基于联邦迁移学习的应用系统日志异常检测研究

曾闽川，方 勇，许益家
(四川大学网络空间安全学院，成都 610065)

摘 要：迄今为止，基于日志的异常检测研究已经取得了很多进展，然而，在现实条件下仍旧存在两个挑战：(1) 是日志数据通常以“数据孤岛”形式储存在不同的服务器上，单一公司或组织的日志数据中异常样本量不足，且异常模式较为固定，很难通过这些数据训练出一个准确率高的检测模型. 为了解决这个问题，将不同来源的日志数据整合成更大的数据集可以提高模型训练的效果但可能会在数据传输过程中产生日志数据泄露问题；(2) 是不同应用系统类型的日志数据通常在结构和语法上存在差异，简单地整合并用于训练模型效果不佳. 基于以上原因，本文提出一种基于联邦迁移学习的日志异常检测模型训练框架 LogFTL，该框架利用基于匹配平均的联邦学习算法，在保证客户端数据隐私安全的前提下于服务器聚合客户端的模型参数形成全局模型，再将全局模型分发给客户端并基于客户端的本地数据进行迁移学习，优化客户端本地模型针对自身常见异常行为的检测能力. 经过实验表明，本文提出的 LogFTL 框架在联邦学习场景下效果超过了传统的日志异常检测方法，同时也证明了该框架中迁移学习的效果.

关键词：日志异常检测；联邦学习；迁移学习；LSTM；数据孤岛
中图分类号：TP391.1 **文献标识码**：A DOI：10.19907/j.0490-6756.2023.033002

Research on application system log anomaly detection based on federated transfer learning

ZENG Min-Chuan, *FANG Yong*, *XU Yi-Jia*
(School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China)

Abstract: Significant progress has been made in the research of log anomaly detection. However, two challenges still exist in reality. Firstly, log data is often stored on different servers, creating " data islands", the number of abnormal samples in the log data of a single company or organization is insufficient and the abnormal patterns are relatively limited, it is a challenge to train a detection model with high accuracy through these data. Integrating log data from different sources can improve the model's performance but may result in log data leakage during transmission; Secondly, the log data of different application system types varies in log structure and syntax, and simple integration for training models is ineffective. To address these issues, this paper proposes a log anomaly detection training framework called LogFTL based on federated transfer learning, which uses federated learning algorithm based on matching average. On the premise of ensuring the privacy and security of the client's data, LogFTL aggregates the model parameters of the client on the server side to form a global model which is then dis-

收稿日期：2023-01-04
基金项目：国家自然科学基金(U20B2045)
作者简介：曾闽川(1998－)，男，四川乐山人，硕士研究生，研究方向为网络信息对抗.E-mail：2422342691@qq.com
通讯作者：许益家.E-mail：xuyijia@stu.scu.edu.cn

tributed to the client side. Using the client's local data, the LogFTL framework migrates and learns to optimize the client's local model and the detection effect of local log data is improved. The experiment resluts show that the LogFTL framework proposed in this paper outperforms traditional log anomaly detection methods in federated learning scenarios, and demonstrate the transfer learning effectiveness of LogFTL.

Keywords: Log anomaly detection; Federal learning; Transfer learning; LSTM; Data islands

1 引 言

随着云计算产业的蓬勃发展,众多应用系统、业务系统依托分布式服务器构建计算集群,导致服务器数量快速增长,同时对服务器的稳定性和可靠性有较高的要求. 系统出现异常可能引发程度较为严重的后果,导致直接或间接损失[1]. 如果能够及时、准确地对系统产生的异常进行检测,能够帮助提高运维人员的响应速度,增加解决异常的应急时间,提高各类应用系统的稳定性.

运行日志记录着应用系统中不同级别的运行信息,是判断软件系统运行状态的重要数据. 通过分析日志数据能够检测出应用系统异常并定位问题产生原因[2-5]. 现有的基于日志的应用系统异常检测存在两个挑战:(1)日志数据以"数据孤岛"形式存在于不同服务器中且同一应用系统日志数据中的异常模式较为固定. 在实际环境中,不同的组织、机构各自拥有大量的日志,但日志数据中往往包含众多敏感数据,直接共享这些数据可能会产生隐私泄露问题,因此形成了"数据孤岛"问题. 并且日志数据作为运行信息的记录,正常样本数量占比远大于异常样本数量,这些日志数据中只有一小部分包含有价值的信息. 同时,单一应用系统由于其执行的业务和功能可能触发的异常行为模型较为固定,通过单一来源的日志数据训练出来的模型只能学习到一部分异常行为模型.(2)不同类型的日志在结构和语法上存在差异. 例如常见的基于 Hadoop 的分布式文件系统就存在多种类型的日志结构和语法[5]. 针对每一种日志类型训练一个检测模型成本高昂,且对于数据较少的日志类型效果不佳.

