云平台上的面向IT运营的人工智能（AIOps）：综述、机遇与挑战

钱成 *†、Doyen Sahoo *、Amrita Saha、Wenzhuo Yang、Chenghao Liu、Gerald Woo、Manpreet Singh、Silvio Saverese 和 Steven C. H. Hoi，Salesforce AI

arXiv:2304.04661v1 [cs.LG] 10 Apr 2023

摘要——面向信息技术运维的人工智能（AIOps）旨在将人工智能的强大能力与信息技术运维进程所产生的大数据相结合，尤其是在云基础设施中，以提供可操作的洞见，首要目标是最大限度地提高可用性。需要解决的问题种类繁多，也有多种应用场景，可借助人工智能能力提升运维效率。本文回顾了智能运维的愿景、趋势、挑战与机遇，并着重探讨其底层人工智能技术。我们深入讨论信息技术运维活动所产生的主要数据类型、分析这些数据的规模与挑战，以及这些数据可能发挥的作用。我们将智能运维的关键任务归纳为事件检测、故障预测、根因分析和自动化操作。我们探讨每项任务的问题表述，并介绍解决这些问题的技术分类体系。我们还指出了相对缺乏探索的主题，尤其是那些能够显著受益于人工智能领域进展的主题。此外，我们也提供了对该领域趋势的见解，并探讨了其中的关键投资机遇。

索引词—智能运维、人工智能、IT 运维、机器学习、异常检测、根因分析、故障预测、资源管理

I. 引言

在数字化转型时代，现代软件一直在快速演进。云计算、软件即服务（SaaS）、微服务、DevOps 等新型基础设施、技术和设计模式相继开发出来，以推动软件开发。管理和运维此类现代软件的基础设施如今正面临新的挑战。例如，当传统软件转变为软件即服务时，软件公司不再是将安装包交给用户，而是需要为所有基于订阅的用户提供全天候的软件访问服务。除了开发和测试之外，服务管理和运维如今也成为软件即服务公司的新职责。同时，传统的软件开发会将整个软件生命周期中的各项功能分开。编码、测试、部署和运维通常由不同的团队负责。每个团队都需要不同的技能组合。然而，敏捷开发和DevOps开始模糊各个进程之间的边界，DevOps工程师也被要求承担端到端（E2E）责任。对于 DevOps 团队而言，平衡开发与运维对整个团队的生产力至关重要。

软件服务需要向客户保证服务级别协议（SLA），并且通常会设定内部服务级别目标（SLO）。满足 SLA和 SLO 是首席信息官选择合适服务提供商的首要任务之一[1]。意外的服务中断可能会影响可用性目标，并导致严重的财务和信任问题。例如，AWS于 2021 年 12月发生了一次重大服务中断，导致多个第一方和第三方网站以及使用频繁的服务出现停机 [2]。

IT 运维在现代软件公司的成功中发挥着关键作用，因此人们引入了多个相关概念，例如专门针对软件即服务（SaaS）的 IT 服务管理（ITSM），以及面向一般 IT 基础设施的 IT 运维管理（ITOM）。这些概念关注 IT 运维的不同方面，但其底层工作流程非常相似。软件系统的生命周期可以划分为几个主要阶段，包括规划、开发/编码、构建、测试、部署、维护/运维、监控等。 [3]。DevOps 的运维部分还可以进一步分解为四个主要阶段：观测、检测、参与和行动，如图 1 所示。观测阶段包括收集不同的遥测数据（指标、日志、追踪数据等）、对收集的遥测数据进行索引和查询，以及将其可视化等任务。观测时间（TTO）是用于衡量观测阶段性能的指标。检测阶段包括检测事件、预测故障、发现相关事件等任务，其性能通常通过检测时间（TTD）以及精确率/召回率来衡量。参与阶段包括问题分诊、定位、根因分析等任务，其性能通常通过分诊时间（TTT）来衡量。行动阶段包括立即执行修复操作，例如重启服务器、扩展资源或横向扩展资源、回滚到先前版本等。解决时间（TTR）是行动阶段所衡量的关键指标。与软件开发和发布不同，持续集成和持续交付（CI/CD）流水线已经相对成熟，而许多发布后的运维工作通常仍由人工完成。这类人工运维流程面临着若干挑战：

- 手动运维难以扩展。手动运维的能力受限于DevOps团队的规模，而团队规模只能线性增长。当软件使用量处于增长阶段时，吞吐量和工作负载可能在规模和复杂性方面呈指数级增长。对于 DevOps 团队来说，很难以同样的速度增长到

* 贡献相同 † 作者任职于 Salesforce AI 期间完成的工作

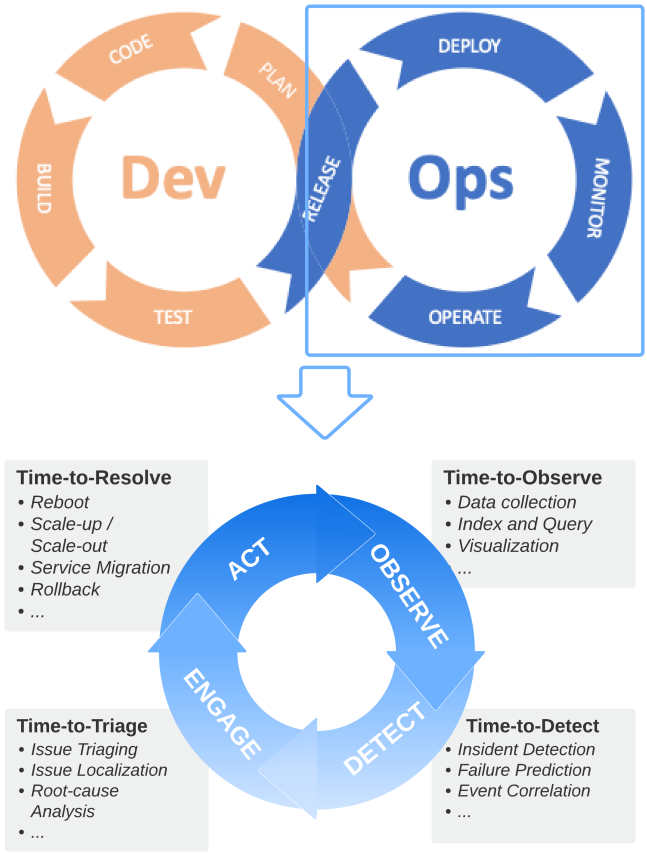

图 1. 常见的 DevOps 生命周期[3] 及运维流程分解。运维可分为四个阶段：观测、检测、分诊和处理。每个阶段都有相应的衡量指标：观测时间、检测时间、分诊时间和解决时间。

以相同的速度来处理不断增加的运营工作负载。

- 手动运维难以标准化。 由于 DevOps 团队成员背景各异（例如技能水平、对服务的熟悉程度、任职时间等），要在整个团队中始终保持同样的高标准非常困难。培养一名能够有效处理事件的运维领域专家，需要投入大量时间和精力。这些专家意外离职可能会严重损害 DevOps 团队的运维效率。

- 手动运维容易出错。 人为操作失误导致重大事件的情况十分常见。近年来，即使是最可靠的云服务提供商，也曾因人为错误而发生重大事件。

鉴于这些挑战，由人工智能能力驱动的全自动化运维流水线，是实现 SLA 和 SLO 目标的一种有前景的方法。AIOps（人工智能赋能信息技术运维的简称）一词由 Gartner 于2016年提出。根据 Gartner 术语表，“AIOps 结合大数据与机器学习，实现信息技术运维进程的自动化，包括事件相关性分析、异常检测和因果关系判定”[4]。为了实现全自动化的信息技术运维，投资 AIOps 技术势在必行。AIOps 是实现高可用性、可扩展性和运维效率的关键。例如，AIOps 可以利用人工智能模型自动分析海量遥测数据，更快速、更稳定地检测和诊断事件，帮助实现 99.99% 可用性等雄心勃勃的目标。AIOps 能够根据增长需求动态扩展自身能力，并利用人工智能自动执行事件管理和资源管理，从而减轻招聘和训练领域专家以满足增长需求的负担。此外，通过 AIOps 实现自动化有助于节省宝贵的开发人员时间，并避免疲劳。作为一项新兴的人工智能技术，AIOps 与深度强化学习、自然语言生成和通用人工智能等其他热门主题一同出现在 Gartner 人工智能技术成熟度曲线的趋势图中，时间为 2017 [5],。截至2022年，许多企业已在其信息技术基础设施中加大对 AIOps 解决方案的采用。预计到2023年底，AIOps 市场规模将达到 110.2 亿美元，年复合增长率（CAGR）为 34%。

智能运维涉及一系列复杂问题。借助智能运维从手动运维转向自动化运维，并非一蹴而就。我们根据人工智能技术的采用程度，将智能运维成熟度划分为四个不同级别，具体取决于 AIOps能力的采用情况，如图 2 所示。

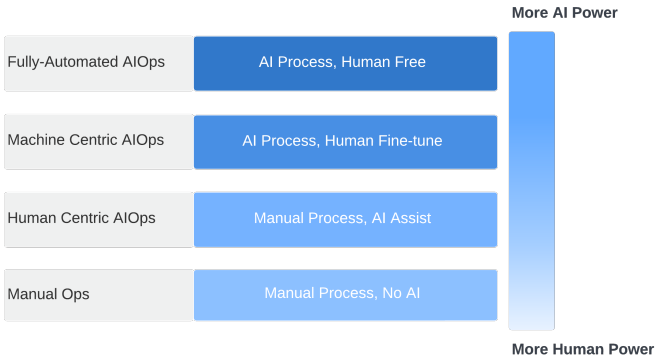

图2。智能运维转型。基于人工智能技术采用情况划分的不同成熟度级别：手动运维、以人为中心的智能运维、以机器为中心的智能运维、全自动化智能运维。

手动运维。 在这一成熟度级别，DevOps遵循传统最佳实践，所有进程均通过手动方式设置。不存在人工智能或机器学习模型。这是进行智能运维转型对比的基准。

以人为中心。在这一层级，运维主要以手动流程完成，并采用人工智能技术替代工作流中的子流程，主要充当助手。例如，DevOps 或站点可靠性工程师无需一直盯着屏幕查看事件告警，而是可以根据异常检测模型设置动态告警阈值。同样，根因分析流程需要查看多个仪表板来获取洞察，而人工智能可以帮助自动获得这些洞察。

以机器为中心。在这一层级，端到端运维流程的所有主要组件（监控、检测、参与和执行）都由更复杂的人工智能技术赋能。

人类基本无需亲自操作，但仍需参与人在回路流程，以帮助微调和改进人工智能系统的性能。例如，DevOps / 站点可靠性工程师负责运行和管理人工智能平台，以确保训练和推理流水线正常运行；领域专家则需要为人工智能做出的决策提供反馈或标签，以提升性能。

全自动化。在这一阶段，AIOps平台能够以最少或零人工干预实现全面自动化。借助全自动化的AIOps平台，现有的持续集成/持续部署（持续集成和持续部署）流水线可以进一步扩展为持续集成/持续部署/持续监控/持续纠正（持续集成、持续部署、持续监控和持续纠正）流水线。

不同的软件系统和公司可能处于不同的AIOps成熟度阶段，在采用哪些具体AIOps能力方面，其优先事项和目标也可能有所不同。设定合适的目标对AIOps应用的成功至关重要。我们预计，未来将逐渐从手动运维转向全自动化的AIOps，并运用越来越复杂的人工智能技术来解决具有挑战性的问题。为了帮助业界更快地采用AIOps能力，本文全面综述了各种AIOps问题和任务，以及业界为解决这些问题和任务而开发的方案。

二、本综述的贡献

近期，AIOps领域涌现出越来越多的研究和工业产品，以解决各种问题。Sabharwal 等人出版了《Hands-on AIOps》一书，探讨实用型AIOps及其实施[6]。此外，还有一些AIOps文献综述可供查阅 [7][8] ，帮助读者更好地理解这一领域。然而，针对将AIOps与最新人工智能技术进行深入关联并提供整体视角的研究仍然非常有限。大多数与人工智能相关的文献综述仍然基于主题展开，例如深度学习异常检测 [9] [10], 、故障管理、根因分析 [11], 等。目前，关于AIOps的整体性研究仍然有限，尚未涵盖学术界和工业界的现状。我们开展本综述旨在弥补这一空白，并更加关注AIOps中所使用的人工智能技术。

除监控阶段外，大多数任务聚焦于遥测数据的收集和管理，AIOps还涵盖其他三个阶段，这些阶段的任务更加侧重于分析。在本综述中，我们根据AIOps任务能够贡献的运维阶段对其进行分组，如图3所示。

事件检测。 事件检测任务有助于检测阶段。这些任务的目标是减少平均-检测时间（MTTD）。在我们的综述中，我们涵盖时间序列事件检测（第四节-A）、日志事件检测（第-四节-B）、追踪和多模态事件检测（第四节-C）。

故障预测。 故障预测也有助于检测阶段。故障预测的目标是在潜在问题实际发生之前预测该问题，以便提前采取措施，将影响降至最低。故障预测也有助于缩短平均检测时间（MTTD）。在本综述中，我们介绍指标故障预测（第 V-A 节）和日志故障预测（第 V-B 节）。文献中对追踪数据故障预测和多模态故障预测的研究十分有限。

根因分析。根因分析任务有助于多个运维阶段，包括分诊、处理，甚至支持更高效的长期问题修复和解决。作为对事件的即时响应，其目标是最大限度地缩短分诊时间（MTTT），同时有助于减少平均解决时间（MTTR）。此外，还可以减少人工负担。我们进一步将根因分析细分为时间序列根因分析（第 VI-B 节）、日志根因分析（第 VI-B 节）以及追踪数据与多模态根因分析（第 VI-C 节）。

自动化操作。 自动化操作有助于进入执行阶段，其主要目标是减少平均解决时间（MTTR），以及实现长期问题修复和解决。在本综述中，我们讨论了一系列自动修复（第VII-A节）、自动扩缩容（第VII-B节）和资源管理（第VII-C节）的方法。

III. 智能运维的数据

在深入探讨问题设置之前，了解可用于执行 AIOps任务的数据非常重要。现代软件系统会生成海量的可观测性指标。随着数字化转型的发展，数据量持续呈指数级增长 [12]。存储在大型非结构化数据湖系统中的数据量不断增加，使 DevOps 团队难以有效使用新信息并解决用户的问题 [13]。如今，成功的产品和平台已经能够应对监控和日志记录问题。可观测性平台（例如 Splunk、AWS CloudWatch）现已支持大规模遥测数据的发出、存储和查询。

与其他人工智能领域类似，可观测性数据对智能运维至关重要。遗憾的是，该领域的公开数据集非常有限，许多成功的智能运维研究工作都使用自有的生产数据，而这些数据通常不会公开。在本节中，我们将介绍主要的遥测数据类型，包括指标、日志、追踪数据和其他记录，并针对每种数据类型列出一系列公开数据集。

A. 指标

指标是随时间测量的数值数据，可提供系统行为的快照。指标可以表示广泛的信息，大致分为计算指标和服务指标。计算指标（例如 CPU 利用率、内存使用量、磁盘 I/O）可反映计算节点（服务器、虚拟机、Pod）的健康状态。它们在系统级别进行收集，使用 Slurm [14] 获取作业和节点的使用统计信息，并使用 Lustre 并行分布式文件系统获取 I/O 信息。服务指标（例如请求数、页面访问量、错误数）用于衡量面向客户的应用程序的服务质量和水平。此类数值数据的聚合统计

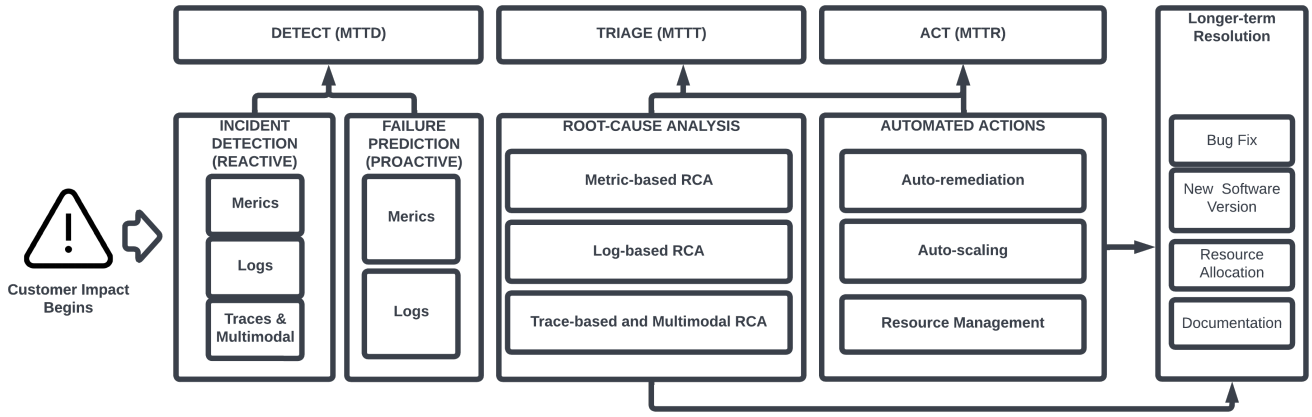

图 3. AIOps 任务。在本综述中，我们 讨论了一系列 AIOps 任务，并根据这些任务所涉及的运维阶段以及它们所采用的可观测性数据类型进行分类。

也属于指标这一类别，能够以更粗粒度的方式呈现系统行为。

云平台生命周期的所有组件都会持续生成指标，使其成为最普遍的 AIOps 数据形式之一。云平台和超级计算机集群可以生成拍字节级的指标数据，存储和分析这些数据颇具挑战，但与此同时，这些数据也为整个信息技术运维的健康状况带来了极其丰富的可观测性。指标属于数值型时间序列数据，易于解读和分析，因此可以采用简单的基于阈值的规则来采取相应操作。同时，指标包含足够丰富的信息，可用于支持更复杂的、基于人工智能的告警和操作。

利用指标数据获取洞察的主要挑战源于其多样性。指标数据可能呈现各种模式，例如周期性模式（每小时、每天、每周等重复出现的模式）、稀疏间歇性尖峰和噪声信号。指标的特征最终取决于底层服务或作业。

在表 I 中，我们简要介绍了指标数据的数据集和基准。指标数据已用于研究云数据中心的工作负载特征，以及各种 AIOps任务，例如事件检测、根因分析、故障预测，以及自动扩缩容和虚拟机预配置等各类规划与优化任务。

B. 日志

软件日志由软件开发人员专门设计，用于记录系统内运行的进程所产生的各种运行时信息，因此已成为任何现代系统或软件维护中无处不在的组成部分。系统上线后，在整个生命周期中会持续产生海量日志数据，其中自然包含大量丰富的动态运行时信息，这些信息与系统的 IT 运维和事件管理密切相关。因此，在人工智能驱动的 IT 运维流水线中，基于日志的自动化分析在事件管理中发挥着重要作用，尤其是在事件检测与因果分析、故障预测等任务中，正如已有研究所示。

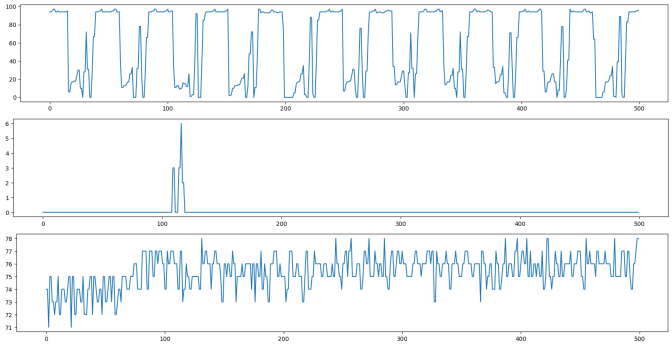

图 4. MIT Supercloud数据集中的GPU利用率指标，呈现出多种模式（周期性、稀疏且间歇性、含噪）。

过去已有多篇文献综述对此进行了研究 [15], [16], [17], [18], [19], [20], [21], [22], [23]。

在大多数实际场景中，尤其是在工业环境下，日志量每周可达到数拍字节的日志行。此外，由于日志内容的特性，日志数据转储的体积远大于时间序列遥测数据。这就需要以数据流的形式对日志可观测性数据进行特殊处理；如今，Splunk、Datadog、LogStash、NewRelic、Loggly、Logz.io 等各种服务被用于高效存储和访问日志流，并使用专门的结构化查询语言对过往日志数据进行可视化、分析和查询。

日志数据的性质。通常，这些日志由半结构化数据组成，即结构化数据与非结构化数据的组合。常见的非结构化数据类型包括自然语言词元和编程语言结构（例如方法名）；结构化部分则可以包含定量或类别遥测数据，或可观测性指标数据。这些数据由嵌入源代码中的各种日志记录语句在运行时打印，有时也会由日志记录器或日志代理自动生成。根据日志所转储自的服务类型，日志数据可能

形式和内容各异，种类繁多。例如，日志可能来自分布式系统（如 Hadoop 或 Spark）、操作系统（Windows 或 Linux ）、复杂的超级计算机系统，也可能在硬件层级（如交换机日志）、中间件层级（如服务器，例如 Apache 日志）或特定应用程序（如健康应用）中转储。通常，每条日志行都包含一个固定部分，即由开发者设计的模板，以及一个或多个可变部分或参数，用于捕获系统的某些运行时信息。

日志数据的复杂性。 因此，除了是信息技术运维中最通用、因而也最关键的数据源之一之外，日志也是最复杂的可观测性数据形式之一，因为其形式开放，且包含系统运行时信息的粒度非常精细。在云计算环境中，日志是云用户了解底层服务器的真实依据；这些服务器运行着他们的应用程序，因为云服务提供商不会向用户开放对服务器和平台的全部访问权限。此外，由于日志由开发人员设计，开发人员对源代码或日志记录语句所做的任何更改都会立即影响日志。这会导致日志记录词汇，甚至日志所依赖的整个结构或模板出现非平稳性。

日志可观测性任务。 日志可观测性通常涉及不同任务，例如在事件检测期间对日志进行异常检测（第 IV-B 节）、基于日志的根因分析（第 VI-B 节）以及基于日志的故障预测（第 V-B 节）。

数据集与基准。 在不同的日志可观测性任务中，基于日志的异常检测是最客观的任务之一，因此，大多数公开发布的基准数据集都围绕异常检测设计。表 B 全面介绍了文献中用于异常检测任务的不同公开基准数据集。其中，Switch 数据集以及 HPC 和 BGL 的子集也经过重新设计，用于故障预测任务。另一方面，目前没有用于基于日志的根因分析任务的公开基准，这类任务通常使用企业私有数据进行评估。

```
public void NGPStatsMsgHandler(){
                    ...         ...         ...
   logger.warning("Invalid NGPQueue::addTask L2Stats mQueue
       size = {} Remaining capacity is {} {}", queue_size,
       rem_capacity, address)
                    ...         ...         ...
   logger.info("EventActions {} Executing alarm rules for
       Alerts {}, Event Id {}PNET Event Id (), First time
       {}",thread_id, alert_value, event_id, first_time)
```

⇩

```
WARNING 08/09 17:35:19 [NGPStatsMsgHandler] Invalid
NGPQueue::addTask L2Stats mQueue size = 0 Remaining
capacity is 150000 null

INFO 08/09 17:35:19 [NGPStatsMsgHandler]
EventActions[Thread-956703] 302704 Executing alarm rules
for Alerts (502640,163770,null): Event Id
(mwcontrolm@172.25.153.37:3181.1626882403.6807769), PNet
Event Id (), First time (21/08/09 17:51:14)

INFO 08/09 17:35:18 [NGPStatsMsgHandler]
EventActions[Thread-956704] 302704 Executing alarm rules
for Alerts (502640,163770,null): Event Id
(mwcontrolm@172.25.153.37:3181.1626882403.6807770), PNet
Event Id (), First time (21/08/09 17:51:14)

INFO 08/09 17:35:17 [NGPStatsMsgHandler]
EventActions[Thread-956703] 302704 Executing alarm rules
for Alerts (502640,163771,null): Event Id
(mwcontrolm@172.25.153.37:3181.1626882403.6807769), PNet
Event Id (), First time (21/08/09 17:51:14)
```

