获顺序关系。该模型可以有效解决异常检测中概念漂移的问题。DeepLog^[74]是主要使用长短期记忆的神经网络模型,将系统日志建模为自然语言序列。DeepLog从正常执行中自动学习日志模型,并通过该模型,对正常执行下的日志数据进行异常检测。LogGAN^[75]是一种基于LSTM的生成对抗网络,以基于时态系统日志区分即将到来的异常事件。首先,利用自定义的日志解析器提取结构化信息(即时间戳、签名和参数),并将每个日志转换为事件。其次,事件的组合(即模式)使用滑动窗口从时态系统日志中收集相应的即将发生的事件。LogRobust^[76]是一种基于不稳定日志数据的异常检测方法。LogRobust提取日志事件的语义信息并将它们表示为语义向量。然后通过使用基于注意力的Bi-LSTM模型检测异常,该模型具有捕获日志序列中的上下文信息的能力,并且自动了解不同日志事件的重要性。文献[77]提出了一个基于行为和属性进行无监督学习的日志数据分析方法。首先,通过长短期记忆来预测当前的一级日志数。通过比较预测数量的一级日志,它可以检测异常行为,如日志数量激增。进而,执行语义分析属性信息以构建属性语法森林,协助维护人员监视系统通过日志中的关键指纹信息。此外,

采用属性信息和状态值来训练无监督学习算法模型,共同实现对日志的分析,协助操作员对系统进行后期维护。文献[46]设计一个分类器,该分类器使用长短期记忆将用户日志建模为自然语言顺序并实现基于角色的分类。LSTM 模型可以通过自动提取特征学习不同用户的行为模式,并检测日志中的异常行为。此外,文献[78]提出基于日志信息和 CNN-text 的异常检测方法,通过将 CNN-text 与主流的 SVM、决策树等对比,显示该算法的优越性。文献[79]提出了一种利用卷积神经网络从大数据系统日志中进行检测的异常处理方法。该方法可以学习系统日志中事件关系,并精确进行异常检测。

总之,深度学习方法更多关注连续事件之间的关系,不再仅针对单一离散事件进行分析,但深度学习较大程度上依赖数据集,如何构建通用且高效的方法是下一步研究的重点。

综上所述,基于日志的异常检测技术已经得到广泛关注,越来越多的机器学习相关技术融入其中。本文对各方面进行了量化分析,分析了其方法的准确性和召回率。由于各方法实现不同,且难以获得源代码进行验证,因此本文使用原文献中的相关描述作为评测依据。表 3 中记录了各方案的准确率和召回率(部分文献未提供其方法的结果以“/”表示)。但是,大部分现有方法是针对特定场景进行分析、提取特征及构建检测算法,缺乏将不同类型日志联合分析,并构建通用且高准确性的检测方法。

5 未来研究方向与挑战

在分析和调研现有基于日志异常检测技术相关文献基础上,本文列出一些未来研究的主要方向和面临的挑战。

(1) 日志数据集的统一:一方面,现有日志分析的数据集种类繁多,各类日志差异较大,给日志分析对比带来较大的挑战;另一方面,日志的标记是庞大的工作,标记数据的质量直接影响着后续学习的效果。因此,是否可以提供统一的多类型数据集,并提供相应的标记数据供实验分析和对比验证,将成为后续研究的基础。此外,对日志数据集的划分方式的进一步探索,会对日志分析效率和准确性产生较大影响。

(2) 加强多日志间的关联分析:现有技术大多关注某一设备的日志,分析其中的异常行为。但现在网络攻击步骤复杂,攻击者操作更为隐蔽,是否可以将相关设备的日志进行关联分析,利用深度学习等技术挖掘隐藏的攻击路径,发现异常行为将成为新的挑战。

(3) 有限恶意数据情况下的智能模型训练:在实际工作部署中,恶意攻击产生的带标签数据相对较少,在图像领域的图像增强方法难以对恶意数据进行扩充增强,因此有必要研究针对日志检测的数据增强方法以及模型训练方法,在恶意攻击带标签数据有限的情况下实现高准确性模型的训练,并完成实际的检测。

