计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
62(10): 2481−2494, 2025

一种云平台中的异常检测轻量化模型智能构建方案

周颖杰　杨　敏　吴　迪　刘凡兴　赵　伟　邓怡然　吕建成
（四川大学计算机学院　成都　610065）
（yjzhou@scu.edu.cn）

An Intelligent Lightweight Model Construction Solution for Anomaly Detection in Cloud Platforms

Zhou Yingjie, Yang Min, Wu Di, Liu Fanxing, Zhao Wei, Deng Yiran, and Lü Jiancheng
（*College of Computer Science*, *Sichuan University*, *Chengdu* 610065）

Abstract　In order to ensure the safety and efficient operation of cloud services, it is necessary to conduct comprehensive monitoring and analysis of various accessing behaviors and operations in cloud platforms, aiming at detecting potential abnormal behaviors in a timely and accurate manner. There are numerous anomaly detection tasks in cloud platforms. Existing approaches usually require task-specific customized design and fine-tuning, lacking generalizable model construction capabilities for various anomaly detection tasks. Moreover, this process heavily relies on machine learning expertise, making it challenging for domain experts to develop effective models suitable for practical applications. To address the above problems, an intelligent lightweight model construction solution is proposed for anomaly detection in cloud platforms. Compared with existing technology, the proposed solution 1) supports diverse types of anomaly detection tasks in cloud platforms, allowing domain experts to achieve a timely and self-service construction of task-oriented anomaly detection models by simply providing basic task-specific configuration information; 2) enables domain experts to intelligently construct lightweight models for target anomaly detection tasks through automated feature construction, feature optimization, and hyperparameter tuning, even with minimal machine learning and deep learning expertise. Both the case studies and the experimental results based on data collected from large-scale real-world cloud scenarios demonstrate the proposed solution could build models for relevant anomaly detection tasks in a timely and self-service manner, while strong detection capabilities are also achieved in the constructed models.

Key words　cloud platform; abnormal behavior; anomaly detection; lightweight model; intelligent construction

摘　要　通过全面监控和分析云平台中的各类访问与操作，及时准确地发现潜在的异常行为，对保障云服务的安全高效运行具有重要意义. 云平台中存在大量异常检测任务，现有方法通常需要针对性地对各个任务进行专门设计和调优，不具备对相关异常检测任务通用的模型构建能力. 同时，这一过程高度依赖机器学习相关专业知识，使得领域专家难以构建出适用于实际任务的有效模型. 针对上述问题，提出了一种云平台中的异常检测轻量化模型智能构建方案. 该方案相较于现有技术：1）支持不同类型的云平台异常检测任务，领域专家仅需提供任务相关基本配置信息，即可基于该方案快速自助地搭建面向目标任务的

收稿日期：2025-05-29；**修回日期**：2025-07-15
基金项目：国家自然科学基金项目（62171302）；四川省自然科学基金杰出青年科学基金项目（2023NSFSC1965）；高等学校学科创新引智计划（B21044）
This work was supported by the National Natural Science Foundation of China (62171302), the Sichuan Science and Technology Program (2023NSFSC1965), and the 111 Project (B21044).
通信作者：吕建成（lvjiancheng@scu.edu.cn）

异常检测模型;2)能使专家在具备尽可能少的机器学习和深度学习知识情况下,通过特征自动构造、特征自动优化和模型超参数自动优化等实现目标异常任务的轻量化模型智能构建.基于大规模真实云场景所收集数据的案例分析和实验结果表明,所提出方案能针对相关异常检测任务快速自助地搭建模型并具备良好的检测能力.

关键词 云平台;异常行为;异常检测;轻量化模型;智能构建

中图法分类号 TP309

DOI: 10.7544/issn1000-1239.202550349 **CSTR:** 32373.14.issn1000-1239.202550349

随着云服务业务需求的不断增加,云服务的种类逐渐丰富,其涉及的内在交互关系也日趋复杂.由于云平台中的资源往往包含大量有价值的数据(例如用户操作数据、用户商业信息等),攻击者可能通过非法操作甚至冒用内部人员合法权限对云平台中各类资源进行窃取或破坏.云平台异常检测能够通过全面监控和分析云平台中的各种访问与操作,及时识别出潜在的内部威胁和异常情况,从而保证云服务中相关资源及其存储关键信息的安全.因此,云平台异常检测对保障云平台的安全性和高效响应能力至关重要.

当前,云服务场景中有多种类型的不同异常检测任务[1-4],领域专家对这些任务通常具备丰富的业务经验而对如何构建异常检测模型并不熟悉.云服务场景中异常检测任务所涉及各种异质资源不仅数量众多,而且相互之间往往包含高度复杂的关系.不同类型服务对这些资源的访问和操作会产生海量的日志和审计数据.由于数据规模极其庞大并且人工总结分析相关异常判定规则极其困难,在上述场景中仅通过专家人工监测和分析相关数据,难以实现准确高效的异常检测.现有基于机器学习的方法在进行上述任务时存在2方面问题:一方面,云平台中通常存在大量异常检测任务,尽管监控指标或应用场景都不尽相同,但其对应的异常检测模型构建具有相似性;然而,现有方法大多有针对性地对特定任务进行专门设计和调优,不具备对相关异常检测任务通用的模型构建能力.另一方面,在对模型进行专门设计和调优的过程中,十分依赖机器学习方面的专业知识,使得领域专家即使在已有相关算法的情况下仍难以搭建好对实际具体问题行之有效的模型.因此,如何设计一种能够便于领域专家自助使用的、对云平台中相关异常检测任务具有通用性的异常检测轻量化模型智能构建方案是十分重要的.这让领域专家在具备尽可能少的机器学习和深度学习知识的情况下,能够基于该方案快速搭建异常检测模型来实现面向目标任务的异常检测.

针对上述问题,本文提出了一种云平台中便于领域专家自助使用的异常检测轻量化模型智能构建方案,以实现:1)根据专家给定的异常检测任务自动地构建检测模型所需的特征输入;2)结合专家领域知识自动地进行特征选择以精简检测模型输入;3)自动地优化超参数以提升检测模型性能.该方案首先可基于特定的云平台异常检测任务和领域专家所提供的相关属性字段等配置,通过分别计算不同类别属性字段的相关统计量,自动构造检测模型所需的输入特征;然后利用互信息作为评价特征统计量之间的依赖表征指标,并结合专家领域知识自动地进行特征精简以优化检测模型输入;最后基于前2个步骤,自适应地构建基于自动编码器的轻量化异常检测模型,并通过超参数自动优化完成模型训练,以获得专家所定义异常检测任务的判定结果.

所提出方案无需专家手动创建规则对异常情况进行描述,能使领域专家在具备尽可能少的机器学习和深度学习知识的情况下,自助、快速地搭建异常检测模型以实现面向目标任务的异常检测.该方案的创新性主要包括以下3方面:

1)基于少量配置信息的模型输入特征自动构造.仅需通过专家给定异常检测任务的类型、目标描述和相关字段等基本配置信息,即可实现目标模型输入特征的自动生成,在大幅降低方案使用门槛的同时提升了模型构建的便捷性.

2)支持模型输入特征自动优化.基于互信息和重建损失的综合评分进行自动特征选择,可实现异常检测模型输入特征的自动优化,在显著降低训练时间的同时提高了模型的泛化能力.

3)支持不同类型异常检测任务的轻量化模型自动构建.通过所提出方案,领域专家可根据不同的目标异常检测任务配置相应信息完成模型描述;在此基础上,进行特征自动生成、特征自动优化和模型超参数自动优化,从而实现面向云平台异常检测任务

的轻量化模型智能构建.

1 相关工作

近年来，云平台中的内部威胁检测和异常检测重要性日益凸显. 本节从工业界和学术界2方面介绍相关研究工作.

1.1 工业界相关服务

当前，业界一些主流厂商如亚马逊、微软、谷歌、阿里云和腾讯云等公司都已推出了各自的云安全服务. 例如，亚马逊公司的CloudTrail、GuardDuty和Detective可分别用于活动跟踪、威胁检测和问题定位[5-7]；微软公司的Azure Defender和Azure Sentinel提供了相关的态势管理、负载保护和智能安全分析[8-11]；谷歌公司的Security Command Center专注于威胁防护和检测[12]；阿里云公司的网络应用防火墙(Web application firewall, WAF)和云安全访问服务针对Web应用和企业运营提供保护[13-14]；腾讯云公司的T-Sec高级威胁检测系统和T-Sec网络入侵防护系统识别网络攻击及高级威胁，并提供流量反欺诈功能[15-16]. 以应用程序编程接口(application programming interface, API)网关应用场景为例，现有技术大多通过WAF和机器学习方法对API请求进行异常检测与防护. WAF通过专家手动创建规则对API请求进行分析，虽然对已知类型异常有效但费时费力，且难以应对内部威胁和新型攻击. 机器学习方法需要针对云服务场景中特定的异常检测任务进行专门设计和调优. 当面临多个不同的异常检测任务时，需要逐一地专门构建多个模型.