图 5. 信息技术运维中生成的日志数据示例

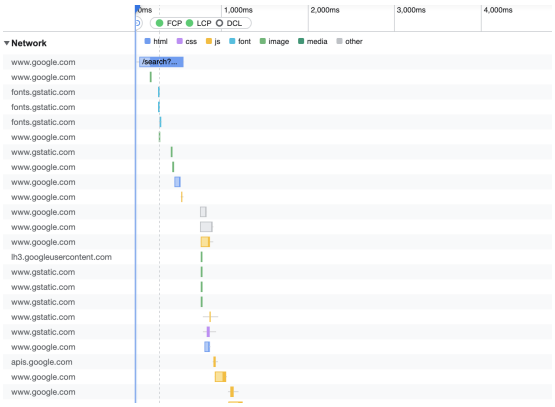

图 6. 使用 Google 搜索时用户请求追踪图的快照。

C. 追踪数据

追踪数据通常以半结构化日志的形式呈现，其中包含用于重建应用程序拓扑图和目标请求网络流的标识符。例如，当用户使用 Google 搜索时，该用户请求的典型追踪图如图 6 所示。追踪由系统事件（跨度）组成，用于跟踪请求或执行的整个过程。一个跨度是一系列半结构化事件日志。追踪数据使我们能够将不同的数据模态置于同一上下文中。请求会经过多个服务／应用程序，而每个应用程序的行为可能截然不同。追踪记录通常包含两个必需部分：时间戳和跨度 ID。利用时间戳和跨度 ID，我们可以轻松地从追踪日志中重建追踪图。

追踪分析需要可靠的追踪系统。ReTrace [24] 等追踪收集系统有助于实现快速、低成本的追踪数据收集。追踪收集器通常与代码无关，并且能够近乎实时地将不同级别的性能追踪数据发送回追踪存储库。追踪收集过程中还会进行早期摘要，以帮助生成细粒度事件 [25]。

尽管追踪收集在系统可观测性中很常见，但要获取高质量的追踪数据来训练人工智能模型仍然颇具挑战。据我们所知，带有高质量标签的公开追踪数据集非常少。此外，现有的少数公开追踪数据集（如 [26] ）也未在智能运维研究中得到广泛采用。相反，大多数与智能运维相关的追踪分析研究使用自有的生产或模拟追踪数据，

通常不向公众公开。

D. 其他数据

除了机器生成的可观测性数据（如指标、日志、追踪数据等）之外，还有其他类型的运维数据可用于智能运维。

人类活动记录是这些有价值数据的一部分。工单系统用于DevOps/站点可靠性工程师之间进行沟通并高效解决问题。该进程会生成大量的人类活动记录。人类活动数据包含有关现有问题解决方案的丰富知识和经验，可用于在未来解决类似问题。

用户反馈数据对于提升智能运维系统的性能也非常重要。与问题工单不同，用户需要在工单中提供大量上下文信息来描述和讨论问题，而用户反馈可以简单到只需点击一下，确认告警是好是坏。收集运行中系统的实时用户反馈，并设计人在回路中的工作流，对于智能运维解决方案的成功也至关重要。

尽管许多公司会收集这些类型的数据，并利用它们改进运维工作流程，但目前公开发表的研究中，探讨如何将这些其他类型的运维数据系统地纳入智能运维解决方案的内容仍然非常有限。这既带来了挑战，也为进一步改进智能运维领域提供了机遇。

接下来，我们将讨论智能运维的关键任务——事件检测、故障预测、根因分析和自动化操作，并系统地回顾这些领域文献中的主要贡献。

IV. 事件检测

事件检测采用各种异常检测技术。异常检测旨在检测异常、离群值，或通常来说检测不正常的事件。在智能运维语境下，异常检测被广泛用于检测各种类型的异常系统行为。为了检测此类异常，检测器需要利用不同的遥测数据，例如指标、日志、追踪数据。因此，异常检测还可以进一步细分为处理一种或多种特定遥测数据源，包括指标异常检测、日志异常检测、追踪异常检测。此外，如果检测过程中涉及多个遥测数据源，则可以采用多模态异常检测技术。近年来，基于深度学习的异常检测技术 [9]也得到了广泛讨论，并可用于智能运维中的异常检测。区分异常检测技术的另一种方式，是根据不同的应用程序使用场景进行区分，例如检测服务运行状况问题、检测网络问题、检测安全问题、检测欺诈交易等。通常，这些不同的技术都源自同一组基础检测算法，并经过本地化处理以应对特定任务。从技术角度来看，从不同遥测数据源中检测异常，更符合人工智能技术的定义，例如，指标通常是时间序列，日志是文本/自然语言，追踪数据是事件序列/图等。在本文中，我们将讨论通过不同遥测数据源进行的异常检测。

A. 基于指标的事件检测

问题定义

为确保服务的可靠性，数十亿个指标都在相等的时间间隔内持续受到监控和采集，时间戳为[27]。因此，将指标整理为时间序列数据以便后续分析十分直接。基于指标的事件检测旨在找出监控指标中显著偏离其他观测值的异常行为，这对运维人员及时检测软件故障并触发故障诊断、从而减少损失至关重要。针对指标进行事件检测最基本的方式是基于规则的方法，即当指标超过某个阈值时发出警报。这种方法只能捕获由指标超过阈值所定义的事件，无法检测更复杂的事件。基于规则的指标事件检测方法通常过于简单，只能处理最简单的事件。此外，这些方法对阈值十分敏感：阈值过低会产生过多误报，阈值过高则会导致漏报。由于事件具有开放式的特征，系统架构日益复杂，系统规模和指标数量不断增长，人工监控和基于规则的方法已不再足够。因此，迫切需要利用人工智能能力来实现更先进的基于指标的事件检测方法。

由于指标属于时间序列数据的一种形式，而事件表现为数据中的异常情况，因此指标事件检测通常被表述为时间序列异常检测问题 [28], [29], [30]。下文聚焦于智能运维设置，并基于几个关键标准对其进行分类：(i)学习范式、(ii)维度、(iii)系统，以及 (iv)流式更新。我们进一步总结了一系列时间序列异常检测方法，并在表 IV 中根据这些标准进行了比较。

学习设置

a) 标签可获取性： 一种自然的异常检测问题表述方式是将其视为有监督二分类问题，以判断给定的观测是否为异常 [31], [32]。这样的表述方式的优势在于可以应用任何有监督学习方法，而这类方法在过去几十年中已得到深入研究 [33]。然而，由于难以获取用于指标事件检测的带标签数据 [34]，且异常标签容易出错 [35]，，通常更倾向于采用不需要标签来构建异常检测器的无监督方法，而且这类方法也更为普遍。尤其是，无监督异常检测方法大致可分为基于密度的方法、基于聚类的方法和基于重构的方法 [28], [29], [30]。基于密度的方法会计算局部密度和局部连通性，以进行离群点判定。基于聚类的方法将异常分数定义为到聚类中心的距离。基于重构的方法会显式地对生成过程进行建模。

数据，并使用重构误差衡量异常分数。尽管指标异常检测中的方法通常是无监督的，但也存在可以获取部分标签的情况。在此类情形下，半监督、领域自适应和主动学习范式便发挥作用。半监督范式 [36], [37], [38] 使无监督模型能够利用稀疏可用的正标签信息 [39]。领域自适应 [40] 依赖于带标签的源数据集，而目标数据集未带标签，旨在将基于源数据集训练的模型迁移到目标数据集上，以执行异常检测。

b) 流式更新： 由于每分钟都会收集大量指标，模型会在线用于检测异常。指标的时间模式随时间推移而发生变化是非常常见的。在接收到新的输入数据时及时执行模型更新的能力是一项重要标准。一方面，传统模型可以通过定期重新训练整个模型来处理数据流 [31], [41], [32], [38]。然而，这种策略的计算成本可能很高，并会带来其他不容忽视的问题，例如应该多久执行一次重新训练。另一方面，一些方法 [42], [43]内置了高效的更新机制，能够自然地适应这些新的输入数据流。它还支持主动学习范式 [41],，允许模型针对不确定的数据点与用户进行交互式查询，以获取标签，随后使用新标签更新模型。

c) 维度：每个监控数据指标形成一个单变量时间序列，因此一个服务通常包含多个指标，每个指标描述复杂实体的不同部分或属性，共同构成一个多变量时间序列。传统解决方案是针对每个指标构建单变量时间序列异常检测。然而，对于复杂系统而言，这种方法忽略了各指标之间的内在交互，无法很好地表示系统整体状态。简单地组合每个单变量时间序列的异常检测结果，对于多变量异常检测方法而言表现不佳 [44], ，因为它无法对服务的各指标之间的相互依赖关系进行建模。

模型广泛的机器学习模型可用于时间序列异常检测，大致分为深度学习模型、基于树的模型和统计模型。深度学习模型 [45], [36], [46], [47], [38], [48],[49], [50]利用深度神经网络的成功和强大能力来学习时间序列数据的表示。这些时间序列数据的表示包含底层指标丰富的语义信息，并可用作一种基于重构的无监督方法。基于树的方法利用树结构作为一种基于密度的无监督方法 [42]。统计模型 [51]依赖经典的统计检验，这些检验被视为一种基于重构的方法。

工业实践 构建能够处理真实云 IT 运维中产生的大量指标数据的系统，往往是一项挑战。这是因为现实场景中的指标数据种类繁多，而且异常的定义可能因场景而异。此外，几乎所有时间序列异常检测系统都需要以低延迟 [32]并行处理大量指标。因此，这里重点介绍了提出基础设施处理系统的相关工作。EGADS [41] 是雅虎开发的系统，每秒可扩展处理数百万个数据点，重点是优化实时处理。它由批量时间序列建模模块、在线异常检测模块和告警模块组成。该系统采用多种无监督方法进行异常检测，并提供可选的主动学习组件来过滤告警。 [52] 是微软开发的系统，包含数据摄取、实验和在线计算平台三大主要组件。该系统提出了一种高效的深度学习异常检测器，能够同时实现高准确率和高效率。 [32] 是阿里巴巴集团开发的系统，由数据摄取、离线训练、在线服务、可视化和告警模块组成。该系统通过时间序列分解提出了一种稳健的异常检测器，因此能够轻松处理具有不同特征的时间序列，例如季节长度不同、趋势类型不同等。 [38] 是腾讯开发的系统，由离线模型训练组件和在线服务组件组成，并利用主动学习，通过少量不确定样本更新在线模型。

挑战

缺少标签指标异常检测的主要挑战是缺少真实异常标签 [53], [44]。由于服务器架构中事件的开放性和复杂性，很难定义什么是异常。因此，构建带标签数据集是一项极其耗费人力和资源的工作，需要领域专家付出努力，从时间序列数据中识别异常。此外，人工标注可能导致标注错误，因为没有统一且正式的异常定义，从而使对真实标签的判断带有主观性 [35]。

实时推理典型的云基础设施每秒可收集数百万个数据点，因此需要近乎实时地进行推理以检测异常。指标异常检测系统需要具备可扩展性和高效率， [54], [53], 并可选择支持模型重新训练，这会带来巨大的计算、内存和 I/O 负载。随着深度学习方法日益普及，异常检测模型也愈发复杂， [55]这些更大型模型带来的额外计算成本进一步加重了系统负担。

指标流的非平稳性指标数据流的时间模式通常会随时间变化，因为它们是在非平稳环境中生成的 [56]。这些模式的演变通常由不可观测的外部因素引起。例如，某项服务日益普及，会导致客户指标（如请求数）随时间不断上升。忽略这些因素会导致异常检测器的性能下降。一种解决方案是使用近期数据持续更新模型， [57],但这一策略需要谨慎权衡更新频率与成本和模型鲁棒性之间的关系。

公开基准虽然已有针对通用异常检测方法和时间序列异常检测方法的基准 [33], [58], 但AIOps领域的指标事件检测仍缺乏基准测试。鉴于此。

由于时间序列数据具有广泛且多样的特性，它们通常会呈现出取决于具体领域的多种不同类型的异常，这使得理解算法的优缺点变得颇具挑战性 [58]。此外，现有数据集一直被批评过于简单且标注错误 [59]。

未来趋势

主动学习/人在回路为了解决缺少标签的问题，一种更智能的方式是以最低成本整合人类知识与经验。人类作为特殊的智能体，拥有丰富的先验知识 [60]。如果事件检测框架能够促使机器学习模型学习运维专家的智慧与知识，就有助于应对标签稀缺和噪声标签问题。在 [38]中，利用主动学习更新在线模型，就是在标注任务中引入人工投入的典型案例。在其他数据处理步骤中引入人工投入也有很大的研究空间，例如特征提取。此外，人工投入还可以整合到机器学习模型的训练和推理阶段。

流式更新 由于指标流具有非平稳性，持续更新异常检测器至关重要。随着模型日益复杂且对成本效益的要求不断提高，我们将看到更多采用内置高效流式更新能力的方法。深度学习方法在时间序列异常检测任务中取得了巨大成功 [30]。在线深度学习正成为一个越来越受欢迎的研究主题 [61], , 近期我们或许会看到相关技术迁移至时间序列指标异常检测领域。

内在异常检测当前关于时间序列异常检测的研究工作并不区分异常的原因或类型，而这对于智能运维中的后续缓解步骤至关重要。例如，即使异常被成功检测出来，如果该异常是由外部环境导致的，操作员也无法减轻其负面影响。在 [50], [48], 中引入的内在异常检测考虑了受监控指标与环境之间的函数依赖结构。这种设置考虑了环境中的变化，并可能利用在常规（外部）设置中无法获得的信息。例如，当扩展/缩减为应用程序提供服务的资源（可能是由于自动扩缩容规则）时，我们会观察到 CPU指标下降/上升。虽然这在外部设置中可能被视为异常，但事实上它并不是一个事件，因此在内在设置中也不属于异常。

B. 基于日志的事件检测

问题定义

软件和系统日志数据是记录和跟踪系统内所有正在进行的进程的运行时信息（可达到任意粒度级别）最常用的方式之一。总体而言，一个大型分布式系统可能会产生海量异构日志，这些日志由其不同的服务或微服务输出，每条日志都带有时间戳。符合其自身非结构化、半结构化或结构化格式的文本消息。在各种 IT 运维中，可靠性与性能工程师以及核心开发人员广泛使用这些日志，以了解系统的内部状态，并促进监控、管理和故障排查 [15], [16], [17], [18], [19], [20], [21], [22], [62]。具体而言，在智能运维流水线中，日志分析能够承担的首要任务之一是基于日志的事件检测。通常通过对日志进行异常检测来实现这一目标，旨在从系统生成的海量软件日志数据转储中，检测出表明可能发生事件的异常日志行或日志行序列。基于日志的异常检测通常在根据关键绩效指标监控检测到事件后应用，作为更细粒度的事件检测或故障诊断步骤，以确定系统执行过程中哪个服务、微服务或软件模块的行为出现异常。

任务复杂度

日志异常模式的多样性：智能运维中存在非常多样的事件，这些事件可能导致日志数据中出现不同类型的异常模式——它们可能表现为日志模板（即日志行的固定部分）中的异常，也可能表现为日志参数（即包含动态信息的日志行可变部分）中的异常。这些模式包括：i）关键词——日志行中出现带有系统故障、事件或异常领域特定语义的关键词（例如内存不足或崩溃）；ii）模板计数——日志模板或日志事件类型的数量突然增加或减少，表明存在异常；iii）模板序列——任务执行的正常顺序出现显著偏离，表明存在异常；iv）变量值——与某些日志模板或事件相关联的变量可能具有物理意义（例如时间成本），可以将其提取出来并聚合为结构化的时间序列，在该时间序列上应用标准异常检测技术；v）变量分布——对于某些分类变量或数值变量，变量分布偏离标准分布可能表明存在异常；vi）时间间隔——某些性能问题可能不会直接体现在日志行本身，而是体现在特定日志事件之间的时间间隔中。

对人工智能的需求：鉴于日志规模极其庞大，即使是领域专家，也往往难以手动检查日志以检测异常日志行。此外，如上所述，具体取决于事件的性质，日志中可能存在多种不同类型的异常模式，这些模式可能表现为日志模板中的异常关键词（例如“错误”或“异常”）、特定事件日志的数量、变量分布，或两个特定事件日志之间的时间间隔。然而，即使是领域专家，也无法制定规则来检测这些异常模式；即便能够制定，这些规则也很可能无法适应不同类型的事件，以及随着软件功能随时间变化而改变的日志行。因此，这充分说明了.....的必要性。

采用数据驱动模型和机器智能挖掘并分析这一复杂数据源，以实现事件检测的最终目标。

用于事件检测的日志分析工作流

为了处理数据的复杂性，通常需要遵循一系列步骤，以有意义地分析日志并检测事件。从原始日志数据或数据流开始，日志分析工作流首先会对日志进行一些预处理，使其适用于机器学习模型。接下来通常是日志解析，从半结构化数据中提取出一种松散的结构，然后对日志行进行分组和划分，将其组织成日志序列，以便对数据的序列特征进行建模。此后，日志或日志序列会被表示为机器可读的矩阵，在该矩阵上可以执行各种日志分析任务——例如聚类和汇总海量日志转储，将其归纳为少量关键日志模式，以便进行可视化，或检测可能表明某个事件的异常日志模式。图 7 概述了日志分析工作流中的不同步骤。其中一些步骤更多属于工程挑战，另一些则更多由人工智能驱动，还有一些会结合使用机器学习与领域知识规则。

i) 日志预处理：此步骤通常包括根据实际日志分析需求，自定义过滤被认为无关的特定正则表达式模式（如 IP 地址或内存位置）。其他预处理步骤（如分词）则需要针对不同的措辞风格进行专门处理，因为日志具有混合特性，既包含自然语言，也包含编程语言结构。例如，日志行可能混合包含源代码数据中的文本字符串，其中既有蛇形命名法和驼峰命名法标记，也有自然语言中的空格分隔标记。

ii）日志解析：为了支持下游处理，非结构化日志消息首先需要被解析为结构化的事件模板（即由开发人员实际设计的常量部分）和参数（即包含动态运行时信息的变量部分）。图 8 给出了对单条日志行进行解析的示例。文献中既有用于日志解析的启发式方法，也有人工智能驱动的方法，后者包括传统机器学习方法以及较新的神经网络模型。Drain [63]、IPLoM [64]和 AEL [65]等启发式方法利用已知的日志结构归纳偏置，而 Spell [66] 则使用最长公共子序列算法动态提取日志模式。在这些方法中，Drain 和 Spell 最为流行，因为它们能够良好地扩展到工业级应用。传统机器学习方法包括：i）基于聚类的方法，如 LogCluster [67]、LKE [68]、LogSig [69]、SHISO [70]、LenMa [71], 和 LogMine[72]，这些方法假设日志消息类型会在相似的组中对应；ii）频繁模式挖掘和项集挖掘方法，如 SLCT [73], 和 LFA [74]，用于提取常见的消息类型；iii）进化优化方法，如 MoLFI [75]。另一方面，较新的神经网络方法包括 [76] - 神经网络基于 Transformer 的模型，这些模型使用自监督掩码语言建模来学习日志解析；vii）UniParser [77] - 一种面向异构日志数据的统一解析器，配备可学习的相似度模块，以泛化到跨不同系统的多样化日志。还有另一类日志分析方法[78], [79]，旨在采用免解析技术，从而避免解析带来的计算开销，以及由错误解析导致的错误级联，尤其是由于解析方法缺乏鲁棒性所造成的问题。

iii) 日志分区：解析后，下一步是根据某种语义将日志数据划分为多个组，每个组代表一段有限的日志行或日志序列。这样做的主要目的是将通常包含数百万条日志行的原始日志转储分解为逻辑块，以便对这些块进行显式建模，并使模型能够捕捉日志模板序列、日志参数值或两者中的异常模式。日志分区有多种类型 [20],[80] ——固定窗口或基于滑动窗口的分区（窗口长度由日志序列的长度或时间段决定），以及基于标识符的分区（根据某个标识符对日志进行分区，例如日志来源的会话或进程）。图 9 展示了日志分组和分区的这些不同方式。最终，日志事件是否异常，可以在日志行或日志分区层面进行判定。

iv) 日志表示： 在完成日志分区后，下一步是通过从各分区中提取特征，以机器可读的方式（例如向量或矩阵）表示每个分区。这可以通过多种方式实现 [81], [80]——既可以利用领域知识提取特定的手工设计特征，也可以采用 ii) 序列表示，将每个分区转换为按顺序排列的日志事件 ID；还可以采用 ii) 定量表示，使用由日志事件的词项频率和逆文档频率信息加权的计数向量；iii) 语义表示则从日志事件中的语言词元序列中捕捉语言学含义，并为数据集中的每个词元学习一个高维嵌入向量。所选择的日志表示性质会直接影响其能够支持的异常模式类型——例如，对于捕捉基于关键词的异常而言，语义表示可能至关重要；而对于与模板计数和变量分布相关的异常，定量表示可能更为合适。语义嵌入向量本身既可以使用 GloVe、FastText 等预训练神经语言模型，或 BERT、RoBERTa 等预训练 Transformer 获得，也可以将其作为目标任务的一部分，通过可训练嵌入层进行学习。

v) 用于事件检测的日志分析任务：一旦将日志转换为某种紧凑、机器可读且便于人工智能模型处理的形式，就可以对其执行一系列日志分析任务：首先使用聚类和摘要技术进行日志压缩，然后进行基于日志的异常检测。进而，异常检测还可以支持事件管理中的下游任务，

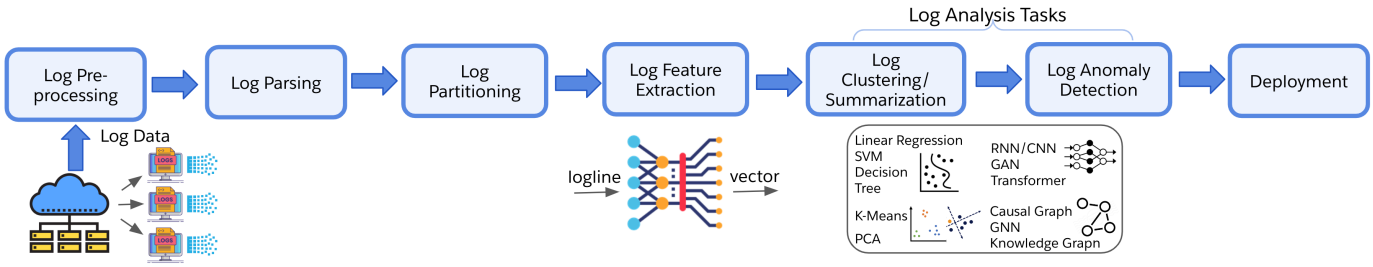

图 7. 事件检测日志分析工作流的步骤

LogLine: 081109 204655 556 INFO dfs.DataNode$PacketResponder: Received block blk_3587508140051953248 of size 67108864 from /10.251.42.84

Timestamp: 081109 204655 556
Level: INFO
Component: dfs.DataNode$PacketResponder
Template: Received block <*> of size <*> from <*>
Parameter: ["blk_3587508140051953248", "67108864", "10.251.42.84"]

图 8. 日志解析示例

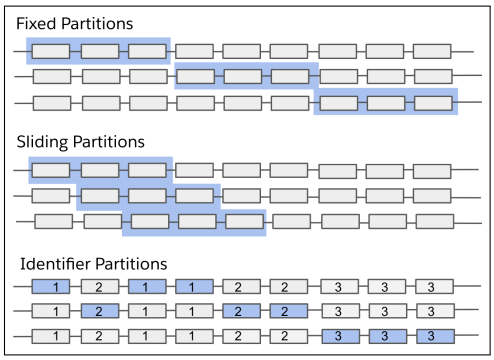

图 9. 不同类型的日志分区

例如故障预测和根因分析。本节仅讨论与事件检测相关的前两项日志分析任务，并将故障预测和根因分析留到后续章节讨论。

v.1) 通过聚类和摘要实现日志压缩：这是迈向分析海量日志数据的一个切实可行的第一步，即通过各种聚类和摘要技术实现日志压缩。这项分析有两个目的：首先，在事件管理期间，这一步可以通过提供一种直观实用的方式来可视化这些规模庞大、结构复杂的原始非结构化日志数据，独立地帮助站点可靠性工程师和服务所有者。其次，日志聚类的输出可以直接用于某些基于日志的异常检测方法。