(4) 人工模型的可解释性研究:当前研究对

表 3 日志异常检测方法对比
Table 3 Comparison of log anomaly detection methods

名称	分类	使用的算法	模式	准确率	召回率
AClog ^[54]	监督学习	SVM, LCS	离线	92.4%	80%
IM ^[56]	监督学习	K-prototype, kNN	离线	89%	85%
LogClass ^[55]	监督学习	PU, SVM	在线	99.048%	99.988%
LogCluster ^[59]	无监督学习	聚类算法	在线	60%	36.2%
MCL ^[23]	无监督学习	PCA	在线	99.8%	/
LACT ^[63]	无监督学习	TCA, NLP	离线	97.08%	95.45%
CausalConvLSTM ^[73]	深度学习	CNN, LSTM	离线	89.59%	99.72%
DeepLog ^[74]	深度学习	LSTM	在线	95%	96%
LogGAN ^[75]	深度学习	LSTM	离线	100%	35.6%
LogRobust ^[76]	深度学习	Bi-LSTM	在线	98%	100%

于深度学习模型的可解释性存在疑虑,这也是深度学习模型在安全领域难以被大规模应用的主要原因之一。在日志检测过程中,模型对于检测结果应具备可解释性,能够让分析人员理解被标记的数据异常的原因,这对于分析人员对结果的判定具有依据。在未来的工作中,模型本身的可解释性相关研究应与日志检测方向结合,从而提高判定依据的可信度。

(5) 人工智能模型的安全性研究: 相对于运用规则的传统方法,智能化方法虽然带来了检测性能的提高,但同时面临着其模型本身的安全缺陷和/或多种攻击,如模型脆弱性问题、对抗样本攻击、数据投毒攻击等,这些攻击将影响模型本身的准确性,使攻击能够成功逃逸,避免被检测。因此,有必要借鉴于模型的安全性增强方法,减少甚至避免因模型脆弱性产生的攻击。

(6) 异常检测方法的实用性研究: 目前,异常检测方法在多个领域展开了广泛研究,但在实际检测中并没有得到广泛应用。一方面,误检率和漏检率是异常检测的一个难点,能够有效控制误检率和漏检率,异常检测方法才能有效实施;另一方面,增加检测次数可以减少犯错概率(包括误检率、漏检率等),检测次数与犯错概率间存在一定关系,如何找到两者关系,在较少检测次数情况下得到较少的犯错概率,是异常检测方法得以实用化的重要研究方向。

6 结束语

日志已经成为当前信息系统产生的重要信息资源。基于日志的异常检测技术可以有效发现系统中存在的安全问题,发掘潜在的安全威胁,成为当前研究的热点。随着人工智能技术的发展和普及,越来越多的相关研究成果已经应用于基于日志的异常检测。在基于日志的异常检测方法中,包括日志收集、日志解析、特征提取、异常检测等步骤。其中,日志解析和异常检测是核心部分,也是本文重点论述的内容。当前,日志解析技术已经相对成熟,相关解析工具也应用广泛。日志解析的研究从传统的定义正则表达式发展到自动化的方法,主要包括代码分析、机器学习和自然语言处理等。在基于日志的异常检测方法中,主

要分为监督学习、无监督学习和深度学习等。异常检测方法大多针对特定场景和数据集进行离线分析,缺乏通用性和高准确性的实用方法。

现在的攻击越来越隐蔽,攻击步骤越来越烦琐,而对相关设备的日志联合分析可以有效发现潜在攻击,因此,在后续的研究中,不仅关注单一日志来源,还需要结合不同事件、不同设备进行日志解析,进而进行异常检测等;此外,利用机器学习的相关研究将进一步应用于在线检测,构建通用、有效的在线基于日志的异常检测方法,并应用到实际中。

参考文献:

- [1] 廖湘科, 李姗姗. 大规模软件系统日志研究综述[J]. 软件学报, 2016, 27(8): 1934-1947.
LIAO X K, LI S S. Survey on log research of large scale software system[J]. Journal of Software, 2016, 27(8): 1934-1947.
- [2] OLINER A J, GANAPATHI A, XU W. Advances and challenges in log analysis[J]. Communications of the ACM, 2012, 55(2): 55-61.
- [3] RAPIDS. cyBERT: neural network, that's the tech; to free your staff from, bad regex[EB].
- [4] MI H, WANG H, ZHOU Y, et al. Toward finegrained, unsupervised, scalable performance diagnosis for production cloud computing systems[J]. IEEE Trans Parallel Distrib Syst, 2013, 24(6): 1245-1255.
- [5] 崔元, 张琢. 基于大规模网络日志的模板提取研究[J]. 计算机科学, 2017, 44(11A): 448-452.
CUI Y, ZHANG Z. Research on Template Extraction Based on Large-scale Network Log[J]. Computer Science, 2017, 44(11A): 448-452.
- [6] HE P, ZHU J, HE S, et al. An evaluation study on log parsing and its use in log mining[C]// Proc of the 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. 2016: 654-661.
- [7] HE S L, ZHU J M, HE P J, et al. Experience report: system log analysis for anomaly detection[C]// 27th International Symposium on Software Reliability Engineering. 2016: 207-218.
- [8] FU Q, LOU J, et al. Contextual analysis of program logs for understanding system behaviors[C]// MSR '13. 2013: 397-400.
- [9] PECCHIA A, COTRONEO D, KALBARCZYK Z, et al. Improving log-based field failure data analysis of multi-node computing systems[C]// Dependable Systems and Networks. 2011: 97-108.
- [10] LU J, LI F, LI L, et al. CloudRaid: hunting concurrency bugs in the cloud via log-mining[C]// Foundations of Software Engineering, 2018: 3-14.
- [11] AIT EL HADJ M, KHOUMSI A, BENKAOUZ Y, et al. Efficient security policy management using suspicious rules through access