本文拟研究应用程序编程接口网关(application programming interface gateway, APIG)应用场景中特定对象在一段时间内是否具有异常行为. 对国内外5家大型云服务提供商(亚马逊云AWS、微软Azure、Google Cloud、阿里云和腾讯云)的相关业务进行调研如表1所示，以比较本工作和现有产品的联系与区别. 国内两大厂商的云安全服务虽然具有某些异常检测能力，但缺乏APIG应用场景下的异常检测功能；AWS虽然有针对APIG的异常检测，但缺乏详细的功能性说明. 所提出方案落地后预期将成为我国在该细分领域自主研发的首个云安全产品.

Table 1 Cloud Security Relative Service Provided in Industry

表1 业界云安全相关服务提供情况

云服务提供商	云服务异常检测功能	APIG异常检测功能	技术细节介绍	APIG特定对象一段时间内的异常检测功能
AWS	√	√	×	×
Azure	√	×	N/A	×
Google Cloud	√	×	N/A	×
阿里云	√	×	N/A	×
腾讯云	√	×	N/A	×

1.2 学术界相关研究

针对云平台中的内部威胁检测和异常检测，学术界现有相关工作主要包括异常主机检测、异常日志检测和异常时段/会话检测等方面研究.

现有异常主机检测相关研究可分为基于传统机器学习和基于深度学习的方法. 传统机器学习方法通过数据驱动的方式学习数据的内在联系，从而提取特征进行异常检测. Le等人[17-18]基于大量特征工程，分别通过集成多种有监督与无监督机器学习框架进行内部威胁检测. Gu等人[19]基于主机行为日志与主机间通信流日志提取特征并进行聚类，以检测僵尸网络和僵尸节点. García等人[20]对已知的僵尸网络行为建模并聚类，判断待检测节点是否属于相应的簇，从而确定其是否为僵尸节点. 深度学习方法不需要大量特征工程，并能比传统方法更好地学习到主机的深层行为特征，从而显著提升异常检测性能[21-23]. Huang等人[24]提出了一个基于跳计数的异常节点检测模型，通过预测2节点之间的最短路径进行自监督学习. Abou Daya等人[3]采用基于图的深度学习方法以检测僵尸网络. Zheng等人[25]为每个节点构建不同的背景子图，并采用生成属性回归和多视图对比学习进行异常检测.

现有异常日志检测相关研究主要可分为有监督方法和无监督方法. 有监督方法通过同时学习正常日志和异常日志中蕴涵的模式，再根据待检测日志是否具有异常日志的特征来识别异常. Lu等人[26]使用卷积神经网络(convolutional neural network, CNN)挖掘日志间的复杂关系用于日志分类. 孙嘉等人[27]使用结合CNN网络和双向长短期记忆(long short-term memory, LSTM)网络的模型提取日志之间的空间和时间特征，并融合2种特征进行异常检测. 张圣林等人[28]提出的多语法日志通用异常检测机制LogMerge实现了不同类型日志之间的异常模式迁移，降低了异常样本的标注成本. 无监督方法不依赖于数据标签，因此适用于异常样本少、标注成本高等难以获得大量带标签样本的场景[29-31]. 该类方法对数据的正常模式进行描述，将不符合正常模式的数据判断为异常. Du等人[32]将日志建模为自然语言序列，并利用

LSTM 网络学习序列间的依赖关系以描述正常日志的行为模式, 进而实现异常日志的检测. Liu 等人[1] 通过一系列人工定义的规则建模日志之间的多种关联关系, 并提出了一种基于异质图嵌入的企业内部威胁检测方法.

某些情境下, 异常行为所包含的单条日志完全符合正常模式, 而多条日志所形成序列可能反映异常行为[33-35]. Wang 等人[36] 将原始日志转换为结构化日志事件, 并通过特征提取得到日志序列向量以进行异常检测. Zhang 等人[37] 利用基于注意力的双向 LSTM 网络建模日志向量序列, 捕获其上下文信息并衡量序列中不同部分的重要性以进行异常检测. Zhang 等人[38] 将原始日志数据转换为日志事件, 通过粒子群优化算法选择窗口长度并将日志事件划分为序列, 进而提取数值关系以检测异常. Zhang 等人[39] 利用多层卷积和 Transformer 模型分别提取日志序列的局部信息和全局信息以进行异常检测. Yang 等人[40] 通过嵌入模型获取日志序列语义信息, 并使用聚类方法估计无标签日志序列的概率标签, 最后基于注意力的门控循环单元提取特征以检测异常.

2 云平台中的异常检测轻量化模型智能构建方案

本节首先描述云平台异常检测问题; 然后对方案整体结构进行概述, 并详细介绍其各部分组成.

2.1 问题描述

云平台异常检测可通过部署在云服务器上的监测器(如 APIG)对用户的访问或使用行为进行记录和监控以发现其中蕴含的某些异常行为. 但由于账号泄露、内部威胁、权限分配错误等原因, 在单条用户日志中可能完全不体现任何异常特征. 因此, 需要对一段时间内用户日志中的相关信息进行分析, 并结合用户的历史情况考虑.

本研究通过分析云平台中相关审计日志, 旨在检测目标对象(如用户或主机)在特定时间段内是否发生异常行为. 具体地, 将目标对象在一段时间内(如 24 小时)的 API 审计日志作为待测样本原始输入数据, 基于日志中相关字段的统计信息分析异常本质与表征之间的潜在关联, 从而判定目标对象在特定时间段内是否发生异常行为.

2.2 方案概述

本文所提出的云平台中异常检测轻量化模型智能构建方案如图 1 所示.

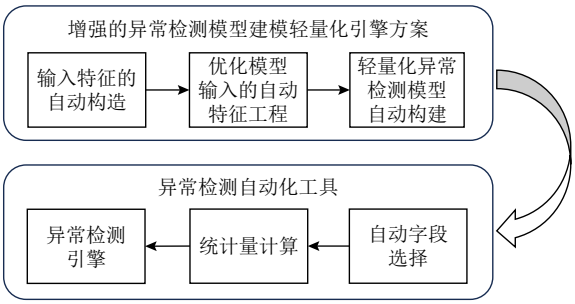


Fig. 1 Intelligent lightweight model construction solution for anomaly detection in cloud platforms

图 1 云平台中的异常检测轻量化模型智能构建方案

所提出方案由 3 个部分组成: 1)输入特征的自动构造. 根据特定的云平台异常检测任务和领域专家所提供的相关属性字段等配置, 分别计算不同类别属性字段的相关统计量以获得检测模型所需的输入特征. 2)优化模型输入的自动特征工程. 利用互信息作为评价特征统计量之间的依赖表征指标, 并结合专家领域知识所确定的与该特定任务相关的字段以进行模型的自动特征选择. 3)轻量化异常检测模型自动构建. 根据前 2 个步骤所得到的特征, 自适应地构建基于自动编码器的轻量化异常检测模型, 并通过超参数自动优化提升所构建模型的性能. 基于所提出方案, 设计了一个针对 APIG 应用场景的自助工具, 详见第 3 节的案例分析.

2.3 输入特征的自动构造

为了使异常检测模型能够适应各类 APIG 应用场景下的检测任务, 本方案通过一个包括 2 个步骤的过程, 以实现所需输入特征的自动构造. 2 个步骤具体为: 1)专家自助的异常检测任务配置, 以专家给定的云平台异常检测任务以及提供的相关属性字段等配置作为输入; 2)自动的模型输入特征计算, 方案基于配置信息输出后续异常检测模型所需的输入特征. 其具体介绍如下.

1)专家自助的异常检测任务配置

基于专家提供的任务相关描述信息, 明确异常检测任务的各项配置. 所提供信息具体包括: ①任务是否时序相关; ②基本统计量计算的时间粒度; ③异常检测的时间粒度; ④字段候选属性(包括已选字段、待选字段、丢弃字段); ⑤任务涉及的对象(要筛选的用户/主机/服务的标识符(identifier, ID). 其中, 任务是否时序相关决定了生成特征的种类; 字段候选属性用于帮助后续进行特征选择.

2)自动的模型输入特征计算

本方案将属性字段分为 4 类, 其分类标准与示例

如表 2 所示. 针对不同种类的属性, 自助计算以下 3 种特征中的 1 种或几种, 分别为特征统计属性、用户访问轮廓、访问轮廓相似性.

Table 2 Relationship Between Attribute Categories and Classification Standards

表 2 属性类别与分类标准的关系

属性类别	分类标准	示例
第一类	情景属性	服务名称
第二类	离散值（种类少且有顺序）	HTTP 状态码
第三类	离散值（种类多）	远程地址
第四类	连续值	HTTP 请求长度

特征统计属性包括基础统计量、描绘统计量和标准化统计量. ①基础统计量为日志时间粒度内服务与调用各 API 的统计信息, 如表 3 所示(统计时间范围为每个时段/全天 24 小时); ②描绘统计量包括: 最大值、平均值、数值标准差、比例标准差、信息熵、类别数; ③标准化统计量为反映单个样本值距离样本总体均值的相对标准距离的 Z 分数, 以反映各样本在描绘统计量上与历史观测期内样本总体间的差异.