在各种日志聚类技术中， [82],[67], [83] 采用层次聚类，并通过构建和检索代表性日志簇知识库来支持在线设置。 [84], [85] 采用频繁模式匹配，并结合 PCA 和局部敏感哈希等降维技术，支持在线和流式处理。 [86], [64], [87] 采用高效的迭代式或增量式支持在线日志和流式处理日志的聚类与分区技术，同时也能够处理罕见日志实例的聚类。现有文献的另一个领域 [88], [89], [90],[91] 关注通过摘要进行日志压缩——例如， [88] 使用日志事件ID和时间等启发式方法进行摘要，而 [89], [21] 则基于开放信息抽取，利用语义信息、领域知识和规则进行三元组提取，以生成摘要；与此同时， [90], [91] 使用基于语言规则的序列聚类，或通过分组常见事件序列来进行聚类。

v.2) 日志异常检测： 日志分析最常见的用途或许是基于日志的异常检测，研究和工业环境中均采用了各种各样的模型。这些模型根据多种因素进行分类：i）学习设置——有监督、半监督或无监督：半监督模型假设对标签具有部分知识，或可访问少量异常实例；无监督模型则使用正常日志数据进行训练，并根据其预测置信度检测异常。ii）模型类型——神经模型或传统的统计非神经模型；iii）所使用的日志表示类型；iv）是否使用日志解析或无解析器方法；v）如果使用解析，则是仅编码日志模板部分，还是同时编码模板和参数表示；iv）是否将异常建模限制在单条日志行层面，还是支持对日志序列上的异常检测进行序列建模。

所采用的日志表示形式和使用的建模类型——这两个因素都会影响能够检测到的异常模式类型。例如，基于关键词和变量值的异常可通过对日志行进行语义表示来捕获，而基于模板计数和变量分布的异常模式，则更明确地通过对日志事件进行定量表示来建模。同样，基于模板序列和时间间隔的异常需要使用能够处理日志序列的序列建模算法。

下面我们简要总结专门研究这两类模型——统计模型和神经模型的文献；表 III 则对更完整的现有异常检测算法和系统进行了比较。

统计模型是更为传统的机器学习模型，它们根据训练数据背后的各种统计数据进行推理。在相关文献中，已有各种统计机器学习模型被用于此任务，依据

不同的训练设置。在有监督方法中，[92], [93], [94]采用传统学习策略，包括线性回归、SVM、决策树、孤立森林，以及从整个日志行中提取的手工设计特征。大多数此类模型在单条日志行的层面上对数据进行建模，无法显式捕获序列级异常。还有无监督方法，例如ii）降维技术，如主成分分析（PCA） [84]iii）聚类，并像 [67], [82], [95], [80]中那样分析日志事件与度量数据之间的相关性。还有无监督模式挖掘方法，包括从奇异值分解中挖掘不变模式 [96] ，以及从执行流和控制流图中挖掘频繁模式 [97], [98], [99], [68]。除此之外，还有一些系统采用基于领域知识构建的规则引擎，并结合不同机器学习模型的集成，以应对不同类型的事件 [20]；同时还使用启发式方法，对正常日志与表明异常的事件日志进行对比分析 [100]。

另一方面，神经模型是较新的一类机器学习模型，它们使用人工神经网络，并已在众多人工智能应用程序中取得了显著成功。它们在编码和表示底层复杂语义方面尤其强大，能够以对预测任务有意义的方式进行编码和表示。一类无监督神经模型使用基于重构的自监督技术来学习词元级或日志行级表示，其中包括 i) 自编码器模型 [101], [102] ii) 更强大的基于自注意力的 Transformer 模型 [103] iv)特定的预训练Transformer，例如 BERT 语言模型[104], [105], [21]。基于重构的模型的另一分支是使用生成对抗或 GAN 训练范式的模型，例如 [106], [107]使用基于 LSTM 或 Transformer 的编码。其他类型的无监督模型基于预测，它们以自监督的方式学习预测下一个日志词元或下一条日志行——例如 i) 基于循环神经网络的模型，如LSTM [108], [109], [110], [18], [111] 和 GRU [104] 或其基于注意力机制的对应模型 [81], [112], [113] ii) 基于卷积神经网络（CNN）的模型 [114] 或使用图神经网络表示日志事件数据的更复杂模型[115], [116]。基于重构和预测的模型都能够处理序列级异常，这取决于训练的性质（即表示是在日志行级还是词元级进行学习）以及模型处理长序列的能力（例如，在上述模型中，自编码器模型是最基础的模型）。

其中大多数模型遵循无监督训练的实际设置，即仅使用非异常日志数据进行训练。然而，其他研究也重点关注了在异常和正常标注数据 [111], [114], [78], [117], 上对长短期记忆网络（LSTM）、卷积神经网络（CNN）和Transformer模型进行有监督训练。另一方面， [104], [110] 基于启发式方法的假设采用弱监督，例如，来自外部系统的日志被视为异常。大多数神经模型使用语义令牌表示，其中一些使用预训练的固定嵌入或可训练嵌入，并以GloVe、FastText或基于预训练Transformer的模型（如BERT、GPT、XLM等）进行初始化。

vi) 日志模型部署： 日志分析工作流的最后一步是在实际工业环境中部署这些模型。它包括：i)训练步骤，通常基于离线日志数据转储，并可选择是否使用由领域专家收集的有监督标签；ii)在线推理步骤，这通常需要应对诸如非平稳流数据等实际挑战，即数据分布在整个时间段内并非独立同分布。为解决这一问题，一些较为传统的统计方法（如 [103], [95], [82], [84] ）支持在线流式更新，而另一些工作还可以通过增量式构建知识库、内存或域外词汇表来适应不断演变的日志数据 [101]。另一方面，大多数无监督模型支持分批进行的在线训练，使模型能够持续适应不断变化的数据分布，并部署于高吞吐量流数据源上。然而，对于一些更先进的神经网络模型而言，即使是常规的分批更新，在线更新的计算成本也可能过高。

除此之外，也有一些研究专门探讨了实际场景中与模型部署相关的其他挑战，例如在半监督设置下，仅利用源系统提供的监督信息，在来自不同领域或应用程序的日志之间进行迁移学习 [110], [103], [18], [18], [118] 。其他研究则着重评估模型的鲁棒性和泛化能力（即模型适应不稳定日志数据的能力），因为在软件演进和更新过程中，日志记录会持续发生变化 [109], [111], [104]。这些研究通过在训练 [18], [18]过程中采用领域对抗范式，或使用反事实解释 [118] ，或针对各种日志分析任务采用多任务设置 [21] 来实现上述目标。

挑战与未来趋势

收集监督标签：与大多数 AIOps任务一样，为日志分析问题的训练甚至评估收集大规模监督标签极具挑战性，也不切实际，因为这需要大量人工干预和领域知识。对于日志异常检测，其目标相当客观，因此仍有可能收集标签，以进行可靠的评估。相比之下，对于聚类和摘要等其他日志分析任务，由于目标相当主观，通常甚至无法从领域专家那里收集监督标签，因此这类任务通常通过下游日志分析或根因分析任务进行评估。

类别不平衡问题：异常检测任务面临的主要挑战之一是类别不平衡，这是因为异常数据本身极为罕见。此外，由于上述各类异常的性质各异，不同系统可能会呈现不同类型的数据偏斜。这既给使用高度偏斜的数据进行模型训练带来了技术挑战，也使评估指标的选择变得困难，因为精确率、召回率和 F 分数可能无法令人满意。此外，在推理阶段，对异常分数设置阈值对无监督模型而言尤其具有挑

战性。尽管在基准测试中，AUROC（ROC 曲线下面积）等评估指标或许足以满足要求，但要在实际部署这些模型时，仍需谨慎校准异常分数、手动调优，或通过启发式方法设定阈值。由于这对具体应用程序非常敏感，因此在将模型推广到来自不同系统的异构日志时，也会带来实际挑战。

处理海量数据：日志分析任务的另一项挑战是处理巨量日志。大多数大型云系统每天或每周都可能产生数拍字节的日志。因此，日志处理算法不仅要有效，还必须足够轻量，才能实现快速、高效的处理。

处理非平稳日志数据：除了庞大的数据量，日志分析最自然且最实用的场景是在线流式场景，其中数据分布并非平稳：来自不同互连微服务的异构日志流，以及软件日志数据本身，都会随着时间推移而不断演变，因为开发人员在敏捷云开发环境中会持续改进软件。这要求学习算法采用高效的在线更新方案，并投入专门工作来构建鲁棒模型，评估这些模型面对不稳定或不断变化的日志数据时的鲁棒性。

处理噪声数据：即使是领域专家，也很难为日志数据添加标注；有监督和半监督模型在训练过程中需要处理这些噪声，而对于无监督模型，噪声则可能严重误导评估。尽管受影响的日志只占一小部分，极端的类别不平衡仍会加剧这一问题。另一项相关挑战是，在执行异常检测等下游任务时，日志分析工作流中的各个处理步骤所产生的错误会不断累积并层层传播。

用于异常检测的真实公开基准数据集：在公开可用的日志异常检测数据集中，只有少数数据集包含异常标签。大多数这类基准已在文献中被过度使用，因此进一步推动研究的空间不大。事实上，它们最大的局限在于，无法展现真实部署中通常会出现的各种事件。通常，十分简单的人工制定规则就能成功解决这些数据集上的异常检测任务。此外，这些数据集的原始规模比真实世界用例小好几个数量级，因此不适合展示在线或流式处理场景中的挑战。此外，经过典型的日志处理步骤以移除数据中的无关模式后，独特模式的数量会大幅减少。另一方面，绝大多数文献都以内部专有数据上的实证分析和评估为依据，无法保证可复现性。因此，我们需要更贴近现实的公开基准数据集，以揭示真实环境中的 AIOps 所面临的挑战，并对当前的日志分析模型进行公平的基准测试。

用于解析、聚类和摘要的公开基准：大多数日志解析、聚类和摘要领域的研究，只使用部分公开日志数据集中的极小子集（这些数据集提供预言机解析），或者使用来自工业应用的内部日志数据集，并将模型与实际上无法扩展的预言机解析方法进行比较。这也使得这些任务难以进行公平比较和标准化基准测试。

更优的日志语言模型：近期神经自然语言处理模型（如基于Transformer 的语言模型 BERT、GPT）取得的一些进展，为以自然语言风格表示日志并实现各种日志分析任务展现出了相当大的潜力。然而，在不依赖外部解析器的情况下，构建能够适当编码由固定模板和变量参数组成的半结构化日志的神经语言模型，仍有很大的改进空间。

融入领域知识：虽然现有的日志异常检测系统完全基于规则或自动化，但鉴于事件的复杂性以及异常的多样性，更为实用的方法是将领域知识以静态形式或动态形式融入这些模型中，并遵循人在回路中的反馈机制。例如，在生成海量日志的复杂系统中，哪些类型的事件更为严重，以及针对哪类事件，哪些类型的日志对于监控更为关键。甚至在日志行层面，领域知识也有助于理解日志中提到的某些参数或变量所具有的现实世界语义或物理意义。尤其是在实际的无监督场景中，这些方面通常很难由机器学习系统自行判断。

面向异构日志的统一模型：大多数日志分析模型对日志预处理或分组的方式高度敏感，因此需要针对每种类型的应用程序日志定制预处理流程。这表明，我们需要具有更强泛化能力的预处理层的统一模型，以处理各种异构日志数据以及不同类型的日志分析任务。虽然 [21] 是最早探索这一方向的研究之一，但构建可实际应用的日志分析模型仍有很大的研究空间。

C. 追踪与多模态事件检测

问题定义

追踪数据是半结构化的事件日志，包含关于服务图拓扑结构的跨度信息。追踪异常检测依赖于在给定时刻的拓扑图上寻找异常路径，同时也依赖于直接从追踪事件日志文本中发现异常信息。处理追踪数据有多种方式。追踪数据通常带有时间戳和相关的序列信息，因此可以转换为时间序列数据。追踪数据也以追踪事件日志的形式存储，其中包含丰富的文本信息。此外，追踪数据还存储了拓扑信息，可用于重建表示关系的服务图

系统的组成部分之间。从数据角度来看，追踪数据可以轻松转换为多种数据模态。因此，本节将基于追踪的异常检测与多模态异常检测结合起来进行讨论。近年来，借助多模态深度学习技术，追踪异常检测可以结合追踪数据传递的不同层次信息，并学习更全面的异常检测模型 [119][120]。

实证方法

由于拓扑结构变得非常复杂且动态，追踪数据在微服务系统架构中受到了更多关注。追踪异常检测始于大规模系统调试的实际应用 [121]。经验性的追踪异常检测和根因分析始于构建追踪图，并识别所构建图中的异常结构。从追踪数据构建追踪图通常非常耗时，因此设计了一个离线组件来训练和构建此类追踪图。除此之外，为了适应在大规模系统中检测和定位问题的使用需求，追踪异常检测和根因分析算法通常还包含一个在线部分，以支持实时服务。例如，Cai 等发布了他们关于实时追踪级诊断系统的研究，该系统已被阿里巴巴数据中心采用。这是为数不多的处理真实大型分布式系统的研究之一 [122]。

大多数实证追踪异常检测研究采用离线与在线设计模式来构建图模型。在离线建模中，会使用无监督或半监督技术来构建追踪实体图，这与进程发现和挖掘领域中的技术非常相似。例如，在早期的一项网页图异常检测研究中，曾使用 PageRank 构建网页图 [123]。构建追踪实体图后，可以使用多种技术检测异常。一种常见方法是将当前图模式与正常图模式进行比较。如果当前图模式与正常模式显著偏离，则报告异常追踪数据。

另一种方法是使用数据挖掘和统计学习技术，在不构建离线追踪图的情况下进行动态分析。Chen 等人提出了 Pinpoint[124], ，这是一个根因分析框架，它利用数据挖掘技术，在真实客户端请求穿越系统时，实时对这些请求进行粗粒度标记。Pinpoint 会发现这些请求的成功／失败状态与故障组件之间的相关性。该方法会即时处理追踪数据，不依赖任何静态依赖图模型。

基于深度学习的方法

近年来，深度学习技术开始应用于追踪异常检测和根因分析。借助深度学习框架，将通用追踪图信息与每个追踪事件的详细信息相结合，以训练多模态学习模型，也成为可能。

长短期记忆网络(长短期记忆网络)网络 [125] 是早期追踪和多模态领域中非常流行的神经网络模型异常检测。长短期记忆网络是一种特殊类型的循环神经网络（RNN），并且已被证明在许多其他领域取得成功。在智能运维中，长短期记忆网络也常用于指标和日志异常检测应用程序。追踪数据与循环神经网络天然契合，主要体现在两个方面：1）追踪数据的拓扑顺序可以建模为事件序列。这些事件序列可以轻松转换为循环神经网络的模型输入。2）追踪事件通常包含能够传达丰富信息的文本数据。原始文本，包括结构化和非结构化部分，都可以通过标准的分词和嵌入技术转换为向量，并将其作为模型输入馈送给循环神经网络。这类深度学习模型架构可以扩展为支持多模态输入，例如将追踪事件向量与数值时间序列值结合起来 [119]。

为了更好地利用追踪数据的拓扑信息，图神经网络也被引入追踪异常检测。Zhang 等人开发了 DeepTraLog，这是一种采用门控图神经网络 [120]的追踪异常检测技术。DeepTraLog 旨在解决复杂微服务系统中的异常检测问题，因为在这类系统中，服务实体关系不易获取。此外，通过门控图神经网络训练构建的图也可用于定位问题，从而提供额外的根因分析能力。

局限性

随着越来越多的应用从单体架构转向微服务架构，追踪数据变得越来越有吸引力。基于机器学习的追踪异常检测面临着若干挑战。

数据质量。 据我们所知，目前有多个链路追踪数据采集平台，这些平台之间的追踪数据格式和质量并不一致，尤其是在生产环境中。为了使用这些追踪数据进行分析，研究人员和开发人员必须投入大量时间和精力来清理和重构数据，以便将其输入机器学习模型。

难以获取标签。生产数据的标签很难获取。对于给定的事件，为相应的追踪添加标签，需要确定事件发生的时间和地点，以及可能位于完全不同时间和地点的根本原因。要为数千个事件获取如此完整的标签极其困难。因此，现有的大多数追踪分析研究仍使用合成数据来评估模型性能。这让人更加怀疑所提出的解决方案能否解决真实生产环境中的问题。

多模态和图学习模型尚不完善。追踪数据十分复杂。当前的追踪分析会将追踪数据简化为事件序列或时间序列数值，即使在多模态场景下也是如此。然而，现有模型架构未能充分利用追踪数据中的所有信息。基于图的学习或许能提供一种解决方案，但目前对此主题的探讨仍十分有限。

离线模型训练。 现有研究中的深度学习模型依赖离线模型训练，部分原因是模型训练通常非常耗时，而且

与实时服务的目标相矛盾。然而，离线模型训练将静态依赖引入动态系统中。这些依赖可能导致额外的性能问题。

未来趋势

统一追踪数据 近期，OpenTelemetry 引领了统一不同平台上的可观测性遥测数据（包括指标、日志、追踪数据等）的工作。这项工作将为未来的追踪分析带来巨大益处。借助更加统一的数据模型，人工智能研究人员可以更轻松地获取训练更优模型所需的数据。其他各方也可以轻松即插即用地使用训练好的模型，从而进一步提升模型质量。

用于检测和根因分析的统一引擎追踪图包含特定时刻系统的丰富信息。借助追踪数据，可以一步完成事件检测和根因定位，而无需像当前这样分成连续的两个步骤。已有研究表明，只需检查构建好的图，检测模型就能揭示足够的信息来定位根因[120]。

多模态遥测数据的统一模型追踪数据分析为研究人员创建多种遥测数据模态的整体视图带来了机遇，因为追踪数据可以转换为文本序列数据和时间序列数据。这些成果可以扩展到包含来自不同来源的日志或指标。最终，我们可以期待能够使用多模态遥测数据进行事件检测和根因分析的统一学习模型。

在线学习 现代系统是动态且不断变化的。当前的两步式解决方案依赖离线模型训练以及在线服务或推理。两次离线训练周期之间的任何系统演变都可能引发潜在问题，并损害模型性能。因此，支持在线学习对于确保模型在真实生产环境中保持高性能至关重要。

V. 故障预测

事件检测和事件根因分析是在事件已经发生后，用于减轻事件影响并提高服务可用性的更具被动性的措施。另一方面，还可以采取其他主动式措施，预测近期是否可能发生潜在事件，并防止其发生。软件系统中的故障就是这类极具破坏性的事件，通常会先表现出所需系统功能偏离正常运行行为的症状，并最终导致无法满足服务级别协议。故障预测是事件管理中的一项主动式任务，其目标是通过分析不同类型的系统数据（关键绩效指标、日志记录和追踪数据）持续监控系统健康状况，并生成早期预警，以防止故障发生。因此，为了处理不同类型的遥测数据源，可以将故障预测任务定制为基于指标的故障预测和基于日志的故障预测。本节将详细介绍这两种方法。

A. 基于指标的故障预测

指标数据通常对监控系统十分有用。直接利用这些数据提前预测事件发生非常简单。这样一来，便可以采取一些主动式措施来防止事件发生，而不是缩短检测时间。通常，在有故障标签可用时，该问题可以表述为不平衡二元分类问题；如果预先定义了监控指标的正常范围，则可以表述为时间序列预测问题。一般而言，故障预测 [126]通常采用机器学习算法来学习历史故障数据的特征，构建故障预测模型，然后部署该模型来预测未来发生故障的可能性。

方法

通用故障预测：近年来，越来越多的工作开始利用整个监控系统中的故障信号来进行通用故障事件预测。 [127]收集了整个系统中的告警信号，并发现了告警信号之间的依赖关系，随后采用基于梯度提升树的模型来学习故障模式。 [128] 提出了一种有效的特征工程流程，用于处理复杂的告警数据。该方法采用多示例学习来处理噪声告警，并通过可解释性分析生成可解释的预测结果，以便于理解和处理事件。

特定类型故障预测： 相比之下，一些研究致力于主动预测各种特定类型的故障。 [129]从历史交换机日志中提取统计特征和文本特征，并采用随机森林预测数据中心网络中的交换机故障。 [130]收集了 SMART [131] 和系统级信号的数据，并提出了一种结合长短期记忆网络和随机森林的混合模型，用于云服务系统中的节点故障预测。 [132]通过代价敏感排序模型开发了一种磁盘错误预测方法。这些方法针对特定类型的故障预测，因此在实践中的应用受到限制。

挑战与未来趋势

尽管传统的有监督学习可用于处理分类或回归问题中的故障预测，但仍需克服以下主要挑战。首先，由于故障案例数量有限，数据集通常极不平衡。这给预测模型带来了重大挑战，使其难以同时达到较高的精确率和召回率。其次，原始信号通常包含噪声，并非事件发生前的所有信息都有帮助。如何提取预兆特征/模式并滤除噪声，对预测性能至关重要。第三，典型系统通常每分钟会生成大量信号，因此需要以流式方式更新预测模型，并在计算资源有限的情况下处理大规模数据。第四，故障预测的后处理对于故障管理系统提升可用性非常重要。例如，提供可解释的故障预测，有助于工程师采取适当的措施。

B. 基于日志的事件检测

与事件检测和根因分析一样，故障预测也是一项极其复杂的任务，尤其是在企业级系统中：这类系统包含许多彼此分布、相互连接的组件、服务和微服务，它们以异步方式交互。该任务的主要复杂性之一，在于能否及早检测出预示重大中断的信号，即使系统此时可能仅表现出轻微或可控的偏离正常行为。由于问题本身的这一特性，仅监控关键绩效指标可能不足以实现早期检测，因为许多此类指标对逐渐显现的问题反应滞后，或粒度不够细，无法捕捉事件的早期信号。另一方面，系统日志和软件日志广泛存在于系统数据中，能够持续捕获丰富且非常详细的运行时信息，而这些信息往往与检测潜在的未来故障息息相关。

因此，各类主动式日志分析已作为持续监控任务应用于不同的工业应用中，并被证明在更精细地预测故障和定位潜在故障源方面颇为有效。这项工作需要分析日志数据中的事件序列，甚至可能将其与指标等其他数据源关联起来，以检测表明事件正在发展的异常事件模式。文献中通常采用有监督或半监督机器学习模型，通过学习和建模历史故障数据的特征，预测未来发生故障的可能性。在某些情况下，这些模型还可借助有关系统间复杂关系的领域知识进一步增强。尽管这项任务不像日志异常检测和根因分析那样受到广泛研究，公开数据集和基准数据也较少，但软件和系统维护中的日志数据仍在预测未来潜在故障方面发挥着十分重要的作用。文献中，基于日志数据的故障预测任务通常应用于两大类系统：同构系统和异构系统。

同构系统中的故障预测

在同质系统中，例如高性能计算系统或大规模超级计算机，这意味着预测独立故障，其中大多数系统利用序列信息来预测单个组件的故障。

时间序列建模：在同质系统中，[133]，[134]从结构化日志中提取表示系统健康状况的特征，并将其建模为基于时间序列的异常预测问题。同样， [135]通过特征工程提取关键事件期间的特定模式，并构建有监督二元分类器来预测故障。 [136]通过解析将非结构化日志转换为模板，并应用特征提取和时间序列建模来预测异常的激增、频率和季节性模式。

有监督分类器一些较早的研究在有监督分类环境中，使用支持向量机、近邻分类器或基于规则的分类器等传统机器学习模型来预测故障 [137]，[93]，[138]，或分类器集成 [93] 或基于隐马尔可夫模型的分类器 [139] ，基于从日志事件序列中手工构造的特征，或基于随机索引的日志编码，而[140], [141] 使用类似长短期记忆网络的深度循环神经模型处理日志的语义表示。 [142] 通过首次故障识别和基于因果关系的过滤来预测和诊断故障，并通过关联规则挖掘方法合并相关事件以进行过滤。

异构系统中的故障预测

在异构系统中，例如大规模云服务，尤其是在分布式微服务环境中，服务中断可能由异构组件引起。大多数流行方法利用有关系统组件之间关系和依赖关系的知识来预测故障。在这类系统中， [143] 构建了一个贝叶斯网络，用于识别离线环境下从系统日志中提取的告警信号与过去服务中断之间的条件依赖关系，并使用梯度提升树在在线环境下预测未来的服务中断。 [144] 使用一种排序模型，结合长短期记忆网络隐藏状态中的时间特征和随机森林中的空间特征，对故障指示告警与服务中断之间的关系进行排序。 [145] 基于从追踪日志中提取的手工设计特征，训练追踪级和微服务级预测模型，以检测三种常见的微服务故障。

VI. 根因分析

根因分析 (RCA) 是执行一系列行动以发现事件根因的进程。DevOps 中的根因分析侧重于构建标准进程工作流，以更系统地处理事件。没有人工智能时，根因分析更多是创建任何 DevOps成员都可以遵循的规则，以解决重复发生的事件。然而，当系统规模庞大且复杂时，为每种类型的重复事件分别创建规则和进程工作流是不可扩展的。人工智能模型能够处理大量输入数据，并从现有事件及其处理方式中学习表示，而无需人类定义工作流的每一个细节。因此，基于人工智能的根因分析具有重塑根因发现方式的巨大潜力。