- log analysis[J]. Lecture Notes in Computer Science, 2019, 11704: 250-266.
- [12] STUDIAPAN H, FERDOUS S, PAYNE C. A survey on forensic investigation of operating system logs[J]. Digital Investigation, 2019, 29: 1-20.
- [13] CHOW M, MEISNER D, FLINN J, et al. The mystery machine: end-to-end performance analysis of large-scale internet services[C]// 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI '14). 2014: 217-231.
- [14] KARTHIK N, CHARLES K, JENNIFER N. Structured comparative analysis of systems logs to diagnose performance problems[C]// Networked Systems Design and Implementation. 2012: 26-26.
- [15] CHANDOLA V, BANERJEE A, KUMAR V. anomaly detection: a survey[J]. ACM Computing Surveys, 2009, 41(3):30602-30611.
- [16] KULKARNI J, JOSHI S, BAPAT S, et al. Analysis of system logs for pattern detection and anomaly prediction[C]// Proceeding of International Conference on Computational Science and Applications. 2020: 427-436.
- [17] ARIEL R, RANDY K. Chukwa: a system for reliable large-scale log collection[C]// Usenix Large Installation Systems Administration Conference. 2010: 1-15.
- [18] ZHU J M, HE S L, LIU J Y, et al. Tools and benchmarks for automated log parsing[C]// International Conference on Software Engineering. 2019: 121-130.
- [19] Splunk[EB].
- [20] Logentries[EB].
- [21] Logz.io[EB].
- [22] DU M, LI F F. Spell: streaming parsing of system event logs[C]// ICDM 2016. 2016: 859-864.
- [23] XU W, HUANG L, FOX A, et al. Detecting large-scale system problems by mining console logs[C]// Symposium on Operating Systems Principles, 2009: 117-132.
- [24] NAGAPPAN M, WU K, MLADEN A. Efficiently extracting operational profiles from execution logs using suffix arrays[C]// ISSRE. 2009: 41-50.
- [25] BAO L, LI Q, LU P Y, et al. Execution anomaly detection in large-scale systems through console log analysis[J]. Journal of Systems and Software, 2018, 143: 172-186.
- [26] LONVICK C. The BSD syslog protocol[EB].
- [27] VAARANDI R. Mining event logs with slct and loghound[C]// Proceedings of the 2008 IEEE/IFIP Network Operations and Management Symposium. 2008: 1071-1074.
- [28] RISTO V, PIHELGAAS M. LogCluster — a data clustering and pattern mining algorithm for event logs[C]// Conference on Network and Service Management (CNSM). 2015: 1-7.
- [29] MAKANJU A, ZINCIR-HEYWOOD 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.
- [30] TATSUAKI K, KEISUKE I, TATSUYA Mori, et al. Spatio-temporal factorization of log data for understanding network events[C]// IEEE INFOCOM 2014. 2014: 610-618.
- [31] FU Q, LOU J G, WANG Y, et al. Execution anomaly detection in distributed systems through unstructured log analysis[C]// (ICDM'09) Proc of International Conference on Data Mining. 2009: 149-158.
- [32] TANG L, LI T, PERNG C S. LogSig: generating system events from raw textual logs[C]// CIKM'11: Proc. of ACM International Conference on Information and Knowledge Management. 2011. 785-794.
- [33] HE P J, ZHU J M, HE S L, 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.
- [34] STUDIAPAN H, SOHEL F, PAYNE C. Automatic event log abstraction to support forensic investigation[C]// ACSW 2020. 2020: 1-9.
- [35] STUDIAPAN H, PAYNE C, SOHEL F. Automatic graph-based clustering for security logs[C]// Advanced Information Networking and Applications(AINA). 2019: 914-926.
- [36] DAI H, LI H, CHEN C S, et al. Logram: efficient log parsing using n-gram dictionaries[R]. 2020.
- [37] NICOLAS A, YOHAN P, SOPHIE C, et al. Improving performances of log mining for anomaly prediction through NLP-based log parsing[C]// Modeling Analysis And Simulation on Computer and Telecommunication Systems. 2018: 237-243.
- [38] Li G F, ZHU P J, CAO N, et al. Improving the system log analysis with language model and semi-supervised classifier[J]. Multimedia Tools and Applications, 2019, 78(15): 21521-21535.
- [39] PI A D, CHEN W, ZELLER W, et al. It can understand the logs, literally[C]// International Parallel and Distributed Processing Symposium. 2019: 446-451.
- [40] LIU W Y, LIU X, DI X Q, et al. FastlogSim: a quick log pattern parser scheme based on text similarity[C]// Knowledge Science Engineering and Management. 2020: 211-219.
- [41] DU M, LI F F. Spell: online streaming parsing of large unstructured system logs[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(11): 2213-2227.
- [42] MESSAOUDI S, PANICHELLA A, BIANCULLI D, et al. A search-based approach for accurate identification of log message formats[C]// ICPC. 2018: 167-177.
- [43] HE P J, ZHU J M, ZHENG Z B, et al. Drain: an online log parsing approach with fixed depth tree[C]// ICWS. 2017: 33-40.
- [44] BAO L F, BUSANY N, LO D, et al. Statistical log differencing[C]// Automated Software Engineering. 2019: 851-862.
- [45] SIDDHARTHA S, SUPRATIM D, SRIKANT R, et al. Learning latent events from network message logs[J]. IEEE ACM Transac-