Table 3 Basic Statistics

表 3 基础统计量

属性类别	基础统计量	含义
第 1 类	$Count^{i}_{service}(t)$	服务类型为 i 的日志数, i 为具体的服务种类.
	$AttCount^{i,j}_{1st}(t)$	服务类型为 i 且服务唯一标识为 j 的日志数, i 同上, j 为服务唯一标识的不同取值.
第 2 类	$AttCount^{i,j}_{2nd}(t)$	某第 2 类属性服务类型为 i 且取值为 j 的日志数, i 同上, j 为该类属性的不同取值.
第 3 类	$AttCount^{i,j}_{3rd}(t)$	某第 3 类属性服务类型为 i 且取值为 j 的日志数, i 和 j 同上.
第 4 类	$AttCount^{i,j}_{4th}(t)$	某第 4 类属性服务类型为 i 且取值为 j 的日志数, i 和 j 同上.

所有类别属性的基础统计量和标准化统计量均使用相同方式计算. 对于不同类别的属性, 其描绘统计量的计算个数有所区别, 如表 4 所示.

描绘统计量中, 最大值用于反映数据的上界, 均值用于反映数据的平均状况, 数值标准差用于反映数据的离散程度, 比例标准差用于反映数据比例上的不平衡性, 信息熵用于反映数据内在的不确定性, 种类数描述取值的多样性.

其中, 标准差计算如式(1)所示. N 表示该时段中该种统计计数的数量. 在数值标准差计算时, s_i 表示不同种类的统计计数; 在比例标准差计算时, s_i 表示不同种类的统计计数与计数总和的比值.

Table 4 Portrait Statistics Corresponding to Different Attributes

表 4 不同属性对应的描绘统计量

属性类别	最大值	平均值	数值标准差
第 1 类属性	是	是	是
第 2 类属性	是	是	是
第 3 类属性	是	是	是
第 4 类属性	否	是	是
属性类别	**比例标准差**	**信息熵**	**种类数**
第 1 类属性	否	是	是
第 2 类属性	是	是	是
第 3 类属性	是	是	是
第 4 类属性	否	否	否

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(s_i-\mu_c)^2}. \tag{1}$$

信息熵计算如式(2)所示. 其中, s_i^c 表示不同种类的统计计数. $p_r(s_i^c)$代表数据分布概率, N 表示该时段中该种统计计数的数量.

$$H = -\sum_{i=1}^{N} p_r\left(s_i^c\right)\mathrm{lb}p_r\left(s_i^c\right), \tag{2}$$

$$p_r(s_i^c) = \frac{s_i^c}{\sum_{i=1}^{N} s_i^c}. \tag{3}$$

标准化统计量通过描绘统计量计算 Z 分数得到. 对于特定时段内给定服务的每一个描绘统计量 S_i^{first}, 其 Z 分数计算如式(4)所示. 其中, μ为样本总体中该统计量的均值, σ为样本总体中该统计量的标准差.

$$Z_i = \frac{S_i^{\mathrm{first}} - \mu_s}{\sigma}. \tag{4}$$

对于用户访问轮廓的计算, 用户一天中特定服务或全部服务在每小时内的日志数与关于第 2 类和第 3 类属性的信息熵随时间的变化均为用户访问轮廓的描述. 记用户一天中特定服务或者全服务在每小时内的日志数(或信息熵)随时间的变化为一个 24 维的向量 $\boldsymbol{Cur}^{i,j} = (c^{i,j}(t_1), c^{i,j}(t_2), \cdots, c^{i,j}(t_{24}))$, 即为用户一天的访问轮廓, 其中 i 为服务的种类, j 为统计量的种类(日志数或信息熵).

基于用户访问轮廓相似性的行为属性可通过待测样本的用户访问轮廓与参照样本的用户访问轮廓间的余弦相似性得到. 对于训练集中每一个样本的用户访问轮廓 $\boldsymbol{A}$, 都与参照样本的用户访问轮廓 $\boldsymbol{B}$ 计算 1 次余弦相似性. 这里选取训练集中的第 1 个样本作为参照样本. 其中, 余弦相似性的计算方式如式(5)

所示.

$$sim = \cos(A,B) = \frac{A \times B}{\|A\| \times \|B\|}. \tag{5}$$

对所有计算得到的余弦相似性值做 Z 分数归一化如式(6)所示. 归一化后的 $\widehat{sim}(q)$ 可认为服从标准正态分布 $N(0,1)$. 根据 3 倍 delta 原则, 正常样本的 $\widehat{sim}(q)$ 在 [–3, 3] 的值域范围内, 超出这个值域范围的待测样本可能发生了异常行为.

$$\widehat{sim}(q) = \frac{sim(q) - \mu_{sim}}{\delta_{sim}}. \tag{6}$$

完成输入特征的自动构造后, 将所得到的特征统计属性、用户访问轮廓、访问轮廓相似性 3 部分合并可形成原始输入特征.

2.4 优化模型输入的自动特征工程

为了从原始数据中找到有用信息以获得较好的异常检测效果, 本方案通过自动特征工程优化异常检测模型输入. 具体包括 3 个步骤: 1)基于专家知识的字段分类; 2)基于互信息和重建损失的特征重要性评分; 3)基于专家决策的字段选择. 如图 2 所示, 该方案中的自动特征工程以 2.3 节得到的原始输入特征为输入, 输出优化后的、将作为下一步输入的字段列表. 其具体介绍如下:

1)基于专家知识的字段分类

在特征选择过程中, 需要明确已选字段和待选字段. 专家将所确定的、对模型性能一定能起到作用的字段作为初始的已选字段(即确定字段), 将认为可能对模型有一定作用的字段作为待选字段. 通过结合专家知识, 可以显著地减少计算量, 提高特征选择速度.

2)基于互信息和重建损失的特征重要性评分

对待选字段逐一计算其与已选字段(集)间的互信息; 基于每一轮综合计算得到的排序后互信息与其所对应模型在验证集上的性能选出当前轮“最优”的一个字段, 迭代直至待选字段列表为空; 根据迭代过程所得到的字段列表并综合各字段所对应的基于互信息的分数和基于重建损失的分数, 专家从所得到的字段列表中确定一个“阈值”, 以获得将作为下一步输入的字段列表. 其中, 互信息和相关分数的计算方式如下.

对于离散数据, 统计量之间互信息的计算公式如下:

$$I_{\text{stat}}(S_{\text{u}};S_{\text{c}}) = \sum_{s_{\text{u}},s_{\text{c}}} p_r(s_{\text{u}};s_{\text{c}}) \log \frac{p_r(s_{\text{u}};s_{\text{c}})}{p_r(s_{\text{u}})p_r(s_{\text{c}})}, \tag{7}$$

其中, $I_{\text{stat}}(\cdot;\cdot)$ 表示 2 个统计量之间的互信息, S_{u}、S_{c} 分别代表已选字段和待选字段中的某一统计量, s_{u}、s_{c} 分别表示 S_{u}、S_{c} 的取值, $p_r(\cdot;\cdot)$ 表示联合概率分布, $p_r(\cdot)$ 表示概率密度.

根据字段所对应统计量的互信息, 可以得到字段间的互信息. 将已选字段 F_{u} 和待选字段 F_{s} 分别记为 m_1 和 m_2 维, 通过多次迭代完成字段间互信息的计算, 迭代次数为这 2 个字段维数的较小值. 计算过程中, 遍历待选字段的统计量, 步骤如下: ①对待选字段中的每个统计量, 计算其与指定已选字段各统计量之间的互信息; ②选出这些互信息中的最大值, 作为当前待选字段统计量与已选字段对应统计量的互信息, 并将该最大互信息在已选字段中对应的统计

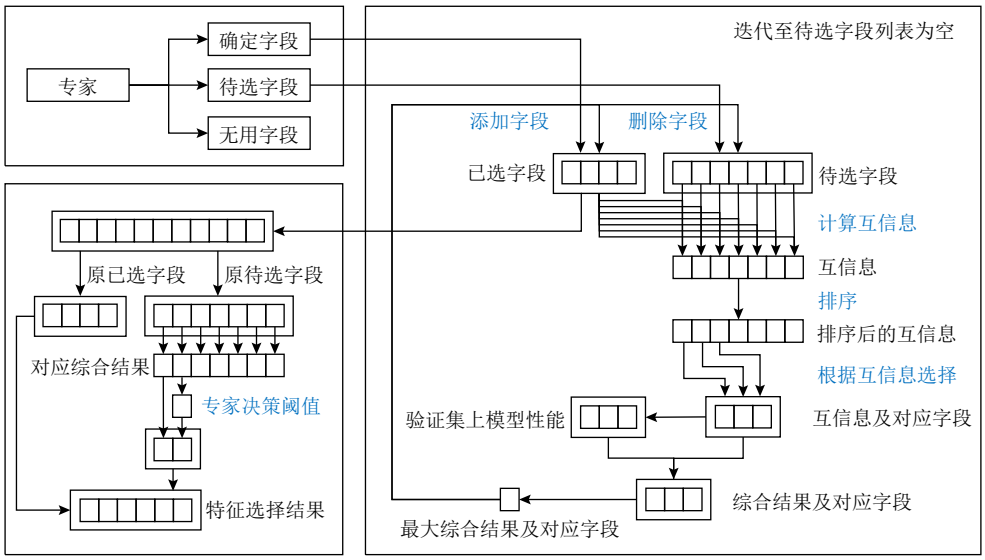


Fig. 2 Overall flow of automatic feature engineering

图 2 自动特征工程的整体流程

量标记为“已使用”，则在后续迭代中不再计算与标记为“已使用”统计量的互信息；③重复上述步骤，直到已选字段所有统计量被标记或待选字段所有统计量遍历完；④计算所有得到的互信息均值，作为当前待选字段与已选字段之间的互信息.