在本节中，我们将讨论一系列基于人工智能的根因分析主题，并按输入数据模态进行分类：基于指标、基于日志、基于追踪以及基于追踪和多模态的根因分析。

A. 基于指标的根因分析

问题定义

随着微服务架构的迅速普及，多服务应用已成为现实世界信息技术应用程序的标准范式。多服务应用通常包含数百个相互交互的服务，因此更难以检测服务故障并确定根因。根因分析（RCA）方法利用这些服务上监控的关键绩效指标，在检测到系统故障时确定根因，从而帮助工程师和

站点可靠性工程师在故障排查过程中提供帮助*。基于关键绩效指标进行根因分析的核心思路是分析这些指标之间的关系或依赖性，并在出现异常时利用这些关系确定根因。通常有两类方法：1）通过指标数据分析，在观测到异常的同时识别异常指标；2）发现能够表示服务之间因果关系的拓扑/因果图，并据此识别根因。

指标数据分析

在多服务应用程序中检测到异常时，关键绩效指标出现异常的服务可能就是根因。第一种方法直接分析这些关键绩效指标，并基于以下假设确定根因：发生异常时，一个或多个关键绩效指标会出现显著变化。因此，关键在于判断在异常时间戳处，某个关键绩效指标在给定长度的回溯窗口或快照中是否出现模式或幅度变化。

Nguyen 等人 [146], [147] 提出了两种类似的根因分析方法，通过分析底层系统指标（例如 CPU、内存和网络统计数据）来进行分析。这两种方法都会在检测到性能异常时，首先通过变点检测算法检测每个组件的异常行为，然后根据按时间顺序排列所有关键变点后得到的传播模式确定根因。由于现实中的多服务应用程序通常有数百个关键绩效指标，因此变点检测算法必须高效且稳健。 [146] 提出了一种结合累积和图与自助法的算法，用于检测变点。为了从该算法发现的变点中识别关键变点，他们使用分离度指标衡量每个变点的变化幅度，并提取分离度值为离群值的关键变点。由于最早出现的异常可能从对应的服务传播到其他服务，因此他们通过按时间顺序排列关键变点来确定根因。为了进一步提高根因定位的准确性， [147] 提出了一种新的故障定位方法，同时考虑传播模式和服务组件依赖关系。

不同于变点检测，Shan 等人 [148] 开发了一种低成本的根因分析方法，称为 ϵ-Diagnosis，用于检测 Web 服务的小窗口长尾延迟的根因。 ϵ-Diagnosis 假设异常服务的根因指标在异常时段和正常时段之间存在显著变化。该方法采用双样本检验算法和 ϵ-统计量来衡量时间序列的相似性，从而识别根因。在双样本检验中，一个样本（正常样本）取自正常时段的快照，另一个样本（异常样本）取自异常时段。如果异常样本与正常样本之间的差异具有统计显著性，那么这些样本对应的指标就可能是根因。

基于拓扑或因果图的分析

度量数据分析方法的优势在于能够处理数百万个指标。但其中大多数方法并未考虑应用程序中各服务之间的依赖关系。第二类根因分析方法利用这类依赖关系，通常包含两个步骤：根据关键绩效指标和领域知识构建拓扑/因果图，以及根据观测到的异常提取异常子图或路径。这类图既可以根据特定应用程序的拓扑结构（领域知识）重建（[149], [150], [151], [152]），也可以通过因果发现技术根据指标自动估计（[153], [154], [155], [156], [157], [158], [159]）。为识别观测到的异常的根因，可以在发现的拓扑/因果图上应用随机游走（例如，[160], [156], [153]）、PageRank（例如，[150]）或其他技术。

当服务图（服务之间的关系）或调用图（服务之间的通信）可用时，多服务应用的拓扑图可以自动重建，例如，[149],[150]。但是，这类领域知识通常不可用或仅部分可用，尤其是在研究关键绩效指标之间的关系而非API调用时。因此，给定观测到的指标，因果发现技术，例如，[161], [162], [163] 在构建描述这些指标之间因果关系的因果图方面发挥着重要作用。应用于根因分析的最流行因果发现算法是众所周知的PC算法 [161] ，这得益于其简单性和可解释性。它从一个完全无向图开始，并通过条件独立性检验消除指标之间的边。随后，通过寻找 V结构并进行方向传播来确定这些边的方向。基于不同的数据属性，还可以应用 PC算法 的某些变体[164], [165], [166] 。

给定所发现的因果图，可以通过随机游走确定观测到的异常的可能根因。图上的随机游走是一种随机过程，从某个节点开始，并在每个时间步随机移动到另一个节点。从一个节点移动到另一个节点的概率定义在转移概率矩阵中。用于 RCA 的随机游走基于以下假设：与异常关键绩效指标相关性更高的指标更有可能是根因。每次随机游走都从一个对应于异常指标的异常节点开始，因此访问频率最高的节点最有可能是根因。随机游走方法的关键在于确定转移概率矩阵。通常，计算转移概率矩阵包括三个步骤，即前向步骤（从一个节点游走到其某个父节点的概率）、后向步骤（从一个节点游走到其某个子节点的概率）和自环步骤（停留在当前节点的概率）。例如，[153], [158], [159], [150]根据每个指标在异常时段内与检测到的异常指标之间的相关性计算这些概率。但是，基于相关性的随机游走可能无法准确定位根因 [156]。因此，[156] 提出使用偏相关而不是相关性来计算转移

*一篇关于云应用异常检测和根因分析的优秀综述[22]

概率，从而可以消除两个指标中混杂因素的影响。

除了随机游走之外，还可以应用其他因果图分析技术。例如， [157], [155] 通过递归访问受异常影响的所有指标来查找观测到的异常的根因，例如，如果某个受影响指标的父节点未受异常影响，则该指标被视为可能的根因。[167] 采用基于广度优先搜索（BFS）算法的搜索算法来查找根因。搜索从一个异常的关键绩效指标开始，并提取因果图中从该指标出发的所有可能路径。随后，根据路径长度以及路径中边的权重总和对这些路径进行排序。排名靠前的路径末端节点被视为根因。 [168] 基于因果图，考虑将反事实用于根因分析；也就是说，给定一个函数因果模型，它通过计算每个噪声项对异常分数的贡献来查找检测到的异常的根因，并使用沙普利值的概念对这些贡献进行对称化。

局限性

数据问题 对于每个服务都监控数百个关键绩效指标的多服务应用程序而言，确定哪些指标对于识别根因至关重要，是一项极具挑战性的任务。收集到的数据通常无法完整呈现系统架构，例如可能缺少一些重要指标。这些缺失指标可能是其他指标的因果父节点，这违反了 PC 算法关于不存在潜在混杂因素的假设。此外，由于真实世界的关键绩效指标存在噪声、非平稳性和非线性关系，恢复准确的因果图变得更加困难。

缺乏领域知识 关于受监控应用程序的领域知识（例如服务图和调用图）有助于提升根因分析性能。但对于复杂的多服务应用程序，即使是开发人员也可能无法完全理解所有受监控指标的含义及其相互关系。因此，专家提供的领域知识通常并不完整，有时还会与从观测数据中发现的知识相冲突。

因果发现问题 基于因果图分析的根因分析方法利用因果发现技术来恢复关键绩效指标之间的因果关系。所有这些技术都对数据属性有一定假设，而这些假设可能无法通过真实世界的数据满足，因此发现的因果图总是包含错误，例如连接或方向不正确。近年来，许多因果发现方法在不同假设和特性下被提出，因此很难根据观测数据选择最合适的方法。

人在环路中 DevOps 或 SRE团队收到某种根因分析方法识别出的根因后，会进一步分析，并反馈这些根因是否合理。大多数根因分析方法无法利用此类反馈来提升根因分析性能，也无法解释识别出的根因为何不正确。

缺乏基准不同于事件检测问题，我们缺乏用于评估根因分析性能的基准，例如，只有少量带有真实根因标注的公开数据集可用，而且以往大多数研究都使用内部私有数据集进行评估。尽管可以利用一些多服务应用演示程序/模拟器生成用于根因分析评估的合成数据集，但这些演示应用的复杂度远低于真实应用，因此这类评估可能无法反映实际性能。缺乏公开的真实世界基准，阻碍了新根因分析方法的开发。

未来趋势

根因分析基准用于评估根因分析方法性能的基准，对于实际应用和学术研究都至关重要。这些基准既可以是一组带有真实根因标注的真实数据集，也可以是架构接近实际应用的模拟器。构建此类大规模真实基准，对于推动根因分析中的新颖想法或方法至关重要。

结合因果发现与领域知识专家提供的领域知识对于提高因果发现的准确性十分宝贵，例如，提供指标之间必需或禁止的因果连接。但有时，这类领域知识会在恢复因果图时引发更多问题，例如与数据属性或条件独立性检验相冲突，或在图中引入环。如何以有原则的方式结合因果发现与专家领域知识，是一个有趣的研究课题。

让人在环路中将人的交互融入根因分析方法，对于实际应用程序十分重要。例如，可以通过迭代方式构建因果图：先使用某种因果发现算法重建初始因果图，然后由用户检查该图，并提供领域知识约束（例如，指出哪些关系有误或缺失），供算法据此修订图。DevOps 或 SRE团队编写的、包含事件详细分析的根因分析报告，有助于提升根因分析性能。如何利用这些报告提升根因分析性能，是另一个重要的研究课题。

B. 基于日志的根因分析

问题定义

分诊和根因分析是事件管理生命周期中最复杂且最关键的阶段之一。鉴于该问题的本质是调查事件的起源或根本原因，仅分析最终的关键绩效指标通常是不够的。尤其是在微-服务应用程序设置或数百个服务相互交互的分布式云环境中，根因分析和故障诊断尤其具有挑战性。为了在此类复杂环境中定位根本原因，工- 程师、站点可靠性工程师和服务所有者通常需要调查核心系统数据。日志是此类普遍存在的系统数据之一，其中包含丰富的运行时信息。因此，日志分析任务的最终目标之一是支持事件分诊并定位根本原因，从而诊断故障和故障。

从系统中不同来源和微服务的异构日志数据入手，典型的基于日志的智能运维工作流首先包含一个日志处理与分析层，涉及日志解析、聚类、摘要和异常检测。随后，日志分析和异常检测可支持因果推断层，用于分析日志事件之间的关系与依赖，以及可能检测到的异常事件。可进一步将从单个服务内部或不同服务之间的日志中提取的这些信号，与指标、追踪数据等其他可观测性数据关联起来，以检测事件的根因。通常，这需要构建因果图或挖掘日志事件的知识图谱，并将其与关键绩效指标或追踪数据、服务调用图等其他形式的系统数据关联起来。通过这些方式，目标是分析它们之间的关系与依赖，最终识别异常的可能根因。与日志异常检测这类更具体的问题不同，基于日志的根因分析是一项更加开放式的任务。因此，大多数关于基于日志的根因分析的文献都聚焦于部署在真实环境中的工业应用，并使用由内部领域专家收集的内部基准数据进行评估。

日志根因分析方法的典型类型

在文献中，基于日志的根因分析任务已经通过各种方法进行了探索。一些研究构建知识图谱和知识，并利用基于数据挖掘的解决方案；另一些研究则遵循因果机器学习或因果知识挖掘的基本原则。除此之外，还有一些基于日志的根因分析系统使用传统机器学习模型，通过特征工程、相关性分析或有监督分类器来检测根因。

基于手工设计特征的方法： [169] 针对Spark 日志，采用手工设计特征工程和特定类型根因的概率估计。 [170] 针对结构化日志，在特征组上进行频繁项集挖掘和关联规则挖掘。

基于相关性的方法： [171], [172]基于通过日志和监控指标获得的异常分数之间的互信息进行相关性分析，从而定位根因。同样， [173]使用基于PCA、ICA 的相关性分析来捕捉日志与后续故障之间的关系。 [84], [174] 使用 PCA 检测异常系统调用序列，并通过频繁模式挖掘将其映射到应用程序函数。[175] 使用基于 LSTM的日志模板序列建模，这些日志模板通过对相似日志聚类进行模式匹配来识别，以预测故障。

基于有监督分类器的方法： [176] 通过学习历史故障日志的统计和神经表示的有监督分类器，自动进行异常日志的检-测，并将新的错误模式与 OpenStack上的正常云行为进行比较。 [177]采用统计技术分析数据分布，以识别性能问题的细粒度类别，并使用快速矩阵恢复RPCA确定根本原因。 [178], [179] 使用 KNN 或其有监督版本来识别导致故障的日志行。

基于知识挖掘的方法： [180], [181] 通过从日志行中提取自定义实体及其关系，并从整体日志转储中挖掘它们之间的时序依赖和流程依赖，将日志事件汇总为实体关系知识图谱，从而采取了不同的方法。虽然这为日志摘要提供了结构化程度更高的表示，但这也是一种直观的日志知识聚合方式，还能弥合创建日志数据的开发者社区与通常在调查事件时使用日志数据的站点可靠性工程师之间的知识鸿沟。然而，构建这种日志知识图谱表示的最终目标是促进根因分析。尽管这些工作针对这一愿景提供了根因分析案例研究等用例，但在如何将这类知识挖掘更具体地应用于根因分析方面，仍有很大的研究空间。

基于知识图谱的方法：在基于知识图谱的方法中， [182] 通过利用启发式方法和领域知识，从基于日志数据构建的随机森林中提取规则，并持续构建知识库，以在线方式诊断和分流性能故障问题。 [151]结合关键绩效指标和日志数据构建系统图。基于从这些数据源中检测到的异常，它从中提取异常子图，并将其与正常系统图进行比较，以检测根因。其他研究从正常执行中挖掘正常日志模式 [183]或时间加权控制流图 [99]，并估计持续故障期间的执行与这些模式之间的偏差，从而提出根因。 [184], [185], [186] 从正常执行和手动注入的故障，或从性能良好或糟糕的系统中挖掘执行序列或用户操作 [187] ，将其纳入知识库，并利用相似故障会产生相似故障现象这一假设，对故障类型进行匹配和诊断。随着时间推移，大多数这类基于知识的方法都会逐步扩展其知识或规则，以应对更新的事件类型。

基于因果图的方法： [188]将日志表示为错误事件计数，并对其进行多变量时间序列建模。随后，该研究采用类似 PageRank 的中心性检测方法，推断其与 KPI 错误率之间的因果关系，从而识别主要根因。 [167] 基于从日志、指标、追踪数据和系统依赖图中提取的运维实体构建知识图谱，并使用 PC 算法挖掘因果关系，以检测事件的根因。 [189] 基于从指标和基于日志的异常检测中提取的特征，构建知识增强的分层贝叶斯网络，以推断根因。 [190]基于从日志、指标和服务依赖图中提取的事件构建动态因果图。[191]同样通过聚类和挖掘相似事件，构建日志事件的因果依赖图，并利用该图推断故障发生的进程。

此外，在相关的网络分析领域中， [192],[193], [194] 通过将解析后的日志模板计数建模为多变量时间序列，借助因果分析挖掘网络事件的原因。 [195], [156] 利用关键绩效指标和服务调用图进行因果推理，以定位

微服务系统中的根因，而未来的研究方向之一是将非结构化日志也纳入此类因果分析。

挑战与未来趋势 收集监督 la-

标签 由于这是一项复杂且开放式的任务，收集用于根因分析的监督标签颇具挑战，需要大量领域专业知识和人工投入。虽然仍可获取少量监督数据用于评估，但要达到训练这些模型所需的规模并不现实。同时，由于问题本身的复杂性，完全无监督的模型往往表现相当差。**数据质量：** 异构数据上的根因分析工作流程 -

异构非结构化日志数据通常涉及各种不同的分析层、预处理、解析、分区和异常检测。这会导致来自这些组件的错误（包括标注错误和模型预测错误）不断累积并级联，从而需要在根因分析任务中处理噪声数据。此外，根因分析标注任务极具挑战性，进一步增加了产生噪声数据的可能性。**类别不平衡问题：** 根因分析中的

海量日志带来了极端类别不平衡这一额外问题——在数百万条日志行或日志模板中，只有极少数实例可能与真正的根本原因有关。**模型的通用性：** 大多数现有-

根因分析领域的文献往往针对各自的应用程序定制方法，即使是其他类似系统也难以直接采用。这表明，我们需要更具普适性的架构来建模根因分析任务，而这又需要更稳健、通用的日志分析模型，以处理来自不同系统的异构日志数据。**持续学习框架：** 一种

在分布式云环境中，根因分析面临的一项挑战是环境变化迅速，导致出现新的事件类型和不断演变的致因因素。这种非平稳学习设置给根因分析带来了不小的挑战，但它确实是所有实际工业应用中的关键方面。**人在回路框架：** 虽然既非

完全有监督，也非无监督的场景适用于这项任务，但我们需要支持人在回路的框架，以便纳入领域专家的反馈来改进系统，尤其是在致因因素可能随时间演变的敏捷环境中。现实世界的公开

基准测试： 该领域的大多数文献都聚焦于工业应用，并采用内部评估设置。在某些情况下，研究人员会通过在内部仿真环境中注入故障、错误或异常来构建内部测试平台（例如，在 Spark 平台中注入 CPU、内存、网络和磁盘异常），或使用常见的测试环境（如 Grid5000 测试平台、基于在线购物或火车票预订平台构建的开源微服务应用，或开源云操作系统 OpenStack）。其他研究则通过在内部云原生应用的真实环境中部署解决方案进行评估，例如在 IBM Bluemix 平台上，或用于 Facebook 应用程序。也有研究在阿里巴巴等大数据云计算平台上的数百个真实生产服务中，或在 eBay 等电子商务企业的数千个服务中进行评估。关于这一点，一个显著的局限性是，缺乏可复现的开源公共基准，用于在实际工业环境中评估基于日志的根因分析。这会阻碍更开放的研究，也不利于公平评估用于应对这一挑战性任务的新模型。

C. 基于追踪和多模态的根因分析

问题定义。 理想情况下，复杂系统的根因分析需要利用所有可用数据，包括机器生成的遥测数据和人类活动记录，以找出问题的潜在根因。本节将讨论基于追踪的根因分析以及多模态根因分析。我们还将介绍基于人类记录（如事件报告）的根因分析研究。归根结底，根因分析引擎应能够处理任意数据类型并发现正确的根因。

追踪数据的根因分析

在上一节（第四节-C）中，我们讨论了可将追踪视为用于异常检测的多模态数据。与追踪异常检测类似，追踪根因分析也会利用服务映射的拓扑结构。追踪根因分析通常在问题被检测到之后启动，而不是检测异常追踪或路径。追踪根因分析技术有助于简化工程师和站点可靠性工程师的故障排查流程。此外，追踪根因分析可以按需触发，而非持续运行。这使得它可能采用的技术有别于追踪异常检测。

追踪实体图。 从技术角度来看，追踪根因分析和追踪异常检测有着相似的视角。据我们所知，单独讨论追踪根因分析的现有研究并不多。相反，无论是经验方法 [121] [196] 还是深度学习方法 [120] [197]，追踪根因分析都是追踪异常检测的一项附加功能或额外优势。在追踪异常检测中，离线训练后构建的追踪实体图（TEG）能够清晰地呈现应用系统中各组件之间的关系。因此，除了异常检测之外， [122]还实现了一种实时根因分析算法，通过相对重要性分析，将当前异常追踪模式与正常追踪模式进行比较，从而发现问题的最深层根源。其在生产环境中的实验表明，与朴素的固定阈值方法相比，该根因分析算法能够取得更高的精确率和召回率。基于追踪实体图进行根因分析的有效性，也已在基于深度学习的追踪异常检测方法中得到验证。Liu等 [198] 提出了一种用于追踪异常检测的多模态长短期记忆模型。随后，根因分析算法可以将每条异常追踪数据与模型训练追踪数据进行比对，并通过定位不在正常调用路径中的下一个被调用微服务，找出根因。根据这项工作的评估结果，该算法在合成数据集以及四个大型生产服务的生产数据集上都表现良好。

在线学习。另一种方法是使用数据挖掘和统计学习技术，在不构建离线追踪图的情况下执行动态分析。传统的追踪管理系统通常提供基本的分析能力，用于诊断问题并发现根因 [199]。此类分析可以在线执行，无需耗费成本进行模型训练。Chen 等人提出了 Pinpoint[124], ，这是一种用于根因分析的框架，可在真实客户端请求实时穿越系统时，使用这些请求的粗粒度标记数据，并结合数据挖掘技术。Pinpoint发现这些请求的成功/失败状态与故障组件之间的相关性。整个方法会即时处理追踪数据，不依赖任何静态依赖图模型。另一个相关领域是使用故障排除指南数据，其中 [200] 会根据与事件描述的语义相似度推荐故障排除指南，而 [201] 则侧重于将故障排除指南自动化为执行工作流，以此作为修复事件的一种方式。

事件报告根因分析

智能运维文献中的另一个重要方向，是从领域专家整理的数据（事件报告、事件调查数据、错误报告等）中挖掘有用知识，以实现事件根因分析和自动化修复的最终目标。这是一项开放式任务，可用于多种目的——对非结构化或半结构化数据进行结构化和解析，从中提取目标信息或主题（使用主题建模或信息提取），以及将知识挖掘并汇总为结构化形式。

这些任务的最终目标主要是根因分析，其中一些任务也侧重于推荐补救措施以缓解事件。尤其是在大多数基于云的环境中，随着时间推移，反复发生的事件越来越多，它们表现出相似的症状，且具有相似的根因。因此，从各种数据源中挖掘和整理知识至关重要，以便数据驱动的人工智能模型或领域专家能够利用这些知识，更好地实现知识复用。

因果关系图。 [202] 从历史事件数据中提取并挖掘因果关系图，并使用人在回路的监督和反馈进一步完善因果关系图。[203] 使用分布式频繁模式挖掘算法构建异常相关性图FacGraph。 [204] 通过调整从类似历史事件中检索到的修复方案，推荐适当的修复操作。尽管最终任务涉及修复建议，但该系统仍需要理解事件的性质和根本原因，以便检索出有意义的过往事件。

知识挖掘。 [205], [206]使用基于长短期记忆网络的 CRF 模型，从事件报告中提取命名实体和关系，并据此挖掘知识图谱。 [207] 从以往的事件调查中提取症状、根因和补救措施，并构建神经搜索系统和知识图谱，以便为重复发生的事件提供基于检索的根因分析和补救建议。

未来趋势

更高效的追踪平台。 目前，关于追踪相关主题的研究非常有限。一项根本性挑战在于追踪平台。追踪数据的采集、存储、查询和管理都存在瓶颈。追踪数据的规模通常远大于日志和指标。如何更高效地采集、存储和检索追踪数据，对于追踪根因分析的成功至关重要。

在线学习。 与追踪异常检测相比，在线学习在跟踪根因分析中发挥着更重要的作用，尤其是在大型云系统中。根因分析工具通常需要即时分析证据，并将最可疑的证据与正在发生的事件关联起来，因此对时效性要求很高。例如，我们知道追踪实体图（TEG）能够实现准确的跟踪根因分析，但前提是 TEG 能反映系统的当前状态。如果只能通过离线训练来生成 TEG，那么这类方法在现实生产环境中的表现始终令人存疑。因此，在这种情况下，使用在线学习来生成 TEG，是确保高性能的更好方式。

多模态遥测数据上的因果图。 追踪数据所承载的最宝贵信息，是大型系统复杂的拓扑顺序。没有追踪数据时，系统运维的因果分析依赖时间和几何相关性来推断因果关系，而现有的因果推断方法实际上很少能应用于现实系统。然而，有了追踪数据，就可以非常方便地获得请求流经整个系统的真实情况。因此，我们认为，如果能通过多模态遥测数据进行学习，就更容易获得质量更高的因果图。

系统的完整知识图谱。目前，知识挖掘已针对单一数据类型进行了尝试。然而，为了反映复杂系统的全貌，人工智能模型需要从任何类型的数据中挖掘知识，包括指标、日志、追踪数据、事件报告和其他系统活动记录，然后构建包含完整系统信息的知识图谱。

VII. 自动化操作

虽然智能运维的事件检测和根因分析能力都有助于提供有关当前问题的信息，但采取正确的行动才是解决问题的关键步骤。如果没有自动化操作来采取行动，每项运维任务仍都需要人工操作人员参与。因此，自动化操作对于构建端到端全自动化智能运维系统至关重要。自动化操作既包括短期操作，也包括长期操作：1) 短期修复：快速修复问题的即时操作，包括服务器重启、在线迁移、自动扩缩容等；以及 2) 长期解决方案：针对代码错误修复、软件更新、硬件扩建和资源分配优化等任务采取的操作或提供的指导。本节将讨论三种常见的自动化操作类型：自动化修复、自动扩缩容和资源管理。