针对以上问题,本文设计实现了一种基于联邦迁移学习的应用系统日志异常检测框架(Log Anomaly Detection based on Federated Transfer Learning,LogFTL). LogFTL 通过基于匹配平均的联邦学习将不同参与者的本地模型进行聚合以建立检测能力更加全面的全局模型[7],同时能够更好地保护用户的隐私,打消参与机构将自己数据分享给别人会产生隐私泄露问题的顾虑. 然后利用迁移学习方法来实现不同类型应用系统日志数据之间的知识迁移,降低日志结构和语法对模型训练效果的影响,既能让本地模型拥有不同类型应用系统日志类型的检测能力又能提高本地模型对本地日志数据识别的准确度. 本文的主要贡献如下:(1) 提出并设计了一个基于联邦迁移学习的应用系统日志异常检测框架 LogFTL,该框架利用了基于匹配平均的联邦学习方法和基于长短期记忆网络(Long Short-Term Memory, LSTM)的迁移学习方法,用于解决现实环境中日志数据量不足且不同类型应用系统日志数据结构和语法存在差异的问题;(2) 提出了一种基于长短期记忆网络的迁移学习方法,能够基于 LSTM 模型实现联邦学习条件下的迁移学习,既能让本地模型拥有不同类型应用系统日志的检测能力又能保证本地模型对于本地应用系统日志异常行为的检测能力;(3) 使用真实的日志数据集对 LogFTL 进行了充分的评估实验,实验结果表明,本文提出的方法在克服"数据孤岛"问题后检测效果优于现有的日志异常检测方法.

2 相关工作

日志作为系统正常运行状态的记录,每一条数据都承载系统的运行信息. 通常而言,日志根据事件类型使用不同的文本用以描述事件,同时根据具体的运行状态动态生成参数,直观的如时间. 通过分析日志数据,我们可以知道一个系统的运行是否正常,如果不正常也能通过日志数据定位到问题产生的原因和位置. 但是为了节省服务器存储空间,提高系统运行效率,大部分应用系统很少以结构化的方式对日志数据进行处理,这提高了研究人员分析日志的难度. 因此当前主流的日志异常检测方法都需要首先对日志进行解析后再训练异常检测模型[8].

日志解析是将日志进行结构化,抽取出日志中

的文本信息和参数信息. 一开始，日志事件提取通过正则表达式来提取[9]. 然而这种方式对于正则表达式的设计要求很高，并且十分耗时. 现阶段日志解析最常用的方法是日志模板生成. 本文应用 Drain 这一工具对日志数据进行初步解析[10]. Drain 基于一个固定深度的树，使用节点集合设计的规则来解析日志，是目前性能较好的日志解析方法.

基于日志的异常检测算法可以分为有监督学习方法和无监督学习方法[11]. 有监督学习包括逻辑回归、决策树[12]和 SVM[13]. 虽然有监督学习可以在异常检测中获得很高的性能，但是它需要大量带有标签的训练数据，在隐私安全的前提下难以获得足够的数据进行训练，也不能同时针对多种应用系统类型日志.典型的无监督学习方法有 PCA[3]、LogCluster[14]和 DeepLog[4]. PCA 基于主成分分析的多元时间序列的降维方法依据累积贡献率选择主成分序列；LogCluster 通过线性模式对日志异常进行聚类；DeepLog 从正常执行中自动学习日志模式，检测时如果日志模式与正常执行中学习到的模式存在偏离，则报告异常. 实际上，无监督的方法在标记的训练数据不完全可用的情况下是比较实用的，但它们中大多数的检测精度较低. 随着自然语言处理的快速发展，许多基于自然语言处理的方法被提出. 例如 LogRobust 利用注意力机制的双向长短期记忆神经网络来识别异常日志[15]. LogAnomaly 根据事件的顺序和数量信息发现异常日志[16].