在得到一个待选字段分别与各个已选字段的互信息后，直接取其中的最大值作为当前待选字段与全部已选字段之间的互信息.

为了对所选字段重要性进行有效评价，本文定义了基于互信息的分数、基于重建损失的分数与基于互信息和重建损失的综合分数.

为了将互信息的数值映射到 0 与 1 之间，并且使得通过互信息反映的字段对模型的贡献呈正相关，则基于互信息的分数计算如下：

$$Score_I(I)=1-(2\times sigmoid(I)-1), \tag{8}$$

其中，$Score_I(\cdot)$ 表示根据互信息计算其分数，I 表示一个待选字段与全部已选字段之间的互信息.

为验证已选字段集合中添加一个待选字段后，使用对应训练数据训练所得到模型对验证数据的重建效果，需要计算与所有待选字段分别对应的基于重建损失的分数，以评估其对目标异常检任务的潜在提升. 基于重建损失的分数计算如式(9)所示. 其中，n 表示验证集的样本数量，$\boldsymbol{X}_i$ 表示验证集的第 i 个样本，m 表示验证集中每个样本的特征维度数量，X_i^d 和 $\hat{X}_i^d$ 分别表示验证集中第 i 个样本第 d 个维度的数值和重建值.

$$Loss\left(\boldsymbol{X};\hat{\boldsymbol{X}}\right)=\frac{1}{n}\sum_i\left(\frac{1}{m}\sum_d\left(\hat{X}_i^d-X_i^d\right)^2\right). \tag{9}$$

为了有效预估添加一个字段后模型性能的提升程度，字段综合分数的计算如式(10)所示. 其中，$Score(\cdot)$ 表示计算一个字段的综合分数，$field$ 表示一个字段名称，β 表示可调整参数(默认取 $\beta=1$).

$$Score(field)=\frac{1+\beta^2}{\dfrac{\beta^2}{Loss\left(\boldsymbol{X};\hat{\boldsymbol{X}}\right)}+\dfrac{1}{Score_I(I)}}. \tag{10}$$

3)基于专家决策的字段选择

根据迭代过程所得到的字段列表，并综合各字段得到的互信息分数和重建损失分数，专家确定截止轮次数，以决定下一步输入的字段列表. 具体地，专家通过观察新增字段在不同阶段的分数变化趋势，判断从哪一轮后新增字段对模型性能的提升作用较小. 基于所确定截止轮次数，筛选该轮之前(不包含)所选择的字段，丢弃该轮之后的字段，得到优化后的字段列表. 在优化后的字段列表中，字段的顺序即为原待选字段被选择的顺序.

2.5 轻量化异常检测模型自动构建

本方案设计了一种基于自动编码器(autoencoder, AE)的轻量化异常检测模型构建机制. 其所采用的异常检测模型以及模型超参数自动优化过程如下.

目标任务的异常检测模型可基于自动编码器构建，其结构具体如图 3 所示. 检测模型的输入由 2.4 节提取的优化特征组成. 所构建模型的主体部分基于一个自动编码器，其瓶颈层最优神经元个数由参数搜索确定. 在自动编码器中，编码器输入与解码器输出之间的均方根误差(root mean square error, RMSE)作为异常分数 S，用以判断是否发生异常. RMSE 超过阈值即判定为异常，反之为正常. 阈值的计算方式为：

$$threshold=mean(S)+\gamma\times std(S), \tag{11}$$

其中，$mean(\cdot)$ 表示均值计算，$std(\cdot)$ 表示标准差计算，γ 用于平衡这 2 部分数值对阈值的贡献.

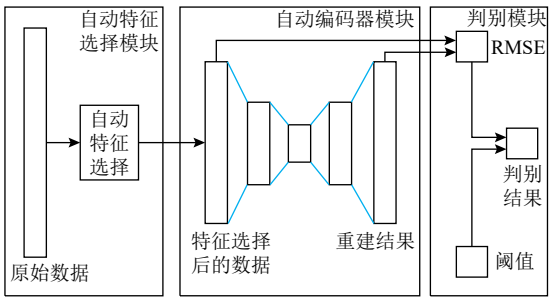


Fig. 3 Lightweight anomaly detection model based on autoencoders

图 3 基于自动编码器的轻量化异常检测模型

编码器和解码器均为 K=3 层全连接的多层感知机(编码器的 3 层分别为输入层、隐藏层、瓶颈层；解码器与之对应). 多层感知机中每层的隐藏元计算如式(12)所示. 其中，k 表示模型第 k 层，σ 为 sigmoid 激活函数，$\boldsymbol{W}$ 为权重矩阵，$\boldsymbol{b}$ 为偏置向量.

$$\boldsymbol{a}_k=\sigma(\boldsymbol{W}_k\times\boldsymbol{a}_{k-1}+\boldsymbol{b}_k). \tag{12}$$

模型具有众多超参数，包括优化器、学习率、激活函数、损失函数、参数初始化方法、轮数、批量大小、隐藏层层数、各隐藏层维度等. 在 APIG 应用场景下，根据模型优化的经验和实际测试，绝大多数超参数都已针对任务特点配置预定规则，使用时可不再进行搜索调优. 但部分超参数包括学习率、各隐藏层的维度对模型性能影响较大. 因此对不同训练场景，需要对这 2 种超参数自动优化.

1)自动编码器各隐藏层神经元个数的优化设置. 以 APIG 应用场景为例，在编码器和解码器均为 3 层

全连接结构的情况下, 记 $N_i(i=1, 2, 3)$ 为编码器第 i 层的维度, 对应解码器第 $4-i$ 层的维度. 自动编码器的输入和输出维度均为 N_1, 瓶颈层的维度为 N_3. 由于模型限制, 输入维度 N_1 的合理取值范围为 [64, 4 096]. 设 N_0 为不小于 N_1 的最小 2 的整数次幂, 即 $2^{\mathrm{ceil}(\mathrm{lb}N_1)}$. 当 $N_0\in[512, 4\,096]$, 即输入维度 $N_1\in(512, 4\,096]$ 时, N_2 取固定值 $N_0/4$, N_3 的搜索空间为 $\{8, 16, 32, \cdots, N_0/8\}$; 当 $N_0\in[64, 256]$, 即输入维度 $N_1\in(64,256]$ 时, N_2 取 $N_0/2$, N_3 的搜索空间为 $\{4, 8, 16, \cdots, N_0/4\}$. 每次计算不同瓶颈层维度取值对应的模型性能, 将性能最好的取值作为最优瓶颈层维度.

2)学习率的优化设置. 根据经验, 较好的学习率范围往往在 10^{-6}~10^{-1}, 将该范围作为搜索空间的区间. 在该区间中, 在对数域上均匀插入 M 个搜索点, 加上边界共 $M+2$ 个搜索点, 即 $m=0, 1, \cdots, M$ 时 $\exp\left(\frac{m}{M+1}\ln 10^{-6}+\left(1-\frac{m}{M+1}\right)\ln 10^{-1}\right)$. 每次计算学习率在这 $M+2$ 个点上对应的模型性能, 将性能最好的取值作为最优学习率.

3 案例分析

为了验证所提出方案在云平台中自动构建异常检测模型的能力, 本文以 APIG 应用场景中相关异常检测模型构建任务为例进行案例分析. 针对实际的具体异常检测需求, 本文基于所提出方案设计并实现了一个针对 APIG 应用场景的自助工具, 以帮助领域专家快速自助地构建目标检测模型.

该自助工具可让专家在具备尽可能少的机器学习和深度学习知识情况下, 帮助专家完成指定的异常检测任务. 主要流程为: 1)专家根据业务知识, 明确异常检测任务; 2)专家进行自助工具的初始配置, 并根据自助工具的提示提供符合所定义格式的输入, 如表 5 所示; 3)自助工具基于配置文件挑选字段并计算其特征统计量; 4)构建异常检测模型实现针对目标任务的异常检测.