A. 自动化修复

问题定义

除了持续监控IT基础设施、检测问题并发现根因之外，以最少甚至无需人工干预的方式修复问题，是迈向下一代全自动智能运维的路径。自动化的问题修复（自动修复）通过利用已知信息、现有工作流和领域知识，采取一系列行动来解决问题。自动修复已经是许多信息技术运维场景中采用的概念，包括云计算、边缘计算、软件即服务等。

传统的自动修复进程基于各种定义明确的策略和规则，以确定针对给定问题应使用哪些工作流。而机器学习驱动的自动修复则利用机器学习模型来决定最适合缓解或解决问题的操作工作流。基于机器学习的自动修复在大型云系统或边缘计算系统中特别有用，因为在这些环境中，不可能为所有问题类别手动创建工作流。

现有工作

端到端自动修复解决方案通常包含三个主要组件：异常或问题检测、根因分析和修复引擎 [208]。这意味着，自动修复解决方案的成功高度依赖于异常检测和根因分析的质量，正如我们已在上述章节中讨论的那样。此外，修复引擎应能够从分析结果中学习、做出决策并执行操作。

知识学习。 这里的知识指的是多种类别。针对这一特定问题的异常检测和根因分析构成了大部分可学习的知识 [208]。修复引擎利用这些信息来定位和分类问题。此外，过去问题的人类活动记录（例如工单、修复漏洞日志）对于修复引擎学习问题在历史上是如何处理的完整情况也具有重要意义。在第 VI-A、VI-B 和VI-C节中，我们讨论了如何从系统指标、日志和人在回路记录中挖掘知识图谱。高质量的知识图谱能够清晰地描述系统组件之间的关系。

决策与执行。 Levy等。 [209]提出了 Narya，这是一个用于处理云系统中正在运行的虚拟机故障修复的系统。对于主机预计将发生故障的给定问题，修复引擎需要从实时迁移、软重启、服务修复等几个选项中决定采取的最佳措施。通过 A/B 测试和强化学习来决定采取哪项措施。通过在其修复引擎中采用机器学习，他们发现，与之前的静态策略相比，虚拟机中断情况显著减少。

未来趋势

自动修复研究与开发仍处于非常早期的阶段。现有研究主要聚焦于一个中间步骤，例如为给定的构建因果图。场景，或针对非常特定用例（例如线上机器中断）的端到端自动修复解决方案。以下是一些可以显著提升自动修复系统质量的主题。

系统集成目前仍没有一个统一的平台能够完成所有问题分析、学习上下文知识、制定决策并执行操作。

学习生成和更新知识图谱自动修复决策的质量在很大程度上取决于领域知识。目前，大部分领域知识由人类收集。未来，探索以更可靠的方式学习和维护系统知识图谱的方法具有重要价值。

人工智能驱动的决策制定与执行目前，大多数决策制定和操作执行都基于规则或统计学习。随着更强大的人工智能技术的出现，修复引擎便能利用丰富的信息，做出更复杂的决策。

B. 自动扩缩容

问题定义

云原生技术正成为在公有云或私有云上构建可扩展应用程序的事实标准，可实现松耦合、弹性、易于管理且可观测的系统[†]。GCP 和 AWS 等云系统按需向用户提供 CPU、存储、内存和数据库等资源。用户需要为其应用程序的工作负载指定这些资源的配置上限。如果应用程序中的某项服务超出特定资源的上限，终端用户就会遇到请求延迟或超时，因此系统运维人员会申请提高该资源的上限，以避免性能下降。但如果有数百项服务在运行，如此高的上限会导致大量资源浪费。自动扩缩容旨在无需人工干预即可解决这一问题，使应用程序能够根据工作负载行为模式动态配置资源，在不降低终端用户服务质量 (QoS) 的情况下尽可能减少资源浪费。

自动扩缩容方法可分为两类：反应式自动扩缩容和主动式（或预测式）自动扩缩容。反应式自动扩缩容会监控应用程序中的服务，并根据工作负载的变化增加或减少服务实例。

反应式自动扩缩容。反应式自动扩缩容非常有效，并且大多数云平台都支持。但它有一个潜在缺点，即只有在工作负载增加后才会扩展资源，因此在短时间内，工作负载已经升高，但额外容量尚未就绪。因而，终端用户在这段时间内可能会遇到响应延迟。主动式自动扩缩容旨在根据历史数据预测未来的工作负载，从而解决这一问题。本文主要讨论基于机器学习的主动式自动扩缩容算法。

主动式自动扩缩容。通常，主动式自动扩缩容包括三个步骤，即预测工作负载、估算

†https://github.com/cncf/foundation/blob/main/charter.md

容量和横向扩展。机器学习技术通常用于预测未来的工作负载，并估算受监控服务的合适容量，然后可以据此进行调整，以避免性能下降。

一种主动式自动扩缩容方法采用回归模型（例如，ARIMA [210], SARIMA [211],MLP、长短期记忆网络 [212]）给定受监控服务的历史指标，这类方法会训练特定的回归模型，以学习工作负载行为模式。例如，[213] 研究了用于工作负载预测的 ARIMA 模型，并表明该模型能够在对服务质量影响最小的情况下，提高资源利用率。 [214] 采用时间窗口多层感知机来预测具有不同类型工作负载的容器所处的阶段，并提出了一种预测式垂直自动扩缩容策略来调整容器大小。 [215] 也利用神经网络（尤其是多层感知机）进行工作负载预测，并将该方法与传统机器学习模型（例如线性回归和 K 近邻）进行了比较。 [216] 采用双向长短期记忆网络来预测 HTTP工作负载的数量，并表明在测试用例中，双向长短期记忆网络的表现优于长短期记忆网络和 ARIMA。这些方法需要准确的预测结果，以避免资源分配过多或不足；然而，由于用户请求中存在噪声和突发峰值，开发一种稳健的、基于预测的方法十分困难。

另一类方法基于强化学习（RL），将自动扩缩容视为一个自动控制问题，其目标是学习最优的自动伸缩策略，以便针对每个观测到的状态采取最佳资源供给操作。 [217] 对基于强化学习的自动扩缩容方法进行了全面综述，并根据一组提出的分类体系对这些方法进行了比较。对于对这一方向感兴趣的开发者或研究人员来说，这篇综述非常值得一读。尽管强化学习在自动扩缩容方面前景广阔，但仍有许多问题亟待解决。例如，基于模型的方法需要对环境建立完美模型，且学到的策略无法适应环境变化；而无模型方法初期性能很差、收敛速度慢，因此应用于实际云平台时会带来高昂成本。

C. 资源管理

问题定义

资源管理是云计算中的另一项重要主题，涵盖资源供给、分配和调度，例如工作负载估算、任务调度、能耗优化等。即使是微小的资源供给低效问题，例如为任务选择了错误的资源，也会影响服务质量（QoS），进而造成巨大的经济损失。因此，资源管理的目标是为任务提供适量的资源，以提升服务质量、缓解负载不均衡，并避免违反服务级别协议。

由于多个租户在云平台上共享存储和计算资源，资源管理是一项困难的任务，需要动态分配资源并调度租户的任务。如何配置资源，可以通过响应式方式确定，例如基于领域知识手动创建静态规则。但与自动扩展类似，响应式方法会导致响应延迟和过多开销。为了解决这一问题，基于机器学习的资源管理方法近来受到广泛关注。

基于机器学习的资源管理

近年来，人们开发了许多基于机器学习的资源管理方法。受篇幅所限，我们不再详细讨论。我们建议对这一研究主题感兴趣的读者阅读以下优秀的综述论文：[218], [219], [220], [221], [222]。这些方法大多采用机器学习技术预测未来的资源消耗，然后根据预测结果进行资源供给或调度。例如， [223] 使用随机森林和 XGBoost 预测虚拟机的行为，包括最大部署规模和工作负载。 [224] 提出了一种基于线性回归的方法，根据虚拟机的历史数据预测其资源利用率，并利用预测结果降低能耗。[225]采用梯度提升模型预测温度，并据此开发了一种动态调度算法，以尽量降低主机的峰值温度。 [226]提出了一种基于强化学习、针对特定工作负载的调度算法，以尽量缩短平均任务完成时间。

机器学习模型的准确性是影响资源管理系统效率的关键因素。应用更复杂的传统机器学习模型，甚至深度学习模型来提高预测准确性，是一个很有前景的研究方向。除了准确性之外，模型预测的时间复杂度也是需要考虑的另一个重要因素。如果机器学习模型过于复杂，就无法处理资源分配和调度的实时请求。如何在准确性和时间复杂度之间取得权衡，还需要进一步探索。

VIII. 智能运维的未来

A. 智能运维面临的常见人工智能挑战

我们根据如何运用人工智能技术，讨论了各任务章节中的挑战和未来趋势。总的来说，不同的AIOps任务面临一些共同挑战。

数据质量。 所有智能运维任务都存在数据质量问题。由于事件只会偶尔发生，大多数真实世界中的智能运维数据极度不平衡。此外，大多数真实世界中的智能运维数据噪声很大。在将数据用作训练机器学习模型的输入之前，需要投入大量精力进行数据清洗和预处理。

缺乏标签。 要充分获取高质量的标签极其困难。我们需要大量非常熟悉系统运维的领域专家，对事件、根因和服务图进行评估，以提供高质量的标签。这一过程极其耗时，并且需要特定的专业知识，无法通过 Mechanical Turk 等通用众包方式完成。

非平稳性和异质性。 系统在不断变化。智能运维面临的是非平稳的问题空间。该领域的人工智能模型需要具备应对这种非平稳性的机制。同时，智能运维数据具有异质性，这意味着相同的遥测数据可能表现出多种不同的潜在行为。例如，当资源用于承载不同的应用程序时，CPU 利用率模式可能完全不同。因此，发现隐藏状态并处理异质性，对于智能运维解决方案取得成功至关重要。

缺乏公开基准测试。尽管 AIOps 研究社区正在快速发展，但可供研究人员用于基准测试和评估研究结果的公开数据集仍然非常有限。运维数据高度敏感。现有研究要么使用模拟数据，要么使用难以与其他团队和组织共享的企业生产数据。

人在回路。人工反馈对于构建智能运维解决方案至关重要。目前，大多数人工反馈都是以临时方式收集的，效率低下。AIOps领域缺乏人在回路方面的研究，无法实现反馈收集自动化，也未能利用反馈来提升模型性能。

B. 机遇与未来趋势

我们对现有智能运维研究的文献综述表明，目前的研究仍然更侧重于基础设施和工具。人工智能技术已成功应用于事件检测、根因分析等应用程序，一些解决方案也已被 AWS、阿里云等大型分布式系统采用。然而，智能运维进程标准化和全面自动化仍处于非常早期的阶段。基于这些证据，我们可以预见未来几年智能运维领域的热门研究主题。

高质量智能运维基础设施与工具

近年来，已有一些成功的智能运维平台和工具投入开发。但人工智能仍有机会帮助提升 IT 运维的效率。人工智能也在快速发展，新的人工智能技术不断涌现，并已成功应用于其他领域。数字化转型趋势也给传统 IT 运维和 DevOps 带来了挑战。因此，对高质量人工智能工具的需求十分巨大，包括监控、检测、根因分析、预测和自动化。

智能运维标准化

在构建基础设施和工具的同时，智能运维专家也能更好地了解整个领域的全貌。可以从传统流程中识别并提取智能运维模块，形成自身的标准。通过明确的目标和衡量指标，可以将智能运维系统标准化，就像推荐系统或自然语言处理等领域所做的那样。通过这种标准化，将更容易尝试各种人工智能技术，以提升智能运维的性能。

从以人为中心转向以机器为中心的智能运维

以人为中心的智能运维意味着，人的流程在整个智能运维生态系统中仍发挥着关键作用，而人工智能模块则帮助人们做出更好的决策并更有效地执行。在以机器为中心的模式下，智能运维系统只需最少的人工干预，并且在其大部分生命周期内都可以处于无人参与状态。智能运维系统会持续监控 IT 基础设施，检测并分析问题，寻找合适的路径来推动修复。在这一阶段，工程师主要专注于开发任务，而不是运维。

IX. 结论

数字化转型带来了对计算资源的巨大需求。这一趋势推动了大型 IT 基础设施的快速增长，例如云计算、边缘计算、搜索引擎等。自 Gartner 于 2016年提出智能运维以来，它迅速兴起，如今已引起大型企业和组织的关注。随着 IT 基础设施规模扩大到人工运维无法跟上的程度，智能运维成为保障这些庞大 IT 基础设施高可用性的唯一可行解决方案。智能运维涵盖软件生命周期的不同阶段，包括开发、测试、部署和维护。

不同的人工智能技术现已应用于智能运维应用中，包括异常检测、根因分析、故障预测、自动化操作和资源管理。然而，整个智能运维工业界仍处于非常早期的阶段，人工智能目前仅发挥辅助作用，帮助人类执行运维工作流。我们预见，在不久的将来，运维将从以人为中心转向以人工智能为中心。在这一转变过程中，智能运维技术的开发也将从构建工具转向创建无需人工参与的端到端解决方案。

在本综述中，我们发现目前大多数智能运维成果都侧重于检测和根因分析，而有关自动化的研究工作仍然非常有限。智能运维中使用的人工智能技术主要是传统机器学习和统计模型。

致谢

我们要感谢所有抽出时间完成这项调查的参与者。他们在人工智能基础知识方面的知识和经验对我们的研究弥足珍贵。我们也感谢 Salesforce AI Research Lab 的同事以及其他组织的合作者所提供的宝贵反馈与支持。

附录 A 术语

DevOps： 现代软件开发不仅要求提高开发质量，也要求提高运维质量。DevOps 是一套结合开发（Dev）和运维（Ops）进程的最佳实践，旨在实现高质量的软件开发和发布后的管理 [3]。

应用性能监控（APM）： 应用程序性能监控是指利用监控软件跟踪应用程序的关键性能指标。

以及遥测数据[227]。应用性能监控用于确保系统具有高可用性、优化服务性能并改善用户体验。最初，应用性能监控主要用于网站、移动应用和其他类似的在线业务应用程序。然而，随着越来越多的传统软件转向采用基于云的高度分布式系统，应用性能监控如今已广泛应用于种类更多的软件应用程序和后端系统。

可观测性：可观测性是通过检查系统的输出，测量系统内部状态的能力 [228]。如果仅使用输出中的信息就能估计当前状态，则该系统是“可观测的”。可观测性数据包括指标、日志、追踪数据以及系统生成的其他信息。

云智能：提升云应用的人工智能特征。

MLOps: MLOps代表机器学习运维。MLOps是将机器学习模型部署到生产环境的完整进程生命周期。

站点可靠性工程（SRE）：弥合软件开发与运维之间鸿沟的工程。

云计算： 云计算是一种技术，也是一种商业模式，通过构建高度可扩展的分布式计算机系统，并将计算资源（例如主机、平台、应用）出租给租户来创造收入。云计算主要分为三类：基础设施即服务（IaaS）、平台即服务（PaaS）和软件即服务（SaaS）

信息技术服务管理（ITSM）：ITSM 是指为客户设计、创建、交付和支持信息技术服务的所有进程和活动。

IT 运维管理（ITOM）：ITOM 与 ITSM 有重叠，但更侧重于 IT 服务和基础设施的运维方面。

附录 B 表格

表 I 指标可观测性常用公开数据集表

Name	描述	任务
Azure 公开数据集	这些数据集包含第一方数据的代表性子集来自某个地理区域的 Azure 线上虚拟机工作负载。	工作负载特征分析，VM 预配置，工作负载预测
Google 集群 Data	来自 Google Borg 单元的连续 30 天信息。	工作负载特征分析，工作负载预测
阿里巴巴集群追踪	来自阿里巴巴集团的真实生产服务器集群追踪数据。	工作负载特征分析，工作负载预测
MIT 超级CLOUD 数据集	高层数据的组合（例如 Slurm 工作负载管理器调度器数据）和低级别的特定作业时间序列数据。	工作负载特征描述
Numenta 异常基准测试（re-alAWSCloud-watch）	AmazonCloudwatch 收集的 AWS 服务器指标服务。例如，指标包括 CPU 利用率、网络输入字节数和磁盘读取字节数。	事件检测
Yahoo S5 (A1)	A1基准测试包含真实的 Yahoo! 网页流量指标。	事件检测
服务器机器数据集	从一家大型互联网公司收集的为期 5 周的数据集包含 CPU 负载、网络使用量、内存使用量等指标。	事件检测
KPI 异常检测数据集 A	一个大规模的真实世界 KPI 异常检测数据集，涵盖各种 KPI 模式和异常模式。该数据集收集自五家大型互联网公司（搜狗、eBay、百度、腾讯和阿里巴巴）。	事件检测

表 II 日志可观测性常用公开数据集表

数据集	描述	时间跨度	数据规模	# 日志	异常标签	# 异常	# 日志模板
分布式系统日志							
HDFS	Hadoop 分布式文件系统日志	38.7 小时	1.47 GB	11,175,629	✓	16,838（块）	30
		N.A.	16.06 GB	71,118,073	✗		
Hadoop	Hadoop MapReduce 作业日志	N.A.	48.61MB	394,308	✓		298
Spark	Spark 作业日志	N.A.	2.75GB	33,236,604	✗		456
ZooKeeper	ZooKeeper 服务日志	26.7 天	9.95MB	74,380	✗		95
OpenStack	OpenStack 基础设施日志	N.A.	58.61MB	207,820	✓	503	51
超级计算机日志							
BGL	Blue Gene/L 超级计算机日志	214.7 天	708.76MB	4,747,963	✓	348,460	619
HPC	高性能集群日志	N.A.	32MB	433,489	✗		104
Thunderbird	Thunderbird 超级计算机日志	244 天	29.6GB	211,212,192	✓	3,248,239	4040
操作系统日志							
Windows	Windows 事件日志	226.7 天	16.09GB	114,608,388	✗		4833
Linux	Linux 系统日志	263.9 天	2.25MB	25,567	✗		488
Mac	Mac OS 日志	7天	16.09MB	117,283	✗		2214
移动系统日志							
安卓	Android 框架日志	N.A.	183.37MB	1,555,005	✗		76,923
健康应用程序	健康应用日志	10.5天	22.44MB	253,395	✗		220
服务器应用程序日志							
Apache	Apache 服务器错误日志	263.9 天	4.9MB	56,481	✗		44
OpenSSH	OpenSSH 服务器日志	28.4 天	70.02MB	655,146	✗		62
独立软件日志							
Proxifier	Proxifier 软件日志	N.A.	2.42MB	21,329	✗		9
硬件日志							
交换机	交换机硬件故障	2 年	-	29,174,680	✓	2,204	-

表 III 现有日志异常检测模型的比较

参考文献	学习设置	模型类型	日志表示	日志词元	解析	序列建模
[92], [93], [94]	有监督	线性回归、支持向量机、决策-决策树	手工设计特征	日志模板	✓	✗
[84]	无监督	主成分分析（PCA）	定量	日志模板	✓	✓
[67], [82], [95], [80]	无监督	聚类与相关性日志与指标之间	序列化、定量	日志模板	✓	✗
[96]	无监督	使用奇异-挖掘不变量奇异值分解	定量, 序列	日志模板	✓	✗
[97], [98], [99], [68]	无监督	从中挖掘频繁模式执行 Flow and 控制流图挖掘	定量、序列	日志模板	✓	✗
[20], [100]	无监督	基于集成方法的规则引擎和基于异常特征的启发式对比分析针对异常特征	序列式（带有 TF-IDF权重）	日志模板	✓	✗
[101]	有监督	日志专用自编码器 word2vec	语义（可训练嵌入）	日志模板	✓	✓
[102]	无监督	自编码器与孤立森林	语义（可训练嵌入）	所有词元	✗	✗
[114]	有监督	卷积神经网络	语义（可训练嵌入）	日志模板	✓	✓
[108]	无监督	基于注意力机制的长短期记忆网络	顺序的, 定量, 语义（GloVe嵌入）	log 模板, 日志参数	✓	✓
[81]	无监督	基于注意力机制的长短期记忆网络	定量与语义（GloVe 嵌入-床上用品）	日志模板	✓	✓
[111]	有监督	基于注意力机制的长短期记忆网络	语义（使用 fastText嵌入和 tf-逆文档频率权重）	日志模板	✓	✓
[104]	半监督 有监督	基于注意力机制的 GRU 与聚类处理	语义（使用 fastText嵌入与 tf-idf 权重)	日志模板	✓	✓
[112]	无监督	基于注意力机制的双向长短期记忆网络	语义（带有可训练嵌入）	所有词元	✗	✓
[109]	无监督	双向长短期记忆网络	语义（来自 BERT 的词元嵌入, GPT、XLM）	所有词元	✗	✓
[113]	无监督	基于注意力的双向长短期记忆网络	语义（BERT词元嵌入）	日志模板	✓	✓
[110]	半监督 有监督	长短期记忆网络，经过监督训练来自源系统	语义（GloVe嵌入）	日志模板	✓	✓
[18]	无监督	结合领域对抗的长短期记忆网络训练	语义（GloVe嵌入）	所有词元	✗	✓
[118], [18]	无监督	长短期记忆网络与深度支持向量-tor 数据描述	语义（可训练嵌入）	日志模板	✓	✓
[115]	有监督	图神经网络	语义（BERT词元嵌入）	日志模板	✓	✓
[116]	半监督 有监督	图神经网络	语义（BERT词元嵌入）	日志模板	✓	✓
[103], [229], [230], [231]	无监督	自注意力 Transformer	语义（可训练嵌入）	所有词元	✗	✓
[78]	有监督	自注意力 Transformer	语义（可训练嵌入）	所有词元	✗	✓
[117]	有监督	分层 Transformer	语义（可训练的GloVe嵌入）	log 模板, 日志参数	✓	✓
[104], [105]	无监督	BERT 语言模型	语义（BERT词元嵌入）	所有词元	✗	✓
[21]	无监督	各种日志上的统一 BERT 分析任务	语义（BERT词元嵌入）	所有词元	✗	✓
[232]	无监督	对比对抗模型	语义（基于 BERT 和变分自编码器的情感分析。嵌入）和定量	日志模板	✓	✓
[106], [107], [233]	无监督	基于长短期记忆网络、Transformer 的生成对抗网络（生成对抗）	语义（可训练嵌入）	日志模板	✓	✓

日志词元 指日志表示中使用的日志行词元；**解析** 和**序列建模**列分别表示这些模型是否需要进行日志解析，以及是否支持对日志序列进行建模。

表 IV 现有指标异常检测模型的比较

参考资料	标签可获取性	机器学习模型	维度	基础设施	流式更新
[31]	有监督	Tree	单变量	✗	✓(重新训练)
[41]	主动式	-	单变量	✓	✓(重新训练)
[42]	无监督	Tree	多变量	✗	✓
[43]	无监督	统计	单变量	✗	✓
[51]	无监督	统计	单变量	✗	✗
[37]	半监督	Tree	单变量	✗	✓
[36]	无监督, 半监督 有监督	深度学习	单变量	✗	✗
[52]	无监督	深度学习	单变量	✓	✗
[40]	域 适应, 主动式	Tree	单变量	✗	✗
[46]	无监督	深度学习	多变量	✗	✗
[49]	无监督	深度学习	单变量	✗	✗
[45]	无监督	深度学习	多变量	✗	✗
[32]	有监督	深度学习	单变量	✓	✓(重新训练)
[47]	无监督	深度学习	多变量	✗	✗
[48]	无监督	深度学习	多变量	✗	✗
[50]	无监督	深度学习	多变量	✗	✗
[38]	半监督, 主动式	深度学习	多变量	✓	✓(重新训练)

表 V 现有追踪、多模态异常检测与根因分析模型对比

参考资料	主题	深度学习应用	方法
[124]	跟踪根因分析	✗	聚类
[121]	跟踪根因分析	✗	启发式方法
[234]	跟踪根因分析	✗	多输入差分和-汇总
[197]	跟踪根因分析	✗	随机森林，k-NN
[122]	跟踪根因分析	✗	启发式方法
[235]	追踪异常检测	✗	图模型
[198]	多模态异常检测	✓	深度贝叶斯网络
[236]	追踪表示	✓	基于树的 RNN
[196]	追踪异常检测	✗	启发式方法
[120]	多模态异常检测	✓	门控图神经网络和SVDD

表 VI 若干现有指标根因分析方法的比较

参考文献	指标或图表分析	根因评分
[147]	变点	按时间顺序
[146]	变点	按时间顺序
[148]	双样本检验	相关性
[149]	调用图	簇相似度
[150]	服务图	PageRank
[151]	服务图	图相似度
[152]	服务图	层次隐马尔可夫模型
[153]	PC算法	随机游走
[154]	ITOA-PI	PageRank
[155]	服务图与PC	因果推断
[156]	PC算法	随机游走
[157]	服务图与PC	因果推断
[158]	PC算法	随机游走
[159]	PC算法	随机游走
[237]	服务图	因果推断
[168]	服务图	基于贡献的