- tions on Networking, 2019, 27(4): 1728-1741.
- [46] XIE X S, WANG Z, XIAO X H, et al. A confidence-guided evaluation for log parsers inner quality[J]. *Mobile Networks and Applications*, 2020: 1-12.
- [47] ZHANG D X, ZHENG Y, WEN Y, et al. Role-based log analysis applying deep learning for insider threat detection[C]// *SecArch'18*. 2018: 18-20.
- [48] EL-MASRIA D, PETRILLO F, YANN-GAËL G, et al. A systematic literature review on automated log abstraction techniques[J]. *Information & Software Technology*, 2020, 22: 1-18.
- [49] OPREA A, LI Z, YEN T F, et al. Detection of early-stage enterprise infection by mining large-scale log data[C]// *Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 2015: 45-56.
- [50] MARCELLO C, DOMENICO C, ANTONIO P. Event logs for the analysis of software failures: a rule-based approach[J]. *IEEE Transactions on Software Engineering*, 2013, 39 (6): 806-821.
- [51] LOU J G, FU Q, YANG S G, et al. Mining program workflow from interleaved traces[C]// *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2010: 613-622.
- [52] BEZERRA F, WAINER J. Algorithms for anomaly detection of traces in logs of process aware information systems[J]. *Information Systems*, 2013, 38(1): 33-44.
- [53] JIA T, CHEN G, YANG L, et al. An approach for anomaly diagnosis based on hybrid graph model with logs for distributed services[C]// *Proceedings of the IEEE International Conference on Web Services (ICWS)*. 2017: 25-32.
- [54] LI T, MA J F, PEI Q Q, et al. AClog: attack chain construction based on log correlation[C]// *Global Communications Conference*. 2019: 1-6.
- [55] MENG W B, LIU Y, ZHANG S L, et al. Device-agnostic log anomaly classification with partial labels[C]// *International Workshop on Quality of Service*. 2018: 1-6.
- [56] LIU Z L, QIN T, GUAN X H, et al. An integrated method for anomaly detection from massive system logs[J]. *IEEE Access*, 2018: 30602-30611.
- [57] XU W, HUANG L, ATREJA S, et al. Online system problem detection by mining patterns of console logs[C]// *ICDM'09*. 2009: 588-597.
- [58] NANDI A, MANDAL A, ATREJA S, et al. Anomaly detection using program control flow graph mining from execution logs[C]// *KDD 2016*. 2016: 215-224.
- [59] LIN Q W, ZHANG H Y, LOU J G, et al. Log clustering based problem identification for online service systems[C]// *ICSE 2016*. 2016: 1-10.
- [60] LIU F C, WEN Y, ZHANG D X, et al. Log2vec: a heterogeneous graph embedding based approach for detecting cyber threats within enterprise[C]// *CCS'19*. 2019: 1777-1794.
- [61] RISTO V, BERNHARDS B, MARKUS K. An unsupervised framework for detecting anomalous messages from syslog log files[C]// *Network Operations and Management Symposium*. 2018: 1-6.
- [62] LOU J G, FU Q, YANG S Q, et al. Mining invariants from console logs for system problem detection[C]// *USENIX Annual Technical Conference*. 2010: 1-14.
- [63] YUAN Y, ANU H, SHI W C, et al. Learning-based anomaly cause tracing with synthetic analysis of logs from multiple cloud service components[C]// *Computer Software and Applications Conference*. 2019: 66-71.