Table 5 Expert Input Information Form

表 5 专家输入信息表

编号	描述	处理	默认
1	任务是否时序相关	时序相关时计算用户访问轮廓、访问轮廓相似性; 时序无关时不计算	时序相关
2	基本统计量计算的时间粒度		1 h
3	异常检测的时间粒度		24 h
4	候选属性字段及分类	标注该属性属于{确定字段、待选字段、无用字段}中的一种	
5	任务对象为要筛选的用户/主机/服务 ID		

如图 4 所示, 整个工具的运行流程包括自助字段选择、统计量计算和异常检测引擎 3 个部分. 自助工具通过这 3 个模块的输出与输入串联, 形成端到端的异常检测流程, 使不具备机器学习和深度学习知识的领域专家也能自助快速地完成异常检测任务.

自助工具的使用流程如图 5 所示. 首先, 领域专家确定待检测任务, 规范原始日志并进行自助工具

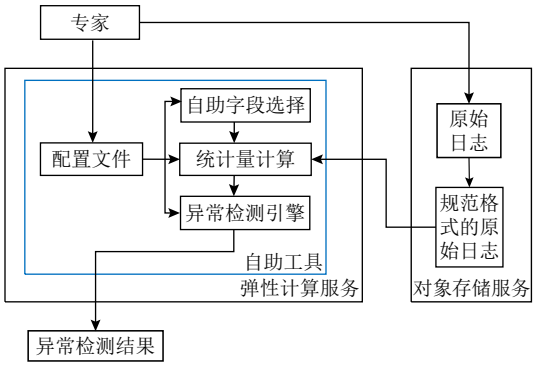


Fig. 4 Workflow of the self-service tool

图 4 自助工具的工作流程

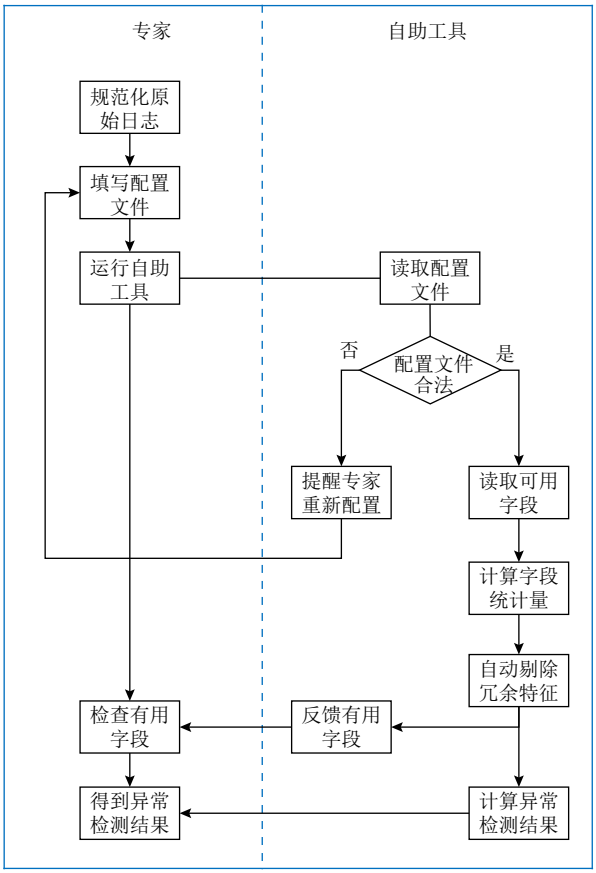


Fig. 5 Flow chart of the self-service tool usage

图 5 自助工具的使用流程图

的初始配置；自助工具读取配置文件，如有参数配置异常，则提醒专家进行修改. 然后，根据配置文件中的字段信息和时间粒度，自助工具调用统计量计算模块处理原始数据并保存结果. 接着，自助工具调用自动特征工程模块优化异常检测模型的输入. 最后，自助工具调用异常检测引擎进行目标异常检测模型的构建和训练，实现对检测样本的异常检测.

基于所提出方案设计的自助工具，领域专家可按照案例分析中的相关流程快速自动地构建具备良好检测能力的目标模型. 本研究通过使用该自助工具在实际 APIG 应用场景中实现了 9 种相关异常检测任务，具体过程详见第 4 节.

4 实验结果与分析

4.1 实验设置与数据集

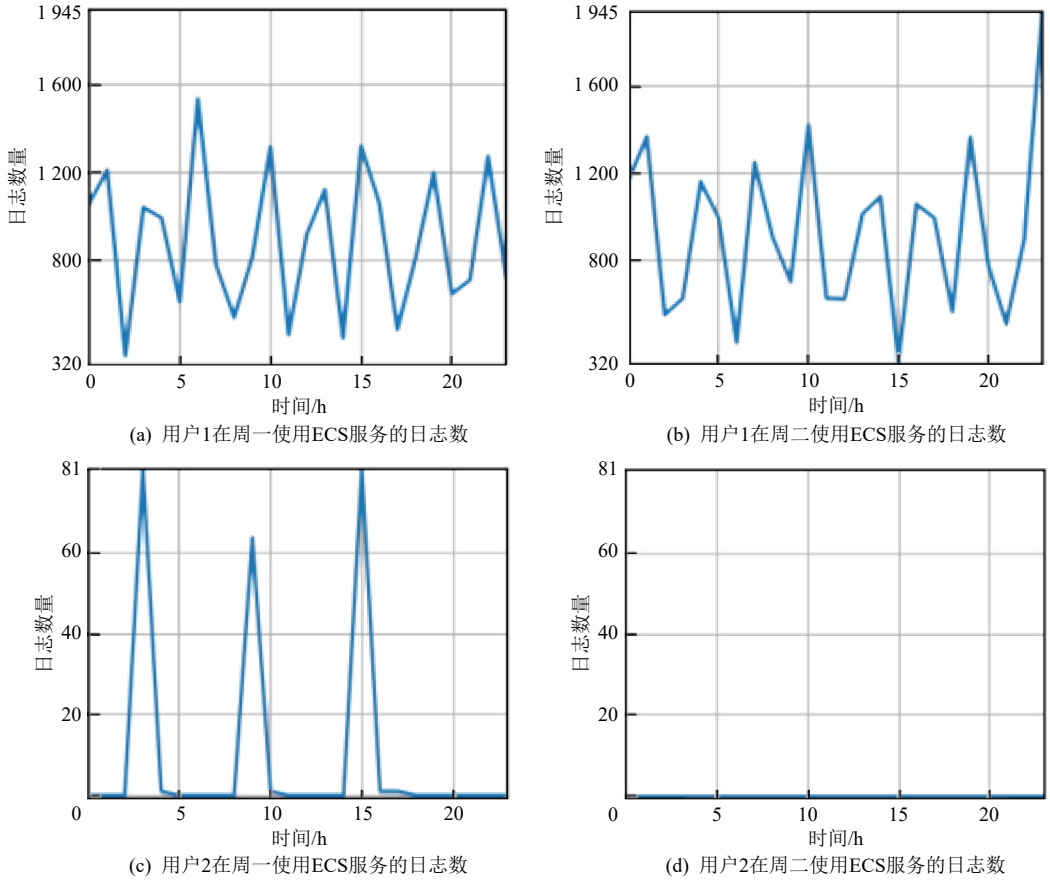
1）实验设置

本工作在企业真实云环境中收集了 APIG 应用场景相关服务的审计日志作为原始数据，以对所提出方案进行验证. 实验环境所使用 CPU 为 Intel Core i7-9700, GPU 为 Geforce GTX 1080Ti（11 GB），内存为 64 GB.

2）正常数据与异常数据获取

本研究基于 APIG 应用场景中的典型用户和重要服务获取真实数据. 通过对各类用户、服务所产生的审计日志数据进行分析发现：并不是每个用户每天都有日志记录，如图 6 所示；单一用户的某些服务即使每天都有日志记录，但其中部分小时的日志记录数也可能为 0，如见图 7 所示. 基于上述观测规律和实际数据情况，选取了 3 类典型用户：日志数量较多且活跃天数多的用户、日志数量一般且活跃天数少的用户，以及日志数量较少且活跃天数多的用户. 此外，基于领域专家经验选取了 3 种重要的云服务：弹性计算服务（elastic compute service, ECS）、弹性云硬盘 EVS（elastic volume service）、虚拟私有云（virtual private cloud, VPC）. 基于上述所选取 3 种典型用户与 3 种重要服务，总计组合获得 9 种典型实验场景.