参考文献

[1] T.Olavsrud，“如何选择云服务提供商”，2012年。 [在线]。可访问：https://www2.cio.com.au/article/416752/ how chooseyourcloud service provider/ [2] “北弗吉尼亚（us-east-1）区域Amazon S3 服务中断总结”，2021年。 [在线]。可访问：https://aws.amazon.com/message/41926/ [3] S.Gunja，“什么是DevOps？ 解析信息技术文化变革的目的和重要性”，2021年。 [在线]。可访问：https: // www.dynatrace.com/news/blog/what-is-devops/ [4] Gartner，“智能运维（人工智能 IT 运维）”。 [在-线]。可访问：https://www.gartner.com/en/information-technology/glossary/aiops-artificial-intelligence-operations [5] S. Siddique，“迈向企业人工智能：以案例研究为驱动的探索”，博士学位论文，05 2018年。[6] N. Sabharwal，《Hands-on AIOps》。Springer，2022年。[7] Y. Dang、Q. Lin和P. Huang，“智能运维：现实世界的挑战与研究创新”，载于《2019年第41届IEEE/ACM国际软件工程会议：ICSE-Companion论文集》，2019年，第4–5 页。[8] L.Rijal、R. Colomo-Palacios 和M.S´anchez-Gord´on，“智能运维：多声部文献综述”，《云计算与边缘计算中的人工智能》，第 31–50 页，2022年。[9] R.Chalapathy和 S.Chawla，“用于异常检测的深度学习：综述”，arXiv预印本 arXiv:1901.03407，2019年。[10] L.Akoglu、H.Tong 和D. Koutra，“基于图的异常检测与描述：综述”，《数据挖掘与知识发现》，第29卷，第 3 期，第626–688 页，2015年。[11] J. Soldani和 A.Brogi，“（微）服务型云应用中的异常检测与故障根因分析：综述”，2021年。 [ ] 2105 12378 在线 。可访问：https://arxiv.org/abs/ 。V. Davidovski，“通过数字化转型实现指数级创新”，载于《第三届信息技术应用国际会议论文集》，系列号 ICAIT’2018。美国纽约州纽约市：计算机协会，2018年，第 3–5 页。 [ ] 10 1145 3274856 3274858 在线。可访问：https://doi.org/ 。/ 。D. S.Battina，“美国信息技术领域的人工智能与 DevOps 及其未来”，《国际创意研究思想期刊》（IJCRT），ISSN，第 2320–2882 页，2021年。[14] A. B. Yoo、M. A. Jette 和 M. Grondona，“Slurm：用于资源管理的简单 Linux 实用程序”，载于《并行处理作业调度策略研讨会》。Springer，2003 年，第44–60 页。[15] J. Zhaoxue、L. Tong、Z. Zhenguo、G. Jingguo、Y. Junling 和 L. Liangxiong，“智能运维日志研究综述：方法与趋势”，Mob.Netw. Appl.，第 26 卷，第 6期，第 2353–2364 页，2021年 12 月。 [ ] 10 1007 11036 021 01832 3 在线 。可访问：https://doi.org/。 /s --- S. He、P.He、Z. Chen、T.Yang、Y.Su 和 M. R. Lyu，“面向可靠性工程的自动化日志分析综述”，ACMComput. Surv.，第 54卷，第 6 期，2021年 7 月。 [在线]。可访问：https://doi.org/ 10 1145 3460345 。 /P.Notaro、J. Cardoso 和 M. Gerndt，“故障管理的智能运维方法综述”，ACMTrans.Intell. Syst. Technol.，第 12 卷，第 6， 2021 [ ] 10 1145 3483424 11 月。在线 。可访问：https://doi.org/。 / X. Han 和 S.Yuan，“通过领域自适应进行无监督跨系统日志异常检测”，载于《第30届ACM国际信息与知识管理会议论文集》，系列号CIKM’21。美国纽约州纽约市：计算机协会，2021年，第 3068–3072 页。 [在线]。可访问：https://doi.org/10.1145/3459637.3482209 [19] V.-H. Le 和 H.Zhang，“基于日志的深度学习异常检测：我们距离目标还有多远？ ”，载于《第44届国际软件工程会议论文集》，系列号ICSE’22。美国纽约州纽约市：计算机协会，2022年，第 1356–1367 页。 [ ] 10 1145 3510003 3510155 在线 。可访问：https://doi.org/ 。 /。N. Zhao、H. Wang、Z.Li、X. Peng、G. Wang、Z. Pan、Y. Wu、Z.Feng、X. Wen、W. Zhang、K. Sui和D. Pei，“在线服务系统实用日志异常检测的实证研究”，载于《第29届ACM欧洲软件工程会议与软件工程基础研讨会联合会议论文集》，系列号 ESEC/FSE2021。美国纽约州纽约市：计算机协会，2021年，第 1404–1415 页。 [在线]。可访问：https://doi.org/ 10 1145 3468264 3473933 。 /。Y. Zhu、W. Meng、Y. Liu、S. Zhang、T. Han、S. Tao 和 D. Pei，“UniLog：部署一个模型并使其专用于所有日志分析任务”，CoRR，第abs/2112.03159 卷，2021年。 [在线]。可访问：https://arxiv.org/abs/2112.03159

[22] J. Soldani 和 A. Brogi，《基于（微）服务的云应用中的异常检测与故障根因分析：一项调查》，《ACM计算调查》，第 55 卷，第 3 期，2022年 2 月。 [线上]。可访问：https://doi.org/ 10 1145 3501297 。/ L. Korzeniowski 和 K. Goczyla，《自动化日志分析的概况：系统性文献综述与映射研究》，IEEE Access，第 10 卷，第 21 892–21 913 页，2022年。[24] M. Sheldon 和 G. V. B. Weissman，《Retrace：使用虚拟机确定性重放收集执行追踪》，载于《第三届建模、基准测试与仿真年度研讨会论文集》（MoBS 2007）。Citeseer，2007年。[25] R. Fonseca、G. Porter、R. H. Katz 和 S. Shenker，《{X-Trace}：一种普适网络追踪框架》，载于第四届 USENIX 网络系统设计与实现研讨会（NSDI 07），2007年。[26] J. Zhou、Z. Chen、J. Wang、Z. Zheng 和 M. R. Lyu，《Trace bench：面向追踪监控的开放数据集》，载于《2014年 IEEE 第六届云计算技术与科学国际会议》。IEEE，2014年，第 519–526 页。[27] S. Zhang、C. Zhao、Y. Sui、Y. Su、Y. Sun、Y. Zhang、D. Pei 和 Y. Wang，《利用部分标签对大规模软件服务进行稳健 KPI 异常检测》，载于《第 32 届 IEEE 软件可靠性工程国际研讨会（ISSRE 2021）》，中国武汉，2021年 10 月 25–28 日，Z. Jin、X. Li、J. Xiang、L. Mariani、T. Liu、X. Yu 和 N. Ivaki 编。IEEE，2021年，第 103–114 页。 [线上]。可访问：https://doi.org/ 10 1109 52982 2021 00023 。/ISSRE . . M. Braei 和 S. Wagner，《单变量时间序列异常检测：最先进技术综述》，ArXiv，第 abs/2004.00433 卷，2020年。[29] A. Bl´azquez-Garc´ıa、A. Conde、U. Mori 和 J. A. Lozano，《时间序列数据中的离群值/异常检测综述》，《ACM计算调查》（CSUR），第 54 卷，第 3 期，第 1–33 页，2021年。[30] K. Choi、J. Yi、C. Park 和 S. Yoon，《时间序列数据异常检测的深度学习：综述、分析与指南》，IEEE Access，2021年。[31] D. Liu、Y. Zhao、H. Xu、Y. Sun、D. Pei、J. Luo、X. Jing 和 M. Feng，《Opprentice：借助机器学习实现实用且自动化的异常检测》，载于《2015年互联网测量会议论文集》，2015年，第 211–224 页。[32] J. Gao、X. Song、Q. Wen、P. Wang、L. Sun 和 H. Xu，《Robusttad：通过分解和卷积神经网络实现稳健的时间序列异常检测》，arXiv 预印本 arXiv:2002.09545，2020年。[33] S. Han、X. Hu、H. Huang、M. Jiang 和 Y. Zhao，《ADBench：异常检测基准》，载于《第三十六届神经信息处理系统会议数据集与基准测试专题》，2022年。 [ ] = 9 0 线上。可访问：https://openreview.net/forum?id foA SFQ zo Z. Li、N. Zhao、S. Zhang、Y. Sun、P. Chen、X. Wen、M. Ma 和 D. Pei，《构建大规模真实世界智能运维基准数据集》，arXiv 预印本 arXiv:2208.03938，2022年。[35] R. Wu 和 E. J. Keogh，《当前的时间序列异常检测基准存在缺陷，正在制造进步的假象》，CoRR，第 abs/2009.13807 卷，2020年。 [线上]。可访问：https://arxiv. org/abs/ 2009 13807 。H. Xu、W. Chen、N. Zhao、Z. Li、J. Bu、Z. Li、Y. Liu、Y. Zhao、D. Pei、Y. Feng 等，《通过变分自编码器对 Web 应用中的季节性 KPI 进行无监督异常检测》，载于《2018年世界互联网大会论文集》，2018年，第 187–196 页。[37] J. Bu、Y. Liu、S. Zhang、W. Meng、Q. Liu、X. Zhu 和 D. Pei，《为大量新兴 KPI 数据流快速部署异常检测模型》，载于《2018年 IEEE 第 37 届国际性能计算与通信会议（IPCCC）》。IEEE，2018年，第 1–8 页。[38] T. Huang、P. Chen 和 R. Li，《面向线上系统多变量时间序列的基于变分自编码器的半监督主动异常检测框架》，载于《2022年 ACM Web 会议论文集》，2022年，第 1797–1806 页。[39] X.-L. Li 和 B. Liu，《从具有不同数据分布的正例和未标记样例中学习》，载于《欧洲机器学习会议》。Springer，2005年，第 218–229 页。[40] X. Zhang、J. Kim、Q. Lin、K. Lim、S. O. Kanaujia、Y. Xu、K. Jamieson、A. Albarghouthi、S. Qin、M. J. Freedman 等，《云系统的跨数据集时间序列异常检测》，载于《2019年 USENIX 年度技术会议（USENIX ATC 19）》，2019年，第 1063–1076 页。[41] N. Laptev、S. Amizadeh 和 I. Flint，《用于自动时间序列异常检测的通用可扩展框架》，载于《第 21 届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集》，2015年，第 1939–1947 页。[42] S. Guha、N. Mishra、G. Roy 和 O. Schrijvers，《基于稳健随机切割森林的流数据异常检测》，载于《国际机器学习会议》。PMLR , 2016, 第2712–2721页。

[43] S. Ahmad、A. Lavin、S. Purdy 和 Z. Agha，《面向流数据的无监督实时异常检测》，Neurocomputing，第262卷，第134–147页，2017年。[44] Z. Li、Y. Zhao、J. Han、Y. Su、R. Jiao、X. Wen 和 D. Pei，《利用分层指标间与时间嵌入进行多变量时间序列异常检测与解释》，载于 KDD '21：第27届AC M SIGKDD知识发现与数据挖掘大会，线上活动，新加坡，2021年8月14日至18日，F. Zhu、B. C. Ooi 和 C. Miao 编。ACM，2021年，第3220–3230页。 [线上]。获取地址：https://doi.org/ 10 1145 3447548 3467075 。/。J. Audibert、P. [45] Michiardi、F. Guyard、S. Marti 和 M. A. Zuluaga，《Usad：多变量时间序列的无监督异常检测》，载于第26届ACM SIGKDD国际知识发现与数据挖掘大会论文集，2020年，第3395–3404页。[46] Y. Su、Y. Zhao、C. Niu、R. Liu、W. Sun 和 D. Pei，《通过随机循环神经网络实现多变量时间序列的鲁棒异常检测》，载于第25届ACM SIGKDD国际知识发现与数据挖掘大会论文集，2019年，第2828–2837页。[47] Z. Li、Y. Zhao、J. Han、Y. Su、R. Jiao、X. Wen 和 D. Pei，《利用分层指标间与时间嵌入进行多变量时间序列异常检测与解释》，载于第27届ACM SIGKDD知识发现与数据挖掘大会论文集，2021年，第3220–3230页。[48] W. Yang、K. Zhang 和 S. C. Hoi，《基于因果关系的多变量时间序列异常检测》，arXiv预印本 arXiv:2206.15033，2022年。[49] F. Ayed、L. Stella、T. Januschowski 和 J. Gasthaus，《大规模异常检测：深度分布式时间序列模型的应用案例》，载于国际面向服务计算大会。Springer，2020年，第97–109页。[50] S. Rabanser、T. Januschowski、K. Rasul、O. Borchert、R. Kurle、J. Gasthaus、M. Bohlke-Schneider、N. Papernot 和 V. Flunkert，《多变量时间序列的内在异常检测》，arXiv预印本 arXiv:2206.14342，2022年。[51] J. Hochenbaum、O. S. Vallis 和 A. Kejariwal，《通过统计学习在云端实现自动异常检测》，arXiv预印本 arXiv:1704.07706，2017年。[52] H. Ren、B. Xu、Y. Wang、C. Yi、C. Huang、X. Kou、T. Xing、M. Yang、J. Tong 和 Q. Zhang，《微软的时间序列异常检测服务》，载于第25届ACM SIGKDD国际知识发现与数据挖掘大会论文集，2019年，第3009–3017页。[53] J. Soldani 和 A. Brogi，《基于（微）服务的云应用中的异常检测与故障根因分析：综述》，ACM Comput. Surv.，第55卷，第3期，第59:1–59:39页，2023年。 [线上]。获取地址：https://doi.org/ 10 1145 3501297 。/ J. Bu、Y. Liu、S. Zhang、 [54] W. Meng、Q. Liu、X. Zhu 和 D. Pei，《面向大量新兴KPI数据流的异常检测模型快速部署》，载于第37届IEEE国际性能计算与通信大会，IPCCC 2018，美国佛罗里达州奥兰多，2018年11月17日至19日。IEEE，2018年，第1–8页。 [线上]。获取地址：https://doi.org/ 10 1109 2018 8711315 。/PCCC. 。Z. Z. [55] Darban、G. I. Webb、S. Pan、C. C. Aggarwal 和 M. Salehi，《时间序列异常检测的深度学习：综述》，CoRR，第abs/2211.05244卷，2022年。 [线上]。获取地址：https://doi.org/ 10 48550 2211 05244 。/ar [56] Xiv. 。B. Huang、K. Zhang、M. Gong 和 C. Glymour，《利用状态空间模型在非平稳环境中进行因果发现与预测》，载于第36届国际机器学习大会，I CML 2019，2019年6月9日至15日，美国加利福尼亚州长滩，机器学习研究论文集系列，K. Chaudhuri 和 R. Salakhutdinov 编，第97卷。PMLR，2019年，第2901–2910页。[线上]。获取地址：http://proceedings.mlr.press/v97/huang19g.html [57] Q. Pham、C. Liu、D. Sahoo 和 S. C. H. Hoi，《在线时间序列预测中的快慢学习》，CoRR，第abs/ 2202 11672, 卷。. 线上。获取地址：https://arxiv.org/abs/。 2022 [ ] 2202 11672 [58] K. Lai、D. Zha、J. Xu、Y. Zhao、G. Wang 和 X. Hu，《重新审视时间序列离群点检测：定义与基准测试》，载于神经信息处理系统数据集与基准专题论文集第1卷，NeurIPS数据集与基准2021，2021年12月，线上活动，J. Vanschoren 和 S. Yeung 编，2021年。 [线上]。获取地址：https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/ec5decca5ed3d6b8079e2e7e7bacc9f2-Abstract-round1.html [59] R. Wu 和 E. Keogh，《当前时间序列异常检测基准存在缺陷，造成进展的假象》，IEEE知识与数据工程汇刊，2021年。

[60] X. Wu、L. Xiao、Y. Sun、J. Zhang、T. Ma 和 L. He，《人在回路机器学习综述》，Future Gener. Comput. Syst.，第 135 卷，第 364–381 页，2022年。 [在线]。获取地址：https://doi.org/ 10 1016 2022 05 014 。/j.future. . . D. Sahoo、Q. [61] Pham、J. Lu 和 S. C. H. Hoi，《在线深度学习：即时学习深度神经网络》，CoRR，第 abs/ 1711 03705，卷。在线。2017 [ ] 1711 03705 获取地址：http://arxiv.org/abs/ 。 [62] Z. Chen、J. Liu、W. Gu、Y. Su 和 M. R. Lyu，《经验报告：基于深度学习的系统日志异常检测分析》， 2021 [ ] 2107 05908 。在线。获取地址：https://arxiv.org/abs/ 。P. He、J. Zhu、Z. Zheng 和 M. R. Lyu，[63] 《Drain：一种基于固定深度树的在线日志解析方法》，载于 2017 年 IEEE Web 服务国际会议（ICWS），2017年，第 33–40 页。[64] A. A. Makanju、A. N. Zincir-Heywood 和 E. E. Milios，《使用迭代划分对事件日志进行聚类》，载于第 15 届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集，系列：KDD '09。美国纽约：计算机协会，2009年，第 1255–1264 页。 [在线]。获取地址：https://doi.org/ 10 1145 1557019 1557154 。/ 。Z. [65] M. Jiang、A. E. Hassan、P. Flora 和 G. Hamann，《将执行日志抽象为企业应用程序的执行事件（短文）》，载于 2008 年第八届国际软件质量会议，2008年，第 181–186 页。[66] M. Du 和 F. Li，《Spell：系统事件日志的流式解析》，载于 2016 年 IEEE 第 16 届数据挖掘国际会议（ICDM），2016年，第 859–864 页。[67] R. Vaarandi 和 M. Pihelgas，《Logcluster——一种用于事件日志的数据聚类与模式挖掘算法》，载于第 11 届网络与服务管理国际会议（CNSM 2015），西班牙巴塞罗那，2015年11月9日至13日，M. Tortonesi、J. Sch¨onw¨alder、E. R. M. Madeira、C. Schmitt 和 J. Serrat 编。IEEE计算机学会，2015年，第 1–7 页。 [在线]。获取地址：https: //doi.org/ 10 1109 2015 7367331 。/CNSM. . Q. Fu、J.-G. Lou、Y. [68] Wang 和 J. Li，《通过非结构化日志分析检测分布式系统中的执行异常》，载于 2009 年第九届 IEEE 数据挖掘国际会议，2009年，第 149–158 页。[69] L. Tang、T. Li 和 C.-S. Perng，《Logsig：从原始文本日志生成系统事件》，载于第 20 届 ACM 信息与知识管理国际会议论文集，系列：CIKM '11。美国纽约：计算机协会，2011年，第 785–794 页。 [在线]。获取地址：https://doi.org/ 10 1145 2063576 2063690 。/ 。M. Mizutani，《系统日志格 [70] 式的增量挖掘》，载于 2013 年 IEEE 国际服务计算会议，2013年，第 595–602 页。[71] K. Shima，《长度很重要：根据词语长度对系统日志消息进行聚类》，CoRR，第 abs/1611.03213 卷，2016年。 [在线]。获取地址：http://arxiv.org/abs/ 1611 03213 。H. [72] Hamooni、B. Debnath、J. Xu、H. Zhang、G. Jiang 和 A. Mueen，《Logmine：面向日志分析的快速模式识别》，载于第 25 届 ACM 信息与知识管理国际会议论文集，系列：CIKM '16。美国纽约：计算机协会，2016年，第 1573–1582 页。 [在线]。获取地址：https://doi.org/ 10 1145 2983323 2983358 。/ 。R. Vaarandi，《一 [73] 种从事件日志中挖掘模式的数据聚类算法》，载于第 3 届 IEEE IP 运维与管理研讨会论文集（IPOM 2003）（IEEE 编号 03EX764），2003年，第 119–126 页。[74] M. Nagappan 和 M. A. Vouk，《将日志行抽象为日志事件类型，以挖掘软件系统日志》，载于 2010 年第 7 届 IEEE 软件仓库挖掘工作会议（MSR 2010），2010年，第 114–117 页。[75] S. Messaoudi、A. Panichella、D. Bianculli、L. Briand 和 R. Sasnauskas，《一种用于准确识别日志消息格式的基于搜索的方法》，载于第 26 届程序理解会议论文集，系列：ICPC '18。美国纽约：计算机协会，2018年，第 167–177 页。 [在线]。获取地址：https://doi.org/ 10 1145 3196321 3196340 。/ 。S. [76] Nedelkoski、J. Bogatinovski、A. Acker、J. Cardoso 和 O. Kao，《自监督日志解析》，载于《数据库中的机器学习与知识发现：应用数据科学方向》，Y. Dong、D. Mladeni´c 和 C. Saunders 编。Cham：Springer国际出版公司，2021年，第 122–138 页。[77] Y. Liu、X. Zhang、S. He、H. Zhang、L. Li、Y. Kang、Y. Xu、M. Ma、Q. Lin、Y. Dang、S. Rajmohan 和 D. Zhang，《UniParser：一种用于异构日志数据的统一日志解析器》，载于 2022 年 ACM Web 会议论文集 ，系列。WWW '22。纽约 ，NY，USA:

计算机协会，2022年，第 1893–1901 页。 [在线]。获取地址：https://doi.org/ 10 1145 3485447 3511993 。/。V.-H. Le 和 H. Zhang，“无需日志解析的基于日志的异常检测”，载于《2021年第36届 IEEE/ACM 自动化软件工程国际会议（ASE）》，2021年，第 492–504 页。[79] Y. Lee、J. Kim 和 P. Kang，“Lanobert：基于 BERT 掩码语言模型的系统日志异常检测”，CoRR，第 abs/ 2111 09564，卷。在线。获取地址：https://arxiv.org/abs/。 2021 [ ] 2111 09564
[80] M. Farshchi、J.-G. Schneider、I. Weber 和 J. Grundy，“经验报告：使用日志和云指标相关性分析检测云应用程序运维中的异常”，载于《2015年 IEEE 第26届软件可靠性工程国际研讨会（ISSRE）》，2015年，第 24–34 页。[81] W. Meng、Y. Liu、Y. Zhu、S. Zhang、D. Pei、Y. Liu、Y. Chen、R. Zhang、S. Tao、P. Sun 和 R. Zhou，“Loganomaly：对非结构化日志中的序列异常和定量异常进行无监督检测”，载于《第28届国际人工智能联合会议论文集》，系列：IJCAI’19。AAAI Press，2019年，第 4739–4745 页。[82] Q. Lin、H. Zhang、J.-G. Lou、Y. Zhang 和 X. Chen，“面向在线服务系统的问题识别：日志聚类方法”，载于《第38届国际软件工程会议伴随论文集》，系列：ICSE ’16。美国纽约州纽约：计算机协会，2016年，第 102–111 页。 [在线]。获取地址：https://doi.org/ 10 1145 2889160 2889232 。/。R. Yang、D. Qu、Y. Qian、Y. Dai 和 S. Zhu，“一种基于层次聚类的在线日志模板提取方法”，《EURASIP 无线通信与网络杂志》，第 2019 卷，第 135 页，2019年。 [在线]。获取地址：https://doi.org/ 10 1186 13638 019 1430 4 。/s - - - W. Xu、L. Huang、A. Fox、D. A. Patterson 和 M. I. Jordan，“通过挖掘控制台日志检测大规模系统问题”，载于《第22届 ACM 操作系统原理研讨会论文集（SOSP 2009）》，2009年10月11—14日，美国蒙大拿州比格斯凯，J. N. Matthews 和 T. E. Anderson 编。ACM，2009年，第 117–132 页。 [ ] 10 1145 1629575 1629587 在线。获取地址：https://doi.org/。/。B. Joshi、U. Bista 和 M. Ghimire，“面向日志数据流的智能聚类方案”，载于《计算语言学与智能文本处理》，A. Gelbukh 编。柏林、海德堡：Springer Berlin Heidelberg，2014年，第 454–465 页。[86] J. Liu、J. Zhu、S. He、P. He、Z. Zheng 和 M. R. Lyu，“Logzip：通过迭代聚类提取隐藏结构以实现日志压缩”，载于《第34届 IEEE/ACM 自动化软件工程国际会议论文集》，系列：ASE ’19。IEEE Press，2019年，第 863–873 页。 [在线]。获取地址：https://doi.org/10.1109/ ASE.2019.00085[87] M. Wurzenberger、F. Skopik、M. Landauer、P. Greitbauer、R. Fiedler 和 W. Kastner，“将增量聚类应用于日志数据的半监督异常检测”，载于《第12届可用性、可靠性与安全国际会议论文集》，系列：ARES ’17。美国纽约州纽约：计算机协会，2017年。 [ ] 10 1145 3098954 3098973 在线。获取地址：https://doi.org/。/。D. Gunter、B. L. Tierney、A. Brown、M. Swany、J. Bresnahan 和 J. M. Schopf，“用于分布式系统故障排查的日志摘要与异常检测”，载于《2007年第8届 IEEE/ACM 网格计算国际会议》，2007年，第 226–234 页。[89] W. Meng、F. Zaiter、Y. Huang、Y. Liu、S. Zhang、Y. Zhang、Y. Zhu、T. Zhang、E. Wang、Z. Ren、F. Wang、S. Tao 和 D. Pei，“在线服务中的非结构化日志摘要”，CoRR，第 abs/ 2012 08938，卷。在线。获取地址：https://arxiv.org/abs/。 2020 [ ] 2012 08938
[90] R. Dijkman 和 A. Wilbik，“事件日志的语言摘要——一种实用方法”，《信息系统》，第 67 卷，第 114–125 页，2017年。 [在线]。获取地址：https://www.sciencedirect.com/science/article/pii/S 0306437916303192
[91] S. Locke、H. Li、T.-H. P. Chen、W. Shang 和 W. Liu，“Logassist：通过日志摘要辅助日志分析”，《IEEE 软件工程汇刊》，第 1–1 页，2021年。[92] P. Bodik、M. Goldszmidt、A. Fox、D. B. Woodard 和 H. Andersen，“数据中心指纹识别：性能危机的自动分类”，载于《第5届欧洲计算机系统会议论文集》，系列：EuroSys ’10。美国纽约州纽约：计算机协会，2010年，第 111–124 页。 [在线]。获取地址：https://doi.org/ 10 1145 1755913 1755926 。/。Y. Liang、Y. Zhang、H. Xiong 和 R. Sahoo，“IBM BlueGene/L 事件日志中的故障预测”，载于《第七届 IEEE 数据挖掘国际会议（ICDM 2007）》，2007年，第 583–588 页。[94] M. Chen、A. Zheng、J. Lloyd、M. Jordan 和 E. Brewer，“使用决策树进行故障诊断”，载于《国际自主计算会议》，*2004*年。会议论文集。，2004，第 36–43 页。
[95] S. He、Q. Lin、J.-G. Lou、H. Zhang、M.R. Lyu 和D. Zhang，“通过日志分析识别具有影响的服务系统问题”，载于《2018年欧洲软件工程会议和软件工程基础研讨会第 26届 ACM联合会议论文集》，系列名ESEC/FSE 2018。美国纽约州纽约：计算机协会，2018年，第 60–70 页。 [在线]。可访问：https://doi.org/ 10 1145 3236024 3236083 。 /。J. Lou、Q. Fu、S. Yang、Y. Xu 和J. Li，“从控制台日志中挖掘不变量以进行系统问题检测”，载于《2010年USENIX 年度技术会议》，美国马萨诸塞州波士顿，2010年6月23日至25日，P. Barham和T.Roscoe 编。USENIX 协会，2010年。[在线]。可访问：https://www.usenix.org/conference/usenix-atc-10/ mining-invariants-console-logs-system-problem-detection[97] A. Nandi、A. Mandal、S. Atreja、G.B. Dasgupta和 S. Bhattacharya，“从执行日志中挖掘程序控制流图进行异常检测”，载于《第22 届ACM SIGKDD 知识发现与数据挖掘国际会议论文集》，系列名 KDD ’16。美国纽约州纽约：计算机协会，2016年，第 215–224 页。 [在线]。可访问：https://doi.org/10.1145/2939672.2939712[98] T. Jia、P. Chen、L.Yang、Y. Li、F. Meng 和 J.Xu，“一种基于日志与混合图模型的分布式服务异常诊断方法”，载于《2017年 IEEE Web服务国际会议（ICWS）》，2017年，第 25–32页。[99] T. Jia、L. Yang、P. Chen、Y. Li、F.Meng和 J.Xu，“Logsed：通过挖掘日志中的时间加权控制流图进行异常诊断”，载于《2017年IEEE 第10 届云计算国际会议（CLOUD）》，2017年，第447–455 页。[100] X. Zhang、Y.Xu、S. Qin、S.He、B. Qiao、Z. Li、H. Zhang、X. Li、Y. Dang、Q. Lin、M. Chintalapati、S. Rajmohan 和D.Zhang，“Onion：识别云系统中的事件指示日志”，载于《第 29 届 ACM 欧洲软件工程会议和软件工程基础研讨会联合会议论文集》，系列名ESEC/FSE2021。美国纽约州纽约：计算机协会，2021年，第 1253–1263 页。 [在线]。可访问：https://doi.org/ 10 1145 3468264 3473919 。 / 。W.Meng、Y.Liu、Y. Huang、S.Zhang、F. Zaiter、B.Chen和 D.Pei，“用于在线日志分析的语义感知表示框架”，载于《2020年第 29届计算机通信与网络国际会议（ICCCN）》，2020年，第 1–7 页。[102] A. Farzad和 T. A. Gulliver，“无监督日志消息异常检测”，《ICT Express》，第 6 卷，第 3期，第 229–237 页，2020年。[在线]。可访问：https://www.sciencedirect.com/science/article/pii/ S2405959520300643[103] S. Nedelkoski、J. Bogatinovski、A. Acker、J.Cardoso 和O. Kao，“非结构化日志中的基于自注意力分类的异常检测”，载于《2020年 IEEE 数据挖掘国际会议（ICDM）》，2020年，第 1196–1201页。[104] L. Yang、J.Chen、Z. Wang、W. Wang、J. Jiang、X.Dong 和 W.Zhang，“通过概率标签估计进行基于日志的半监督异常检测”，载于《2021年 IEEE/ACM 第43 届国际软件工程会议（ICSE）》，2021年，第 1448–1460 页。[105] H. Guo、S. Yuan 和 X.Wu，“Logbert：通过 BERT 进行日志异常检测”，载于《2021年国际神经网络联合会议（IJCNN）》，2021年，第1–8 页。[106] B. Xia、Y. Bai、J. Yin、Y. Li 和 J. Xu，“Loggan：用于通过排列事件建模进行异常检测的日志级生成对抗网络”，《Information Systems Frontiers》，第23 卷，第2 期，第285–298 页，2021年4月。 [在线]。可访问：https://doi.org/ 10 1007 10796 020 10026 3 。 /s- - - Z.Zhao、W.Niu、X. Zhang、R. Zhang、Z. Yu 和 C. Huang，“Trine：一个生成对抗网络中使用三个 Transformer 编码器的系统日志异常检测 ”，《 Applied Intelligence 》，第 52卷，第 8期，第8810–8819 页，2022年6月。 [在线]。可访问：https://doi.org/10.1007/ s10489-021-02863-9[108]M. Du、F. Li、G.Zheng 和 V. Srikumar，“Deeplog：通过深度学习从系统日志中进行异常检测与诊断”，载于《2017年 ACM SIGSAC计算机与通信安全会议论文集》，系列名 CCS ’17。美国纽约州纽约：计算机协会，2017年，第 1285–1298 页。 [在线]。可访问：https://doi.org/ 10 1145 3133956 3134015 。 / 。H. Ott、J. Bogatinovski、A. Acker、S. Nedelkoski和O. Kao，“使用预训练语言模型在日志数据中进行鲁棒且可迁移的异常检测”，载于《2021年 IEEE/ACM云智能国际研讨会（CloudIntelligence）》。美国加利福尼亚州洛斯阿拉米托斯：，美国：IEEE计算机学会，2021年5月，第 19 页–

24. [在线]。可访问：https://doi.ieeecomputersociety.org/10.1109/CloudIntelligence 52565 2021 00013 。. R. Chen、S. Zhang、D. Li、Y. Zhang、F. Guo、W. Meng、D. Pei、Y. Zhang、X. Chen 和 Y. Liu，“Logtransfer：利用迁移学习对软件系统进行跨系统日志异常检测 ”，载于《2020年IEEE第31届国际软件可靠性工程研讨会》（ISSRE），2020年，第37–47页。[111] X. Zhang、Y. Xu、Q. Lin、B. Qiao、H. Zhang、Y. Dang、C. Xie、X. Yang、Q. Cheng、Z. Li、J. Chen、X. He、R. Yao、J.-G. Lou、M. Chintalapati、F. Shen 和 D. Zhang，“针对不稳定日志数据的稳健基于日志的异常检测”，载于《2019年第27届ACM欧洲软件工程会议与软件工程基础研讨会联合会议论文集》，系列丛书：ESEC/FSE 2019。美国纽约州纽约市：计算机协会，2019年，第807–817页。 [在线]。可访问：https://doi.org/10 1145 3338906 3338931 。/。. A. Brown、A. Tuor、B. Hutchinson 和 N. Nichols，“用于可解释的系统日志异常检测的循环神经网络注意力机制”，载于《第一届计算系统机器学习研讨会论文集》，系列丛书：MLCS’18。美国纽约州纽约市：计算机协会，2018年。 [在线]。可访问：https://doi.org/ 10 1145 3217871 3217872 。/。. X. Li、P. Chen、L. Jing、Z. He 和 G. Yu，“Swisslog：一种基于深度学习的稳健统一日志异常检测方法，可检测多种故障”，载于《2020年IEEE第31届国际软件可靠性工程研讨会》（ISSRE），2020年，第92–103页。[114] S. Lu、X. Wei、Y. Li 和 L. Wang，“使用卷积神经网络检测大数据系统日志中的异常”，载于《2018年IEEE第16届可信、自主与安全计算国际会议、第16届普适智能与计算国际会议、第4届大数据智能与计算国际会议及网络科学与技术大会（DASC/PiCom/DataCom/CyberSciTech 2018）》，希腊雅典，2018年8月12日至15日。IEEE计算机学会，2018年，第151–158页。 [在线]。可访问：https://doi.org/10.1109/ DASC/PiCom/DataCom/CyberSciTec. 2018 00037 。Y. Xie、H. Zhang 和 M. A. Babar，“Loggd：使用图神经网络检测系统日志中的异常”，2022年。 [在线]。可访问：https://arxiv.org/abs/ 2209 07869 。Y. Wan、Y. Liu、D. Wang 和 Y. Wen，“Glad-paw：通过位置感知加权图注意力网络进行基于图的日志异常检测”，载于《知识发现与数据挖掘进展》，K. Karlapalem、H. Cheng、N. Ramakrishnan、R. K. Agrawal、P. K. Reddy、J. Srivastava 和 T. Chakraborty 编。Cham：Springer国际出版公司，2021年，第66–77页。[117] S. Huang、Y. Liu、C. Fung、R. He、Y. Zhao、H. Yang 和 Z. Luan，“Hi-tanomaly：用于系统日志异常检测的分层 Transformer”，《IEEE网络与服务管理汇刊》，第17卷，第4期，第2064–2076页，2020年。[118] H. Cheng、D. Xu、S. Yuan 和 X. Wu，“通过反事实解释检测序列数据中的细粒度异常”，2022年。 [ ] 2210 04145 在线。可访问：https://arxiv.org/abs/。S. Nedelkoski、J. Cardoso 和 O. Kao，“使用多模态深度学习检测系统跟踪数据中的异常”，载于《2019年IEEE第12届云计算国际会议》（CLOUD），2019年，第179–186页。[120] C. Zhang、X. Peng、C. Sha、K. Zhang、Z. Fu、X. Wu、Q. Lin 和 D. Zhang，“Deeptralog：通过基于图的深度学习进行结合调用链与日志的微服务异常检测”。美国宾夕法尼亚州匹兹堡：IEEE，2022年，第623–634页。[121] D. C. Arnold、D. H. Ahn、B. R. De Supinski、G. L. Lee、B. P. Miller 和 M. Schulz，“用于大规模调试的堆栈跟踪分析”，载于《2007年IEEE国际并行与分布式处理研讨会》。IEEE，2007年，第1–10页。[122] Z. Cai、W. Li、W. Zhu、L. Liu 和 B. Yang，“阿里巴巴数据中心中的实时调用链级根因诊断系统”，《IEEE Access》，第7卷，第142 692–142 702页，2019年。[123] P. Papadimitriou、A. Dasdan 和 H. Garcia-Molina，“用于异常检测的网页图相似性”，《互联网服务与应用期刊》，第1卷，第1期，第19–30页，2010年。[124] M. Y. Chen、E. Kiciman、E. Fratkin、A. Fox 和 E. Brewer，“Pinpoint：大型动态互联网服务中的问题判定”，载于《国际可信系统与网络会议论文集》。IEEE，2002年，第595–604页。[125] S. Hochreiter 和 J. Schmidhuber，“长短期记忆”，《神经计算》，第9卷，第8期，第1735–1780页，1997年。[126] F. Salfner、M. Lenk 和 M. Malek，“在线故障预测方法综述”，《ACM计算调查》，第42卷，第3期，第10:1–10:42，2010年。 [线上]。可用：https://doi.org/10.1145/1670679。1670680 [127] Y. Chen、X. Yang、Q. Lin、H. Zhang、F. Gao、Z. Xu、Y. Dang、D. Zhang、H. Dong、Y. Xu、H.Li和Y. Kang，“云服务系统的服务中断预测与诊断”，载于《万维网大会》，WWW 2019，旧金山，加利福尼亚州，美国，2019年5月13日至17日，L. Liu、R. W.White、A. Mantrach、F. Silvestri、J. J. McAuley、R.Baeza-Yates和L.Zia编。ACM，2019年，第2659–2665页。[ ] 10 1145 3308558 3313501 线上。可用：https://doi.org/ 。/ 。N. Zhao、J. Chen、Z. Wang、X. Peng、G. Wang、Y. Wu、F. Zhou、Z.Feng、X. Nie、W. Zhang、K. Sui 和 D. Pei，“在线服务系统的实时事件预测”，载于 ESEC/FSE ’20：第28届 ACM 欧洲软件工程会议与软件工程基础研讨会联合会议，线上活动，美国，2020年11月8日至13日，P. Devanbu、M. B.Cohen 和 T.Zimmermann编。ACM，2020年，第315–326页。 [线上]。可用：https://doi.org/ 10 1145 3368089 3409672 。/。S. Zhang、Y. Liu、W. Meng、Z. Luo、J. Bu、S. Yang、P. Liang、D. Pei、J. Xu、Y.Zhang、Y. Chen、H. Dong、X.Qu 和 L. Song，“Prefix：数据中心网络中的交换机故障预测”，《ACM测量、分析与计算系统学报》，第2卷，第1期，第2:1–2:29页，2018年。 [线上]。可用：https://doi.org/ 10 1145 3179405 。/ Q.Lin、K.Hsieh、Y. Dang、H.Zhang、K. Sui、Y. Xu、J. Lou、C. Li、Y. Wu、R. Yao、M. Chintalapati 和D. Zhang，“预测云服务系统中的节点故障”，载于《2018年ACM 欧洲软件工程会议与软件工程基础研讨会联合会议论文集》，ESEC/SIGSOFTFSE 2018，美国佛罗里达州湾湖，2018年11月4日至9日，G. T. Leavens、A. Garcia 和 C. S.Pasareanu 编。ACM，2018年，第480–490页。 [线上]。可用： https://doi.org/ 10 1145 3236024 3236060 。/ 。E. Pinheiro、W.Weber 和 L.A.Barroso，“大规模磁盘驱动器群体中的故障趋势”，载于《第5届USENIX 文件与存储技术会议》，FAST2007，2007年2月13日至16日，美国加利福尼亚州圣何塞，A. C. Arpaci-Dusseau 和 R.H. Arpaci-Dusseau编。USENIX，2007年，第17–28页。 [线上]。可用：http://www.usenix.org/events/fast07/tech/pinheiro.html [132] Y. Xu、K. Sui、R. Yao、H. Zhang、Q. Lin、Y. Dang、P. Li、K. Jiang、W. Zhang、J. Lou、M. Chintalapati 和D. Zhang，“通过预测磁盘错误提高云系统的服务可用性”，载于《2018年 USENIX年度技术会议》，USENIX ATC2018，美国马萨诸塞州波士顿，2018年7月11日至13日，H.S. Gunawi 和 B.Reed 编。USENIX协会，2018年，第481–494页。 [线上]。可用：https://www.usenix.org/conference/atc18/presentation/xu-yong [133] R.K. Sahoo、A.J. Oliner、I. Rish、M. Gupta、J.E. Moreira、S. Ma、R. Vilalta 和 A. Sivasubramaniam，“大规模计算机集群中的主动式管理关键事件预测”，载于《第九届ACM SIGKDD 知识发现与数据挖掘国际会议论文集》，系列：KDD’03。美国纽约，计算机协会，2003年，第426–435页。[ ] 10 1145 956750 956799 线上。可用：https://doi.org/ 。/。F. Yu、H.Xu、S.Jian、C. Huang、Y. Wang和 Z.Wu，“大规模数据中心中的 DRAM故障预测”，载于《2021年IEEE 云计算联合国际会议》（JCC），2021年，第1–8页。[135] J. Klinkenberg、C. Terboven、S. Lankes 和 M.S. M¨uller，“基于数据挖掘的高性能计算中心运维分析”，载于《2017年 IEEE 集群计算国际会议》（CLUSTER），2017年，第766–773页。[136] S. Zhang、Y. Liu、W. Meng、Z. Luo、J. Bu、S. Yang、P. Liang、D. Pei、J. Xu、Y.Zhang、Y. Chen、H. Dong、X.Qu 和 L. Song，“Prefix：数据中心网络中的交换机故障预测”，《ACM测量、分析与计算系统学报》，第2卷，第1期，2018年4月。 [线上]。可用：https://doi.org/ 10 1145 3179405 。/ B. Russo、G. Succi 和 W. Pedrycz，“挖掘系统日志以学习错误预测器：遥测系统案例研究”，《实证软件工程》，第20卷，第4期，第879–927页，2015年。 [线上]。可用： https://doi.org/10 1007 10664 014 9303 2 。/s - - -I. Fronza、A. Sillitti、G. Succi、M.Terho 和 J.Vlasenko，“使用随机索引和支持向量机基于日志文件进行故障预测”，《系统与软件学报》，第86卷，第1期，第2–11页，2013年1月。[ ] 10 1016 2012 06 025 线上。可用：https://doi.org/ 。/j.jss.。F. Salfner 和 M.Malek，“使用隐半马尔可夫模型进行有效的在线故障预测”，载于《2007年第26届 IEEE 可靠分布式系统国际研讨会》（SRDS 2007），2007年，第161–174页。[140] A. Das、F. Mueller、C. Siegel和 A.Vishnu，“Desh：用于预测高性能计算系统健康状况及故障前提前时间的深度学习”，载于《论文集

第27届高性能并行与分布式计算国际研讨会论文集，系列丛书 HPDC '18。美国纽约州纽约：计算机协会，2018年，第40–51页。 [在线]。网址：https://doi.org/ 10 1145 3208040 3208051 。/ 。J. [141] Gao、H. Wang 和 H. Shen，《利用深度学习预测云数据中心中的任务故障》，载于2019年IEEE大数据国际会议（大数据），2019年，第1111–1116页。[142] Z. Zheng、Z. Lan、B. H. Park 和 A. Geist，《系统日志预处理以改进故障预测》，载于2009年IEEE/IFIP可信赖系统与网络国际会议，2009年，第572–577页。[143] Y. Chen、X. Yang、Q. Lin、H. Zhang、F. Gao、Z. Xu、Y. Dang、D. Zhang、H. Dong、Y. Xu、H. Li 和 Y. Kang，《云服务系统的服务中断预测与诊断》，载于万维网大会，系列丛书 WWW '19。美国纽约州纽约：计算机协会，2019年，第2659–2665页。 [在线]。网址：https://doi.org/ 10 1145 3308558 3313501 。/ 。Q. Lin、K. Hsieh、Y. [144] Dang、H. Zhang、K. Sui、Y. Xu、J.-G. Lou、C. Li、Y. Wu、R. Yao、M. Chintalapati 和 D. Zhang，《预测云服务系统中的节点故障》，载于2018年第26届欧洲软件工程会议与软件工程基础研讨会ACM联合会议论文集，系列丛书 ESEC/FSE 2018。美国纽约州纽约：计算机协会，2018年，第480–490页。 [在线]。网址：https://doi.org/ 10 1145 3236024 3236060 。/ 。X. Zhou、X. [145] Peng、T. Xie、J. Sun、C. Ji、D. Liu、Q. Xiang 和 C. He，《通过系统跟踪日志学习，预测微服务应用中的潜在错误并进行故障定位》，载于2019年第27届欧洲软件工程会议与软件工程基础研讨会ACM联合会议论文集，系列丛书 ESEC/FSE 2019。美国纽约州纽约：计算机协会，2019年，第683–694页。 [在线]。网址：https://doi.org/ 10 1145 3338906 3338961 。/ 。H. Nguyen、Y. [146] Tan 和 X. Gu，《Pal：面向云托管分布式应用的传播感知异常定位》，系列丛书 SLAML '11。美国纽约州纽约：计算机协会， 2011 [ ] 10 1145 2038633 2038634 。在线。[147] 网址：https://doi.org/ 。/ 。H. Nguyen、Z. Shen、Y. Tan 和 X. Gu，《Fchain：迈向云系统的黑盒在线故障定位》，载于2013年第33届IEEE分布式计算系统国际会议，2013年，第21–30页。[148] H. Shan、Y. Chen、H. Liu、Y. Zhang、X. Xiao、X. He、M. Li 和 W. Ding，《e-diagnosis：大规模微服务平台中小窗口长尾延迟的无监督实时诊断》，载于万维网大会，系列丛书 WWW '19。美国纽约州纽约：计算机协会，2019年，第3215–3222页。 [ ] 10 1145 3308558 3313653 在线。网址：[149] https://doi.org/ 。/ 。J. Thalheim、A. Rodrigues、I. E. Akkus、P. Bhatotia、R. Chen、B. Viswanath、L. Jiao 和 C. Fetzer，《Sieve：从分布式系统的监控指标中获取可操作的洞见》，载于第18届ACM/IFIP/USENIX中间件大会论文集，系列丛书 Middleware '17。美国纽约州纽约：计算机协会，2017年，第 14 27 [ ] 10 1145 3135974 3135977 –。在线。[150] 网址：https://doi.org/ 。/ 。L. Wu、J. Tordsson、E. Elmroth 和 O. Kao，《Microrca：微服务性能问题的根因定位》，载于 NOMS 2020——2020年IEEE/IFIP网络运维与管理研讨会，2020年，第1–9页。[151] ´Alvaro Brand´on、M. Sol´e、A. Hu´elamo、D. Solans、M. S. P´erez 和 V. Munt´es-Mulero，《面向服务和微服务架构的基于图的根因分析》，《系统与软件杂志》，第159卷，文章编号110432，2020年。 [在线]。网址：https://www.sciencedirect.com/science/article/pii/S 0164121219302067 [152] A. Samir 和 C. Pahl，《Dla：使用马尔可夫模型检测和定位容器化微服务架构中的异常》，载于2019年第7届未来物联网与云国际会议（FiCloud），2019年，第205–213页。[153] P. Wang、J. Xu、M. Ma、W. Lin、D. Pan、Y. Wang 和 P. Chen，《Cloudranger：云原生系统的根因识别》，载于2018年第18届IEEE/ACM集群、云与网格计算国际研讨会（CCGRID），2018年，第492–502页。[154] L. Mariani、C. Monni、M. Pezz´e、O. Riganelli 和 R. Xin，《云系统中的故障定位》，载于2018年第11届IEEE软件测试、验证与确认国际会议（ICST），2018年，第262–273页。[155] P. Chen、Y. Qi 和 D. Hou，《Causeinfer：利用云环境中的分层因果图实现自动化端到端性能诊断》，《IEEE服务计算汇刊》，第12卷，第2期，第214–230页 [156] Y. Meng、S. Zhang、Y. Sun、R. Zhang、Z. Hu、Y.Zhang、C. Jia、Z.Wang、和 D.Pei，“通过因果推理定位微服务中的故障根因”，载于 2020年 IEEE/ACM第 28 届服务质量国际研讨会（IWQoS），2020年，第 1–10 页。[157] J. Lin、P.Chen 和 Z. Zheng，“Microscope：利用因果图在微服务环境中精准定位性能问题”，载于 ICSOC，2018年。[158] M. Ma、W.Lin、D.Pan和P.Wang，“Ms-rank：面向微服务应用的多指标自适应根因诊断”，载于2019年 IEEE Web服务国际会议（ICWS），2019年，第 60–67 页。[159] M. Ma、J.Xu、Y. Wang、P. Chen、Z.Zhang 和 P. Wang，《AutoMAP：自动诊断基于微服务的Web应用》。纽约，纽约州，美国：计算机协会，2020年，第246–258 页。 [在线]。可访问：https://doi.org/10.1145/3366423。 3380111[160] M.Kim、R. Sumbaly 和 S. Shah，“面向服务架构中的根因检测”，载于《ACMSIGMETRICS/计算机系统测量与建模国际会议论文集》，丛书：SIGMETRICS '13。纽约，纽约州，美国：计算机协会，2013年，第 93–104 页。 [在线]。可访问：https://doi.org/ 10 1145 2465529 2465753 。 /。[161] P.Spirtes 和 C. Glymour，“一种快速恢复稀疏因果图的算法”，《社会科学计算评论》，第 9卷，第1 期，第 62–72 页，1991 年。[162] D. M. Chickering，“学习贝叶斯网络结构的等价类”，《机器学习研究期刊》，第2 卷，第 3 期，第 445–498页，2002 年。[163] ——，“ 通过贪心搜索进行最优结构识别 ”，《机器学习研究期刊》，第 3 卷，第 3 期，第 507–554 页，2003 年。[164] J.Runge 、 P. Nowack 、 M. Kretschmer 、 S. Flaxman 和D.Sejdinovic，“在大型非线性时间序列数据集中检测并量化因果关联”，《科学进展》，第 5， 11， 4996， 2019 [ ] 10 1126 卷，第 期，第 页，eaau。在线。可访问：https://www.science.org/doi/abs/ 。/ sciadv.aau4996[165] J. Runge，“在自相关非线性时间序列数据集中发现同时发生的和滞后的因果关系”，载于UAI，2020年。[166] A.Gerhardus和J. Runge，“利用潜在混杂因素对自相关时间序列进行高召回率因果发现”，载于《神经信息处理系统进展》，H. Larochelle、M. Ranzato、R. Hadsell、M. Balcan 和 H. Lin编，第 33 卷。CurranAssociates,Inc.，2020年，第 12 615–12 625 页。[在线]。可访问：https://proceedings.neurips.cc/paper/2020/file/94e70705efae423efda1088614128d0b-Paper.pdf [167] J. Qiu、Q. Du、K.Yin、S.-L.Zhang 和C. Qian，“一种基于因果挖掘和知识图谱的云应用性能异常根因诊断方法”，《应用科学》，第 10 卷，第 6 期，2020年。 [在线]。可访问：https://www.mdpi.com/ 2076-3417/10/6/2166[168] K. Budhathoki、L. Minorics、P.Bloebaum和 D. Janzing，“基于因果结构的离群值根因分析”，载于 ICML 2022 [ ] 2022， 。在线。可访问：https://www.amazon.science/publications/causal-structure-based-root-cause-analysis-of-outliers[169] S. Lu、B. Rao、X.Wei、B. Tak、L. Wang 和L. Wang，“面向Spark 的基于日志的异常任务检测与根因分析”，载于2017年 IEEE Web服务国际会议（ICWS），2017年，第 389–396 页。[170] F. Lin、K. Muzumdar、N. P.Laptev、M.-V. Curelea、S. Lee 和 S. Sankar，“大规模服务环境中的快速维度分析与根因调查”，《ACM测量、分析与计算系统论文集》，第 4卷，第 2 期，2020年 6 月。 [在线]。可访问：https://doi.org/ 10 1145 3392149 。/L. Wang、N.Zhao、J. Chen、P. Li、[171] W.Zhang 和K. Sui，“通过日志异常检测定位微服务系统的根因指标”，载于2020年 IEEE Web服务国际会议（ICWS），2020年，第142–150页。[172] C. Luo、J.-G. Lou、Q. Lin、Q.Fu、R.Ding、D.Zhang和 Z. Wang，“将事件与时间序列关联用于事件诊断”，载于第20届ACM SIGKDD 知识发现与数据挖掘国际会议论文集，丛书：KDD'14。纽约，纽约州，美国：计算机协会，2014年，第 1583–1592 页。 [ ] 10 1145 2623330 2623374 在线。可访问：[173] https://doi.org/。 /。E.Chuah、S.-h.Kuo、P. Hiew、W.-C. Tjhi、G. Lee、J. Hammond、M. T. Michalewicz、T.Hung 和J. C.Browne，“从集群日志文件中诊断故障的根因”，载于 2010年高性能计算国际会议，2010年，第 1–10 页。[174] T. S. Zaman、X. Han 和T.Yu，“Scminer：从大型系统调用追踪数据中定位系统级并发故障