- [64] BIBLOP D, MOHIUDDIN S, MUHAMMADALI G, et al. LogLens: a real-time log analysis system[C]// *International Conference on Distributed Computing Systems*. 2018: 1052-1062.
- [65] DUNIA R, QIN J S. Multi-dimensional fault diagnosis using a subspace approach[C]// *ACC*. 1997: 1-5.
- [66] PAPINENI K. Why inverse document frequency? [C]// *NAACL '01*. 2001: 1-8.
- [67] ASTEKIN M, OZCAN S, SOZER H. Incremental analysis of large-scale system logs for anomaly detection[C]// *International Conference on Big Data*. 2019: 2119-2127.
- [68] ASTEKIN M, ZENGİN H, SÖZER H. Evaluation of distributed machine learning algorithms for anomaly detection from large-scale system logs: a case study[C]// *2018 IEEE International Conference on Big Data (Big Data)*. 2018: 2071-2077.
- [69] BROWN A, TUOR A, HUTCHINSON B, et al. Recurrent neural network attention mechanisms for interpretable system log anomaly detection[C]// *MLCS 2018*. 2018: 1-8.
- [70] BERTERO C, ROY M, SAUVANAUD C, et al. Experience report: log mining using natural language processing and application to anomaly detection[C]// *International Symposium on Software Reliability Engineering (2017)*. 2017: 351-360.
- [71] AMEY W, TANISHQ G, ROHIT V, et al. Hybrid CAE-VAE for unsupervised anomaly detection in log file systems[C]// *International Conference on Computing Communication and Networking Technologies*. 2019: 1-7.
- [72] YOON-HO C, PENG L, SHANG Z T, et al. Using deep learning to solve computer security challenges: a survey[J]. *arXiv: Cryptography and Security*. 2019.
- [73] STEVEN Y, MELODY M, TENG-SHENG M. CausalConvLSTM: semi-supervised log anomaly detection through sequence modeling[C]// *International Conference on Machine Learning and Applications*. 2019: 1334-1341.
- [74] DU M, LI F F, VIVEK S. DeepLog: anomaly detection and diagnosis from system logs through deep learning[C]// *CCS 2017*. 2017: 1285-1298.
- [75] XIA B, YIN J J, XU J, et al. LogGAN: a sequence-based generative adversarial network for anomaly detection based on system logs[C]// *SciSec 2019: Science of Cyber Security, Switzerland*. 2019: 61-76.
- [76] ZHANG XU, XU Y, ZHANG H Y, et al. Robust log-based anomaly detection on unstable log data[C]// *ESEC/FSE'19*. 2019: 807-817.
- [77] WANG X, WANG D, ZHANG Y, et al. Unsupervised learning for log data analysis based on behavior and attribute features[C]// *International Conference on Artificial Intelligence*. 2020: 510-518.
- [78] 梅御东, 陈旭, 孙毓忠, 等. 一种基于日志信息和CNN-text的软件系统异常检测方法[J]. *计算机学报*, 2020, 43(2): 366-380.

MEI Y D, CHEN X, SUN Y Z, et al. A method for software system anomaly detection based on log information and CNN-Text[J]. Chinese Journal of Computers, 2020, 43(2): 366-380.

- [79] LU S Y, WEI X, LI Y D, et al. Detecting anomaly in big data system logs using convolutional neural network[C]// Dependable Autonomous and Secure Computing. 2018: 151-158.

[作者简介]



张颖君（1982- ），女，山西太原人，博士，中国科学院软件研究所副研究员，主要研究方向为网络信息安全。



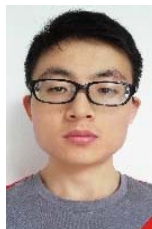
杨牧（1984- ），男，北京人，北京市公安局网络安全保卫总队工程师，主要研究方向为网络安全。



张海霞（1981- ），女，河北石家庄人，博士，中国科学院软件研究所高级工程师，主要研究方向为网络信息安全。



刘尚奇（1971- ），男，湖南桃江人，北京市公安局网络安全保卫总队高级工程师，主要研究方向为网络安全。



黄克振（1988- ），男，山东德州人，中国科学院软件研究所工程师，主要研究方向为网络信息安全。