正常数据可通过部署在云服务器上的 APIG 对用户的正常访问或使用行为进行记录和监控获得. 一个正常样本为目标用户在一天内关于特定服务的观测. 由于真实场景中缺乏异常标注，因而本工作人

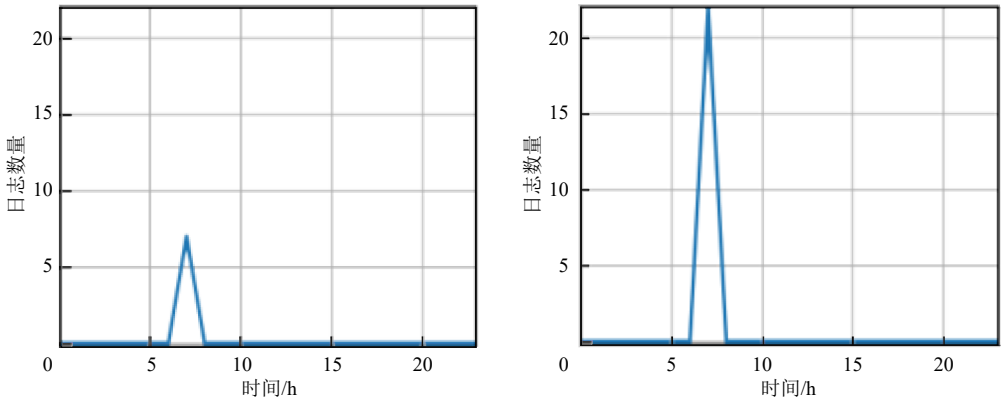


(a) 用户1在周一使用ECS服务的日志数 (b) 用户1在周二使用ECS服务的日志数

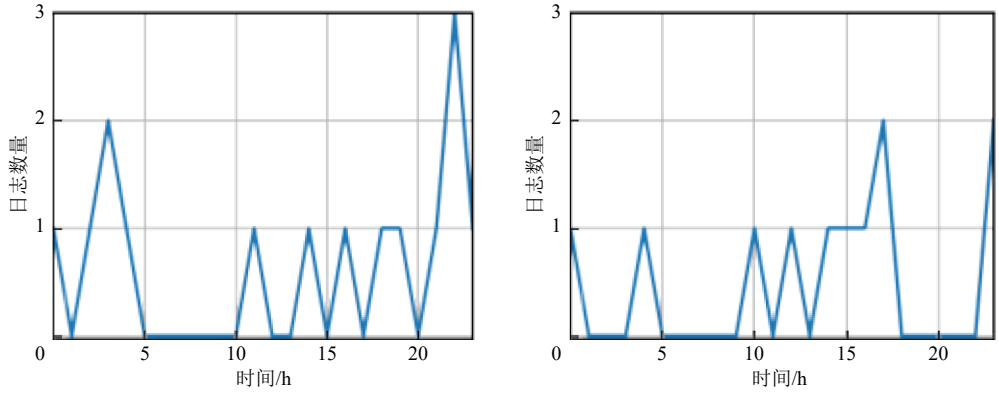
(c) 用户2在周一使用ECS服务的日志数 (d) 用户2在周二使用ECS服务的日志数

Fig. 6 The number of ECS logs used by two users over two consecutive days

图 6 2 个用户在连续 2 天内使用 ECS 的日志数



(a) 用户3在周一使用VPC服务的日志数 (a) 用户3在周二使用VPC服务的日志数



(b) 用户4在周一使用VPC服务的日志数 (b) 用户4在周二使用VPC服务的日志数

Fig. 7 The number of VPC service logs used by two users over two consecutive days

图 7 2 个用户在连续 2 天内使用 VPC 服务的日志数

工进行了异常数据生成. 异常数据生成过程中, 本研究考虑和保留了 API 审计日志蕴含的内在时序关联. 具体地, 异常数据生成包括以下步骤: 1)理解业务场景及相应字段物理意义, 定义典型的异常类型, 如表 6 所示; 2)根据异常类型, 分别确定针对真实背景数据的详细模拟攻击方式, 并实现各类异常的模拟和生成对应异常行为的审计日志; 3)将异常日志与正常日志记录融合, 形成含有异常的数据输入.

本研究从真实云环境中获取了 2021 年连续 6 周的相关审计日志作为模型训练的正常数据, 并以小时为时间粒度对日志进行划分; 另采集了 1 周的相关审计日志用于测试.

Table 6 Anomaly Types and Definitions

表 6 异常的种类与定义

异常种类	定义
访问频率全局升高	在一段时间内, 某用户访问某服务的次数在全局范围内突然升高.
访问频率局部升高	在一段时间内, 某用户访问某服务的次数从全局来看没有发生突变, 但是相对于该服务的历史访问次数出现了突变.
访问频率全局降低	在多个连续发生访问行为的时段中, 该段时间内访问频率全局大幅下降.
访问服务类型的群体规模异常	在某一时段, 用户访问的服务类型规模发生了明显的改变.
访问行为分布发生变化	在一段时间内, 访问行为由负倾斜分布转变为正倾斜分布.

4.2 评价指标

实验结果评价指标为检测率(*DR*)、误报率(*FPR*)、准确率(*ACC*)和 *AUC*(ROC 曲线下面积). *DR*、*FPR* 和 *ACC* 的计算公式为:

$$DR = TP/(TP+FN), \tag{13}$$

$$FPR = FP/(FP+TN), \tag{14}$$

$$ACC = (TN+TP)/(TN+FN+TP+FP), \tag{15}$$

其中, *FP* 是实际为正常样本且被误检为异常样本的样本数; *TN* 是实际为正常样本且被检测正确的样本数; *TP* 是实际为异常样本且被检测正确的样本数; *FN* 是实际为异常样本且被误检为正常样本的样本数. *AUC*[41] 用于衡量模型在不同阈值下区分正常样本和异常样本的能力, 其值越接近 1, 表明模型的检测性能越好.

4.3 对比实验

本节针对 4.1 节中介绍的 9 种典型实验场景分别测试所提出方案自助构建的异常检测模型以及其他基准方法的性能, 以验证所提出方案的有效性. 具

体地，如表 7 所示，3 个基准方法分别为基于自动编码器的异常检测模型(AE)[42]、基于 LSTM 网络的自动编码器异常检测模型(LSTM-AE)[43]以及基于单类支持向量机[30]的异常检测方法(OC-SVM)[44]. 3 个基准方法和所提出方法均采用无监督训练方式，神经网络模型损失函数均采用均方误差. VPC 服务场景下对典型用户 1、2、3 进行异常检测时测试集中正样本/异常样本数量分别为 7/35、7/21、7/35; EVS 服务场景下对典型用户 1、2、3 进行异常检测时测试集中正常样本/异常样本数量分别为 7/21、6/21、7/21; ECS 服务场景下对典型用户 1、2、3 进行异常检测时测试集中正常样本/异常样本数量分别为 7/21、7/21、7/21.

Table 7 Performance Comparison of Anomaly Detection Models When Using VPC Service

表 7 使用 VPC 服务时异常检测模型性能对比

检测模型	典型用户 1				典型用户 2				典型用户 3			
	DR	*FPR*	*ACC*	*AUC*	*DR*	*FPR*	*ACC*	*AUC*	*DR*	*FPR*	*ACC*	*AUC*
AE	0.228 6	0.000 0	0.614 3	1.000 0	1.000 0	0.000 0	1.000 0	1.000 0	0.885 7	0.000 0	0.942 9	0.963 3
LSTM-AE	0.828 6	0.000 0	0.914 3	1.000 0	1.000 0	1.000 0	0.500 0	1.000 0	0.971 4	0.714 3	0.628 6	0.959 2
OC-SVM	0.000 0	0.142 9	0.428 6	N/A	0.000 0	0.000 0	0.500 0	N/A	0.400 0	0.000 0	0.700 0	N/A
本文模型	0.971 4	0.000 0	0.985 7	1.000 0	0.904 8	0.000 0	0.952 4	1.000 0	0.885 7	0.000 0	0.942 9	0.963 3

注：N/A 表示不适用.

实验结果如表 8~10 所示. 观察实验结果发现，基于 LSTM-AE 的检测方法和基于 AE 的检测方法在 9 种典型实验场景的检测性能均显著优于基于 OC-SVM 方法的检测性能. 所提出方案自助构建的异常检测模型与基于 AE 方法的区别主要在于：1)不预先固定网络结构，根据输入智能搜索各隐藏层较优的神经元数量；2)无需人工调参，自动设置和搜索较优的各项超参数值. 观察实验结果发现，所提出方法在 9 种典型实验场景的 *AUC* 平均性能分别比基于 LSTM-AE 的检测方法和基于 AE 的检测方法高出 1.4 个百分点和 10.1 个百分点. 由于真实场景通常缺乏用以构建验证集的异常标注，故实验中基于领域专家经验设置了固定的判定阈值. 在该阈值设置下，相较于其余 3 种对比方法，所提出方法在绝大多数典型实验场景中能够对异常样本进行更好地判定. 上述结果表明，所提出方案自助构建的异常检测模型能够对典型场景中的各项云平台异常检测任务具有良好的检测能力.