*IEEE/ACM*自动化软件工程国际会议（*ASE*），2019年，第515–526页。[175] K. Zhang、J. Xu、M. R. Min、G. Jiang、K. Pelechrinis和H. Zhang，“自动化IT系统故障预测：一种深度学习方法”，载于*2016*年*IEEE*国际大数据会议（大数据），2016年，第1291–1300页。[176] Y. Yuan、W. Shi、B. Liang和B. Qin，“一种基于OpenStack异常日志的云执行故障诊断方法”，载于*2019*年*IEEE*第*12*届云计算国际会议（*CLOUD*），2019年，第124–131页。[177] H. Mi、H. Wang、Y. Zhou、M. R.-T. Lyu和H. Cai，“面向生产环境云计算系统的细粒度、无监督、可扩展性能诊断”，*IEEE*并行与分布式系统汇刊，第24卷，第6期，第1245–1255页，2013年。[178] H. Jiang、X. Li、Z. Yang和J. Xuan，“是什么导致了我的测试告警？系统与集成测试中测试告警的自动原因分析”，载于第*39*届国际软件工程会议论文集，系列：ICSE '17。IEEE出版社，2017年，第712–723页。

[ ] 10 1109 2017 71 在线。可访问：[179] https://doi.org/ 。/ICSE。 。A. Amar和P. C. Rigby，“挖掘历史测试日志以预测缺陷并定位测试日志中的故障”，载于*2019*年第*41*届*IEEE/ACM*国际软件工程会议（*ICSE*），2019年，第140–151页。[180] F. Wang、A. Bundy、X. Li、R. Zhu、K. Nuamah、L. Xu、S. Mauceri和J. Z. Pan，“Lekg：一种从日志抽取构建知识图谱的系统”，载于第*10*届国际知识图谱联合会议，系列：IJCKG'21。美国纽约州纽约：计算机协会，2021年，第181–185页。 [在线]。可访问：https://doi.org/

10 1145 3502223 3502250 。/ 。A. Ekelhart、[181] F. J. Ekaputra和E. Kiesling，“用于自动化日志知识图谱构建的Slogert框架”，载于语义网，R. Verborgh、K. Hose、H. Paulheim、P.-A. Champin、M. Maleshkova、O. Corcho、P. Ristoski和M. Alam编。Cham：Springer国际出版，2021年，第631–646页。[182] C. Bansal、S. Renganathan、A. Asudani、O. Midy和M. Janakiraman，“Decaf：大规模云服务中的性能问题诊断与分诊”，载于第*42*届*ACM/IEEE*国际软件工程会议论文集：软件工程实践，系列：ICSE-SEIP '20。美国纽约州纽约：计算机协会，2020年，第201–210页。 [在线]。可访问：https://doi.org/

10 1145 3377813 3381353 。/ 。B. C. Tak、S. Tao、L. [183] Yang、C. Zhu和Y. Ruan，“Logan：利用基于日志的参考模型进行云端问题诊断”，载于*2016*年*IEEE*国际云工程会议（*IC2E*），2016年，第62–67页。[184] W. Shang、Z. M. Jiang、H. Hemmati、B. Adams、A. E. Hassan和P. Martin，“在Hadoop云上部署大数据分析应用时为开发人员提供辅助”，载于*2013*年第*35*届国际软件工程会议（*ICSE*），2013年，第402–411页。[185] C. Pham、L. Wang、B. C. Tak、S. Baset、C. Tang、Z. Kalbarczyk和R. K. Iyer，“使用定向故障注入诊断分布式系统故障”，*IEEE*并行与分布式系统汇刊，第28卷，第2期，第503–516页，2017年。[186] K. Nagaraj、C. Killian和J. Neville，“通过对系统日志进行结构化比较分析来诊断性能问题”，载于第*9*届*USENIX*网络系统设计与实现会议论文集，系列：NSDI'12。美国：USENIX 协会，2012年，第26页。[187] H. Ikeuchi、A. Watanabe、T. Kawata和R. Kawahara，“使用用户操作生成的日志进行根因诊断”，载于*2018*年*IEEE*全球通信会议（*GLOBECOM*），2018年，第1–7页。[188] P. Aggarwal、A. Gupta、P. Mohapatra、S. Nagar、A. Mandal、Q. Wang和A. Paradkar，“通过使用黄金信号挖掘日志中的因果依赖关系，定位云应用程序中的运行故障”，载于面向服务的计算——*ICSOC 2020*研讨会，H. Hacid、F. Outay、H.-y. Paik、A. Alloum、M. Petrocchi、M. R. Bouadjenek、A. Beheshti、X. Liu和A. Maaradji编。Cham：Springer国际出版，2021年，第137–149页。[189] Y. Zhang、Z. Guan、H. Qian、L. Xu、H. Liu、Q. Wen、L. Sun、J. Jiang、L. Fan和M. Ke，“CloudRCA：云计算平台的根因分析框架”，载于第*30*届*ACM*国际信息与知识管理会议论文集，系列：CIKM '21。美国纽约州纽约：计算机协会，2021年，第4373–4382页。 [在线]。可访问：https://doi.org/

10 1145 3459637 3481903 。/ 。H. Wang、Z. Wu、H. [190] Jiang、Y. Huang、J. Wang、S. Kopru和T. Xie，“Groot：一种基于事件图的根因分析方法】【。

工业环境中的分析”，载于《第36届IEEE/ACM自动化软件工程国际会议论文集》，系列：ASE '21。IEEE出版社，2021年，第419–429页。 [在线]。获取地址：https://doi.org/

10 1109 51524 2021 9678708 。/ASE . . X. Fu、R. [191] Ren、S. A. McKee、J. Zhan和N. Sun，“深入挖掘集群系统日志以进行故障预测和根因诊断”，载于《2014年IEEE国际集群计算会议（CLUSTER）》，2014年，第103–112页。[192] S. Kobayashi、K. Fukuda和H. Esaki，“利用因果推断从日志数据中挖掘网络事件的成因”，载于《2017年IFIP/IEEE综合网络与服务管理研讨会（IM）》，2017年，第45–53页。[193] S. Kobayashi、K. Otomo和K. Fukuda，“结合分层协议和拓扑知识对网络日志进行因果分析”，载于《2019年第15届网络与服务管理国际会议（CNSM）》，2019年，第1–9页。[194] R. Jarry、S. Kobayashi和K. Fukuda，“网络日志数据的定量因果分析”，载于《2021年IEEE第45届年度计算机、软件与应用会议（COMPSAC）》，2021年，第1437–1442页。[195] D. Liu、C. He、X. Peng、F. Lin、C. Zhang、S. Gong、Z. Li、J. Ou和Z. Wu，“Microhecl：大规模微服务系统中的高效根因定位”，载于《第43届国际软件工程会议：软件工程实践论文集》，系列：ICSE-SEIP '21。IEEE出版社，2021年，第338–347页。 [在线]。获取地址：https://doi.org/

10 1109 52600 2021 00043 。/ [196] ICSE-SEIP . . Z. Li、J. Chen、R. Jiao、N. Zhao、Z. Wang、S. Zhang、Y. Wu、L. Jiang、L. Yan、Z. Wang等，“通过追踪分析对微服务系统进行实用的根因定位”，载于《2021年IEEE/ACM第29届服务质量国际研讨会（IWQOS）》。IEEE，2021年，第1–10页。[197] X. Zhou、X. Peng、T. Xie、J. Sun、C. Ji、D. Liu、Q. Xiang和C. He，“通过系统跟踪日志学习，对微服务应用进行潜在错误预测和故障定位”，载于《2019年第27届ACM欧洲软件工程会议与软件工程基础研讨会联合会议论文集》，2019年，第683–694页。[198] P. Liu、H. Xu、Q. Ouyang、R. Jiao、Z. Chen、S. Zhang、J. Yang、L. Mo、J. Zeng、W. Xue等，“通过服务级深度贝叶斯网络无监督检测微服务追踪异常”，载于《2020年IEEE第31届国际软件可靠性工程研讨会（ISSRE）》。IEEE，2020年，第48–58页。[199] B. H. Sigelman、L. A. Barroso、M. Burrows、P. Stephenson、M. Plakal、D. Beaver、S. Jaspan和C. Shanbhag，“Dapper：大规模分布式系统追踪基础设施”，2010年。[200] J. Jiang、W. Lu、J. Chen、Q. Lin、P. Zhao、Y. Kang、H. Zhang、Y. Xiong、F. Gao、Z. Xu、Y. Dang和D. Zhang，“如何缓解事件？面向在线服务系统的有效故障排查指南推荐技术”，载于《第28届ACM欧洲软件工程会议与软件工程基础研讨会联合会议论文集》，系列：ESEC/FSE 2020。美国纽约州纽约：计算机协会，2020年，第1410–1420页。[在线]。获取地址：https://doi.org/

10 1145 3368089 3417054 。/ . M. Shetty、C. Bansal、[201] S. P. Upadhyayula、A. Radhakrishna和A. Gupta，“Autotsg：事件故障排查的学习与合成”，CoRR，第abs/2205.13457卷，2022年。 [在线]。获取地址：https://doi.org/

10 48550 2205 13457 。/arXiv. . X. Nie、[202] Y. Zhao、K. Sui、D. Pei、Y. Chen和X. Qu，“为自动化基于Web的服务诊断挖掘因果关系图”，载于《2016年IEEE第35届国际性能计算与通信会议（IPCCC）》，2016年，第1–8页。[203] W. Lin、M. Ma、D. Pan和P. Wang，“Facgraph：用于微服务架构根因诊断的频繁异常相关图挖掘”，载于《2018年IEEE第37届国际性能计算与通信会议（IPCCC）》，2018年，第1–8页。[204] R. Ding、Q. Fu、J. G. Lou、Q. Lin、D. Zhang和T. Xie，“挖掘历史问题仓库以修复大规模在线服务系统”，载于《2014年第44届IEEE/IFIP国际可信系统与网络会议》，2014年，第311–322页。[205] M. Shetty、C. Bansal、S. Kumar、N. Rao、N. Nagappan和T. Zimmermann，“从云服务事件中提取神经知识”，载于《第43届国际软件工程会议：软件工程实践论文集》，系列：ICSE-SEIP '21。IEEE出版社，2021年，第218–227页。 [在线]。获取地址：https://doi.org/

10 1109 52600 2021 00031 。/ICSE-SEIP . . [206] M. Shetty、C. Bansal、S. Kumar、N. Rao和N. Nagappan，“Softner：从云事件中挖掘知识图谱

，”实证。

《软件工程》，第27卷，第4期，第93页，2022年。 [在线]。可访问：https://doi.org/10.1007/s10664-022-10159-w [207]A.Saha和S. C. H.Hoi，“从云服务事件调查中挖掘面向智能运维的根因知识”，载于第44届IEEE/ACM软件工程国际会议：软件工程实践，ICSE (SEIP)2022，美国宾夕法尼亚州匹兹堡，2022年5月22日至24日。IEEE，2022年，第197–206页。 [在线]。可访问：https://doi.org/10 1109 55303 2022 9793994 。/ [208]ICSE-SEIP 。。S. Becker、F.Schmidt、A. Gulenko、A. Acker和 O. Kao，“迈向边缘计算环境中的智能运维”，载于2020年 IEEE大数据国际会议（大数据），2020年，第3470–3475页。[209] S. Levy、R. Yao、Y. Wu、Y. Dang、P.Huang、Z. Mu、P.Zhao、T. Ramani、N. Govindaraju、X. Li、Q. Lin、G. L. Shafriri和 M.Chintalapati，“用于避免生产云虚拟机中断的预测性和自适应故障缓解”，载于第14届 USENIX操作系统设计与实施研讨会（OSDI 20）。USENIX协会，2020年11月，第1155–1170页。 [在线]。可访问：https://www.usenix.org/conference/osdi20/presentation/levy[210] J. D.Hamilton，《时间序列分析》。普林斯顿大学出版社，1994年。[211] R.Hyndman和G. Athanasopoulos，《预测：原理与实践》，第2版。OTexts，2018年。[212] S.Hochreiter 和J. Schmidhuber，“长短期记忆”，《神经计算》，第9卷，第8期，第1735–1780页，1997年。[213] R. N. Calheiros、E. Masoumi、R. Ranjan和R. Buyya，“使用 ARIMA 模型进行工作负载预测及其对云应用服务质量的影响”，IEEE云计算汇刊，第3卷，第4期，第449–458页，2015年。[214] D. Buchaca、J. L.Berral、C. Wang 和 A. Youssef，“面向云原生机器学习服务的主动式容器自动扩缩容”，载于2020年 IEEE 第13届云计算国际会议（CLOUD），2020年，第475–479页。[215] M. Wajahat、A. Gandhi、A.Karve和 A. Kochut，“使用机器学习进行黑盒自动扩缩容”，载于2016年第七届绿色与可持续计算国际会议（IGSC），2016年，第1–8页。[216] N.-M. Dang-Quang 和 M. Yoo，“使用双向长短期记忆进行基于深度学习的Kubernetes 自动扩缩容”，《应用科学》，第11卷，第9期，2021年。 [在线]。可访问：https://www.mdpi.com/2076-3417/11/9/3835[217] Y.Gar´ı、D. A. Monge、E.Pacini、C. Mateos 和C. Garc´ıaGarino，“云中基于强化学习的应用程序自动扩缩容：综述”，《人工智能工程应用》，第102卷，第104288页，2021年。 [在线]。可访问：https://www.sciencedirect.com/ science/article/pii/S0952197621001354 [218] S. Mustafa、B. Nazir、A. Hayat、A.ur Rehman Khan和 S.A.Madani，“云计算中的资源管理：分类、前景与挑战”，《计算机与电气工程》，第47卷，第186–203页，2015年。 [在线]。可访问：https://www.sciencedirect.com/science/article/pii/S004579061500275X [219] T.Khan、W. Tian、G. Zhou、S.Ilager、M. Gong 和R.Buyya，“云计算中以机器学习（ML）为中心的资源管理：综述与未来方向”，《网络与计算机应用期刊》，第204卷，第103405页，2022年。 [在线]。可访问：https://www.sciencedirect.com/science/article/pii/S1084804522000649 [220] F. Nzanywayingoma 和 Y. Yang，“云计算环境中的高效资源管理技术：综述与讨论”，《国际计算机与应用期刊》，第41卷，第3期，第165–182页，2019年。 [在线]。可访问：https://doi.org/10.1080/1206212X.2017.1416558[221] N. M. Gonzalez、T. C. M. D. B.Carvalho和 C. C. Miers，“云资源管理：面向在复杂基础设施上高效执行大规模科学应用程序和工作流”，《云计算杂志》，第6卷，第1期，2017年12月。 [在线]。可访问：https://doi.org/10 1186 13677 017 0081 4 。/ [222]s- - - R.Bianchini、M. Fontoura、E. Cortez、A.Bonde、A. Muzio、A.-M. Constantin、T. Moscibroda、G. Magalhaes、G. Bablani 和 M. Russinovich，“迈向以 ML 为中心的云平台”，《ACM 通讯》，第63卷，第2期，第50–59页，2020年1月。 [在线]。可访问：https://doi.org/10 1145 3364684 。/ E. Cortez、A.Bonde、A. Muzio、[223]M. Russinovich、M. Fontoura 和 R.Bianchini，“资源中心：理解和预测工作负载，以改善大型云平台中的资源管理”，载于第26届操作系统原理研讨会论文集，系列：SOSP ’17。美国纽约州纽约：计算机协会，2017年，第153–167页。 [在线]。可访问：https://doi.org/10.1145/3132747.3132772 [224] K.Haghshenas和 S. Mohammadi，“基于预测的未充分利用主机和目标主机选择方法，用于数据中心中节能的动态虚拟机整合”，超级计算杂志，第 76卷，第 12期，第 10240–10 257 页，2020年 12 月。 [在线]。可访问：https://doi.org/10 1007 11227 020 03248 4 。 /s- - - S. [225]Ilager、K. Ramamohanarao 和 R. Buyya，“使用机器学习预测云的热量以实现高效能源管理”，*IEEE*并行与分布式系统汇刊，第32卷，第 5 期，第1044–1056 页，2021年。[226] H. Mao、M. Schwarzkopf、S. B. Venkatakrishnan、Z. Meng和 M. Alizadeh，“为数据处理集群学习调度算法”，载于*ACM*数据通信特别兴趣小组会议论文集，SIGCOMM ’19 系列。美国纽约州纽约市：计算机协会，2019年，第 270–288 页。 [在线]。可访问：https://doi.org/10 1145 3341302 3342080 。 / . [227]D. Anderson，“什么是应用性能监控？ ” . 在线 . 可访问：https: //www.dynatrace.com/news/blog/what-is-apm-2/ [228] J. Livens，“什么是可观测性？不仅仅是日志、指标和追踪数据”，2021年。 [在线]。可访问：https://www.dynatrace.com/news/blog/what-is-observability-2/ [229] Y.Guo、Y. Wen、C. Jiang、Y. Lian和 Y. Wan，“使用多头注意力检测日志异常（LAMA）”，*CoRR*，第abs/2101.02392 卷，2021年。 [在线]。可访问：https://arxiv.org/abs/2101. 02392[230] S.Zhang、Y. Liu、X. Zhang、W.Cheng、H. Chen 和 H. Xiong，“Cat：超越高效 Transformer 的面向内容的事件序列异常检测”，载于第 *28* 届*ACM SIGKDD*知识发现与数据挖掘大会论文集，KDD ’22 系列。美国纽约州纽约市：计算机协会，2022年，第4541–4550 页。 [ ] 10 1145 3534678 3539155 在线。可访问：https://doi.org/。/ . [231]C. Zhang、X. Wang、H.Zhang、H. Zhang 和 P. Han，“基于局部信息提取和全局稀疏 Transformer 模型的日志序列异常检测”，*IEEE*网络与服务管理汇刊，第18 卷，第 4 期，第4119–4133页，2021年。[232] Q. Wang、X. Zhang、X. Wang 和 Z. Cao，“基于对比对抗训练和双重特征提取的日志序列异常检测方法”，*Entropy*，第 24卷，第 1期，第 69页，2021年 12月。[233] J. Qi、Z. Luan、S. Huang、Y. Wang、C. J. Fung、H. Yang 和 D.Qian，“Adanomaly：使用对抗学习对系统日志进行自适应异常检测”，载于*2022IEEE/IFIP*网络运维与管理研讨会，*NOMS 2022*，匈牙利布达佩斯，*2022*年 *4*月*25–29* 日。IEEE，2022年，第 1–5 页。 [在线]。可访问：https://doi.org/10 1109 54207 2022 9789917 。 /NOMS . . M. [234]Attariyan、M. Chow和 J.Flinn，“X-ray：自动化生产软件性能异常的 {根因} 诊断”，载于第 *10* 届 *USENIX* 操作系统设计与实现研讨会（*OSDI12*），2012年，第307–320页。[235] X. Guo、X. Peng、H. Wang、W. Li、H.Jiang、D. Ding、T.Xie 和 L.Su，“面向微服务架构理解和问题诊断的基于图的追踪数据分析”，载于第 *28* 届 *ACM* 欧洲软件工程会议与软件工程基础研讨会联合会议论文集，2020年，第1387–1397 页。[236] Y. Xu、Y. Zhu、B. Qiao、H. Che、P.Zhao、X. Zhang、Z.Li、Y.Dang 和 Q. Lin，“Tracelingo：云服务中的追踪数据表示与学习，用于性能问题诊断”，载于 *2021 IEEE/ACM* 云智能国际研讨会（*CloudIntelligence*）。IEEE，2021年，第37–40 页。[237] M.Li、Z. Li、K. Yin、X.Nie、W.Zhang、K. Sui 和 D. Pei，“基于因果推断和干预识别的在线服务系统根因分析”，载于第 *28* 届 *ACM SIGKDD*知识发现与数据挖掘大会论文集，KDD ’22 系列。美国纽约州纽约市：计算机协会，2022年，第 3230–3240 页。 [在线]。可访问：https://doi.org/10.1145/3534678.3539041