4.4 消融实验

为了验证所提出方案中优化模型输入的自动特征工程是否有效，本节进行了消融实验. 针对 4.1 介绍的 9 种典型实验场景，图 8 和表 10 分别对比了所

Table 8 Performance Comparison of Anomaly Detection Models When Using EVS Service

表 8 使用 EVS 服务时异常检测模型性能对比

检测模型	典型用户 1				典型用户 2				典型用户 3			
	DR	*FPR*	*ACC*	*AUC*	*DR*	*FPR*	*ACC*	*AUC*	*DR*	*FPR*	*ACC*	*AUC*
AE	1.000 0	0.000 0	1.000 0	1.000 0	1.000 0	0.500 0	0.769 2	0.904 8	1.000 0	0.428 6	0.785 7	1.000 0
LSTM-AE	1.000 0	1.000 0	0.500 0	0.993 2	1.000 0	0.500 0	0.769 2	0.785 7	1.000 0	0.428 6	0.785 7	1.000 0
OC-SVM	0.000 0	0.571 4	0.214 3	N/A	0.000 0	1.000 0	0.000 0	N/A	0.000 0	0.428 6	0.285 7	N/A
Proposed	1.000 0	0.000 0	1.000 0	1.000 0	1.000 0	0.142 9	0.928 6	0.993 2	1.000 0	0.000 0	1.000 0	1.000 0

注：N/A 表示不适用.

Table 9 Performance Comparison of Anomaly Detection Models When Using ECS

表 9 使用 ECS 时异常检测模型性能对比

检测模型	典型用户 1				典型用户 2				典型用户 3			
	DR	*FPR*	*ACC*	*AUC*	*DR*	*FPR*	*ACC*	*AUC*	*DR*	*FPR*	*ACC*	*AUC*
AE	1.000 0	0.428 6	0.785 7	0.966 0	0.904 8	0.000 0	0.952 4	1.000 0	1.000 0	0.000 0	1.000 0	1.000 0
LSTM-AE	1.000 0	0.714 3	0.642 9	0.877 6	0.809 5	1.000 0	0.404 8	0.428 6	1.000 0	0.285 7	0.857 1	1.000 0
OC-SVM	0.000 0	0.714 3	0.142 9	N/A	0.095 2	0.000 0	0.547 6	N/A	0.000 0	0.571 4	0.214 3	N/A
Proposed	1.000 0	0.142 9	0.928 6	1.000 0	0.333 3	0.000 0	0.666 7	1.000 0	1.000 0	0.000 0	1.000 0	1.000 0

注：N/A 表示不适用.

Table 10 Performance of Anomaly Detection Models with/Without Feature Selection

表 10 使用和未使用特征选择时的异常检测模型性能

服务	检测模型	典型用户 1				典型用户 2				典型用户 3			
		DR	*FPR*	*ACC*	*AUC*	*DR*	*FPR*	*ACC*	*AUC*	*DR*	*FPR*	*ACC*	*AUC*
VPC	Proposed-	0.971 4	0.000 0	0.914 3	0.995 9	0.333 3	0.000 0	0.666 7	1.000 0	0.885 7	0.000 0	0.942 9	0.963 3
	Proposed	0.971 4	0.000 0	0.985 7	1.000 0	0.904 8	0.000 0	0.952 4	1.000 0	0.885 7	0.000 0	0.942 9	0.963 3
EVS	Proposed-	0.666 7	0.000 0	0.833 3	1.000 0	1.000 0	0.142 9	0.928 6	1.000 0	1.000 0	0.000 0	1.000 0	1.000 0
	Proposed	1.000 0	0.000 0	1.000 0	1.000 0	1.000 0	0.142 9	0.928 6	0.993 2	1.000 0	0.000 0	1.000 0	1.000 0
ECS	Proposed-	1.000 0	0.285 7	0.857 1	1.000 0	0.333 3	0.000 0	0.666 7	1.000 0	1.000 0	0.000 0	1.000 0	1.000 0
	Proposed	1.000 0	0.142 9	0.928 6	1.000 0	0.333 3	0.000 0	0.666 7	1.000 0	1.000 0	0.000 0	1.000 0	1.000 0

提出方法使用优化模型输入的自动特征工程和未使用该组件时每一轮的模型训练时间性能和模型性能. 其中, “Proposed-”和“Proposed”分别表示未使用和使用自动特征工程的检测模型. 实结果表明, 使用优化模型输入的自动特征工程模块后, 所提出方法能够在不影响测试性能的情况下, 显著降低训练时间; 使用自动特征工程模块后, 所提出方案自助构建的异常检测模型每轮训练时间平均减少 27.9%. 具体地, 在 9 种典型实验场景中, 所提出方法在使用优化模型输入的自动特征工程模块后具有以下优点: 1)特征选择后显著地降低了相关字段数量, 便于后续分析; 2)特征选择后, 在相同异常检测阈值设置方式下, 模型对异常样本的判定相较未进行特征选择具有优势.

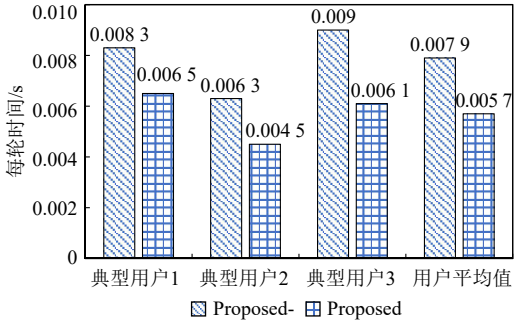


Fig. 8 Runtime per round with/without feature selection

图 8 使用和未使用特征选择时的每轮运行时间

5 结 论

本文提出了一种云平台异常检测轻量化模型智能构建方案. 与现有工作相比, 所提出的异常检测轻量化模型智能构建方案的优势在于: 1)无需专门的设计和调优, 能够针对特定的云平台异常检测任务快速自助地搭建异常检测模型, 实现面向目标任务的异常检测; 2)让领域专家在具备尽可能少的机器学习和深度学习知识的情况下, 能够基于该方案快速搭建异常检测模型, 实现面向目标任务的异常检测模型. 通过真实场景的案例分析和实验表明, 所提出方案能针对 APIG 应用场景异常检测任务快速自助搭建模型并表现出良好检测能力.

作者贡献声明: 周颖杰负责整体方案设计、论文撰写指导与修改; 杨敏负责论文撰写与修改、实验数据分析; 吴迪、刘凡兴和赵伟负责实验数据分析与处理以及方法设计与实现; 邓怡然负责实验方案设计与修改、相关文献调研; 吕建成负责方案和论文修改.

参 考 文 献

[1] Liu Fucheng, Wen Yu, Zhang Dongxue, et al. Log2vec: A heterogeneous graph embedding based approach for detecting cyber threats within enterprise[C]//Proc of the 2019 ACM SIGSAC Conf on Computer and Communications Security (CCS'19). New York: ACM, 2019: 1777−1794

[2] Pan Yuting, Lin Li. A trust-based DDoS discovery approach for encrypted traffic in cloud environment[J]. Journal of Computer Research and Development, 2021, 58(4): 822−833 (in Chinese)
(潘雨婷, 林莉. 云环境下一种基于信任的加密流量 DDoS 发现方法[J]. 计算机研究与发展, 2021, 58(4): 822−833)

[3] Abou Daya A, Salahuddin M A, Limam N, et al. Botchase: Graph-based bot detection using machine learning[J]. IEEE Transactions on Network and Service Management, 2020, 17(1): 15−29

[4] Liu Qixu, Chen Yanhui, Ni Jieshuo, et al. Survey on machine learning-based anomaly detection for industrial Internet[J]. Journal of Computer Research and Development, 2022, 59(5): 994−1014 (in Chinese)
(刘奇旭, 陈艳辉, 尼杰硕, 等. 基于机器学习的工业互联网入侵检测综述[J]. 计算机研究与发展, 2022, 59(5): 994−1014)

[5] Amazon Web Services. CloudTrail service documentation center[EB/OL]. [2024-06-04]. https://aws.amazon.com/cn/cloudtrail/

[6] Amazon Web Services. GuardDuty service documentation center[EB/OL]. [2024-06-04]. https://aws.amazon.com/cn/guardduty/

[7] Amazon Web Services. Detective service documentation center[EB/OL]. [2024-06-04]. https://aws.amazon.com/cn/detective/

[8] Microsoft Azure. Azure defender service overview[EB/OL]. [2024-06-04]. https://azure.microsoft.com/en-us/services/azure-defender/#overview

[9] Microsoft Azure. Azure sentinel service overview[EB/OL]. [2024-06-04]. https://azure.microsoft.com/en-us/services/azure-sentinel/#overview

[10] Microsoft. Cloud Apps defender service overview[EB/OL]. [2024-06-04]. https://www.microsoft.com/en-us/security/business/cloud-apps-defender

[11] Microsoft Azure. Azure defender for IoT service overview[EB/OL]. [2024-06-04]. https://azure.microsoft.com/en-us/services/iot-defender/#overview

[12] Google Cloud. Security command center service documentation[EB/OL]. [2024-06-04]. https://cloud.google.com/security-command-center

[13] Alibaba Cloud. Web application firewall (WAF) service documentation[EB/OL]. [2024-06-04]. https://www.aliyun.com/product/waf

[14] Alibaba Cloud. Cloud security access service (CSAS) documentation [EB/OL]. [2024-06-04]. https://www.aliyun.com/product/security/csas

[15] Tencent Cloud. Network threat analytics (NTA) service documentation[EB/OL]. [2024-06-04]. https://cloud.tencent.com/product/nta

[16] Tencent Cloud. Network intrusion prevention system (NIPS) service documentation[EB/OL]. [2024-06-04]. https://cloud.tencent.com/product/ips

[17] Le D C, Zincir-Heywood N, Heywood M I. Analyzing data granularity levels for insider threat detection using machine learning[J]. IEEE Transactions on Network and Service Management, 2020, 17(1): 30−44

[18] Le D C, Zincir-Heywood N. Anomaly detection for insider threats using unsupervised ensembles[J]. IEEE Transactions on Network and Service Management, 2021, 18(2): 1152−1164

[19] Gu Guofei, Perdisci R, Zhang Junjie, et al. Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection[C]//Proc of the 17th Conf on Security Symp (SS'08). Berkeley, CA: USENIX Association, 2008: 139−154

[20] García S, Grill M, Stiborek J, et al. An empirical comparison of botnet detection methods[J]. Computers & Security, 2014, 45: 100−123

[21] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. Nature, 1986, 323(6088): 533−536

[22] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735−1780

[23] Liu Yixin, Li Zhao, Pan Shirui, et al. Anomaly detection on attributed networks via contrastive self-supervised learning[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 33(6): 2378−2392

[24] Huang Tianjin, Pei Yulong, Menkovski V, et al. Hop-count based self-supervised anomaly detection on attributed networks[C]//Proc of Machine Learning and Knowledge Discovery in Databases. Berlin: Springer, 2023: 225−241

[25] Zheng Yu, Jin Ming, Liu Yixin, et al. Generative and contrastive self-supervised learning for graph anomaly detection[J]. IEEE Transactions on Knowledge and Data Engineering, 2021, 35(12): 12220−12233

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

[27] Sun Jia, Zhang Jianhui, Bu Youjun, et al. Log anomaly detection method based on CNN-BiLSTM model[J]. Computer Engineering, 2022, 48(7): 151−158 (in Chinese)
(孙嘉, 张建辉, 卜佑军, 等. 基于 CNN-BiLSTM 模型的日志异常检测方法[J]. 计算机工程, 2022, 48(7): 151−158)

[28] Zhang Shenglin, Li Dongwen, Sun Yongqian, et al. Unified anomaly detection for syntactically diverse logs in cloud datacenter[J]. Journal of Computer Research and Development, 2020, 57(4): 778−790 (in Chinese)
(张圣林, 李东闻, 孙永谦, 等. 面向云数据中心多语法日志通用异常检测机制[J]. 计算机研究与发展, 2020, 57(4): 778−790)

[29] Yuan Yali, Sripriya S A, Lin Mingkai, et al. Ada: Adaptive deep log anomaly detector[C]//Proc of IEEE Conf on Computer Communications (IEEE INFOCOM 2020). Piscataway, NJ: IEEE, 2020: 2449−2458

[30] Fang Xiaoyu, Cao Chenhan, Xia Bin. Attention based log-level anomaly detection algorithm for large-scale system logs[J]. Journal of Nanjing University, 2021, 57(5): 785−792 (in Chinese)
(房笑宇, 曹陈涵, 夏彬. 基于注意力机制的大规模系统日志异常检测方法[J]. 南京大学学报, 2021, 57(5): 785−792)

[31] Meng Weibin, Liu Ying, Zhu Yichen, et al. Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs[C]//Proc of the 28th Int Joint Conf on Artificial Intelligence (IJCAI'19). Palo Alto, CA: AAAI, 2019: 4739−4745

[32] Du Min, Li Feifei, Zheng Guineng, et al. DeepLog: Anomaly detection and diagnosis from system logs through deep learning[C]//Proc of the 2017 ACM SIGSAC Conf on Computer and Communications Security (CCS'17). New York: ACM, 2017: 1285−1298

[33] Yuan Shuhan, Zheng Panpan, Wu Xintao, et al. Insider threat detection via hierarchical neural temporal point processes[C]//Proc of IEEE Int Conf on Big Data (Big Data). Piscataway, NJ: IEEE, 2019: 1343−1350

[34] Chen Rui, Zhang Shenglin, Li Dongwen, et al. LogTransfer: Cross-system log anomaly detection for software systems with transfer learning[C]//Proc of the 2020 IEEE 31st Int Symp on Software Reliability Engineering (ISSRE). Piscataway, NJ: IEEE, 2020: 37−47

[35] Zhou Pengpeng, Wang Yang, Li Zhenyu, et al. Logsayer: Log pattern-driven cloud component anomaly diagnosis with machine learning[C]//Proc of IEEE/ACM 28th Int Symp on Quality of Service (IWQoS). Piscataway, NJ: IEEE, 2020: 1−10

[36] Wang Jin, Tang Yangning, He Shiming, et al. LogEvent2vec: LogEvent-to-vector based anomaly detection for large-scale logs in internet of things[J]. Sensors, 2020, 20(9): 2451

[37] Zhang Xu, Xu Yong, Lin Qingwei, et al. Robust log-based anomaly detection on unstable log data[C]//Proc of the 27th ACM Joint Meeting on European Software Engineering Conf and Symp on the Foundations of Software Engineering (ESEC/FSE' 2019). New York: ACM, 2019: 807−817

[38] Zhang Bo, Zhang Hongyu, Moscato P, et al. Anomaly detection via mining numerical workflow relations from logs[C]//Proc of Int Symp on Reliable Distributed Systems (SRDS). Piscataway, NJ: IEEE, 2020: 195−204

[39] Zhang Chunkai, Wang Xinyu, Zhang Hongye, et al. Log sequence anomaly detection based on local information extraction and globally sparse Transformer model[J]. IEEE Transactions on Network and Service Management, 2021, 18(4): 4119−4133

[40] Yang Lin, Chen Junjie, Wang Zan, et al. Semi-supervised log-based anomaly detection via probabilistic label estimation[C]//Proc of IEEE/ACM 43rd Int Conf on Software Engineering (ICSE). Piscataway, NJ: IEEE, 2021: 1448−1460

[41] Jin Linpeng, Dong Jun. Deep learning research on clinical electrocardiogram analysis[J]. Science China: Information Sciences, 2015, 45(3): 398−416

[42] Bourlard H, Kamp Y. Auto-association by multilayer perceptrons and singular value decomposition[J]. Biological Cybernetics, 1988, 59(45): 291−294

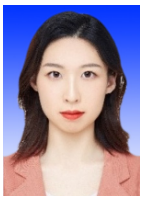
[43] Garg A, Zhang W, Samaran J, et al. An evaluation of anomaly detection and diagnosis in multivariate time series[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(6): 2508−2517

[44] Miao Xuedan, Liu Ying, Zhao Haiquan, et al. Distributed online one-class support vector machine for anomaly detection over networks[J]. IEEE Transactions on Cybernetics, 2019, 49(4): 1475−1488



Zhou Yingjie, born in 1984. PhD, associate professor, PhD supervisor. Senior member of CCF. His main research interests include intelligent behavioral data analysis and anomaly detection.

周颖杰, 1984 年生. 博士, 副教授, 博士生导师. CCF 高级会员. 主要研究方向为行为数据智能分析、异常检测.



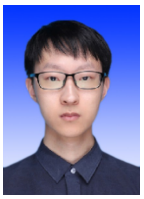
Yang Min, born in 1997. PhD candidate. Her main research interests include network behavior analysis and machine learning.

杨　敏, 1997 年生. 博士研究生. 主要研究方向为网络行为分析、机器学习.



Wu Di, born in 1999. Master. His main research interest includes network abnormal behavior detection.

吴　迪, 1999 年生. 硕士. 主要研究方向为网络异常行为检测.



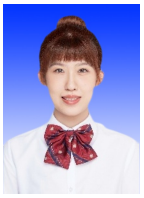
Liu Fanxing, born in 1998. PhD candidate. His main research interests include time series anomaly detection and artificial intelligence.

刘凡兴, 1998 年生. 博士研究生. 主要研究方向为时间序列异常检测、人工智能.



Zhao Wei, born in 1995. PhD candidate. His main research interests include defect detection, image segmentation, and time series analysis.

赵　伟, 1995 年生. 博士研究生. 主要研究方向为缺陷检测、图像分割、时序分析.



Deng Yiran, born in 1998. Master. Her main research interests include time series analysis and anomaly detection.

邓怡然, 1998 年生. 硕士. 主要研究方向为时序分析、异常检测.



Lü Jiancheng, born in 1973. PhD, professor, PhD supervisor. Senior member of CCF. His main research interest includes artificial intelligence and its applications.

吕建成, 1973 年生. 博士, 教授, 博士生导师. CCF 高级会员. 主要研究方向为人工智能及应用.