随着各方对“数据孤岛”现象和隐私安全问题愈发重视，基于联邦学习的日志异常检测算法也随之出现. 联邦学习最早是由谷歌提出来的一种模型训练框架[17]. 它提出的目的是要解决分布式数据的敏感信息泄露问题，能够在每个全局迭代中聚合来自各个参与者的本地模型以更新全局模型. FLOGCNN 方法设计了一个轻量级的卷积神经网络在联邦学习场景下对日志异常行为进行检测[18].

而 LogTransfer 则通过迁移学习将源系统的日志数据知识迁移到目标系统的日志检测模型提高目标系统的检测能力，但是它只提高目标系统的检测能力，无法让所有参与者受益[6].

3　研究方法

3.1　方法概述

LogFTL 的目标是在保障隐私安全的前提下，通过联邦迁移学习将“数据孤岛”场景下的日志数据进行聚合并且实现不同类型应用系统日志数据的知识迁移. LogFTL 的架构如图 1 所示. 我们假设有 N 个参与者愿意参与目标模型的训练，并有一个服务器 S 负责聚合全局模型，使用 $\{C_1, C_2, C_3, \cdots, C_N\}$ 来表示参与者，使用 $\{D_1, D_2, D_3, \cdots, D_N\}$ 表示他们提供的数据. 我们要利用这些分布在不同组织中的数据训练出各自的本地模型 $\{M_1, M_2, M_3, \cdots, M_N\}$，并将这些本地模型的参数上传到服务器 S 聚合成全局模型 M_{FED}，其中任何用户 C_i 不将其数据 D_i 暴露给其他参与者和服务端 S. 整个方法包含以下 4 个步骤.

(1) 日志数据表征构建. LogFTL 将所有参与者的日志数据通过 Drain 进行日志解析，提取日志模板，并且将日志样本与模板进行匹配后进行标识. 在对日志进行解析后，一条日志样本被表述为由多条日志数据组成的日志序列，其中每条日志数据有其对应模板的标识. 为了捕捉日志的语义信息，我们利用通过 word2vec[20] 预训练单词向量来替换相应的标识. 通过这种方式，模板的表示方法将语法的影响降到最低，同时保留了日志的语义信息.

(2) 训练初始模型与分发给用户. 在服务器端利用初始日志数据训练初始的全局模型，并将其模型参数分发给所有参与训练的用户.

(3) 使用联邦学习聚合用户本地模型参数. 每个用户可以在初始模型上基于各自的数据上训练本地模型，训练完成之后将本地模型上传到服务器端聚合至全局模型.

(4) 使用迁移学习增强本地模型对于本地日志数据的检测能力. 将全局模型分发给参与者后，让参与者通过迁移学习在本地数据上对模型进行迁移学习训练以获得更适用于当前参与者场景的本地模型.

本文提出方法的总体框架如图 1 所示.

3.2　日志数据表征构建

进行日志数据表征构建的目的是为了最大限度地保留日志的语义信息，同时尽量减少语法的影响. 在对原始的日志数据进行表征构建转化成日志序列向量后，这些向量将被作为本地模型训练的输入.我们首先使用 Drain 对原始日志文本进行解析，输出解析后的日志模板以及与原日志文本一一对应的日志结构化数据. 将日志按照区块进行整理后形成日志序列，再将日志序列中的每一个日志

事件与其模板进行映射,最终得到一个以日志模板表示的日志序列;之后,通过 word2vec 对整个日志序列进行词嵌入,得到该日志序列的向量化表达.其过程如图 2 所示.

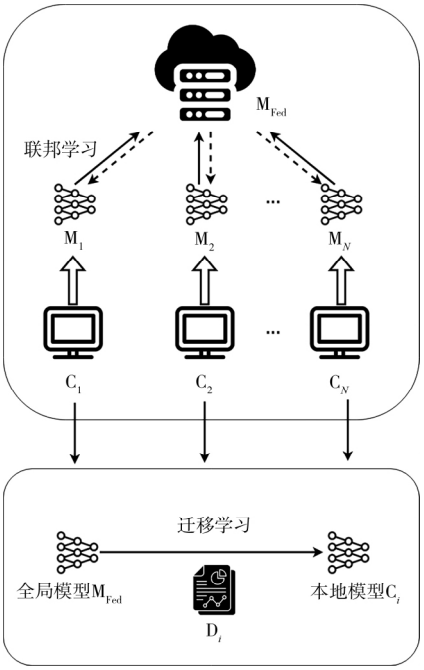

图 1　基于联邦迁移学习的日志异常检测框架

Fig.1　Log anomaly detection framework based on federated transfer learning

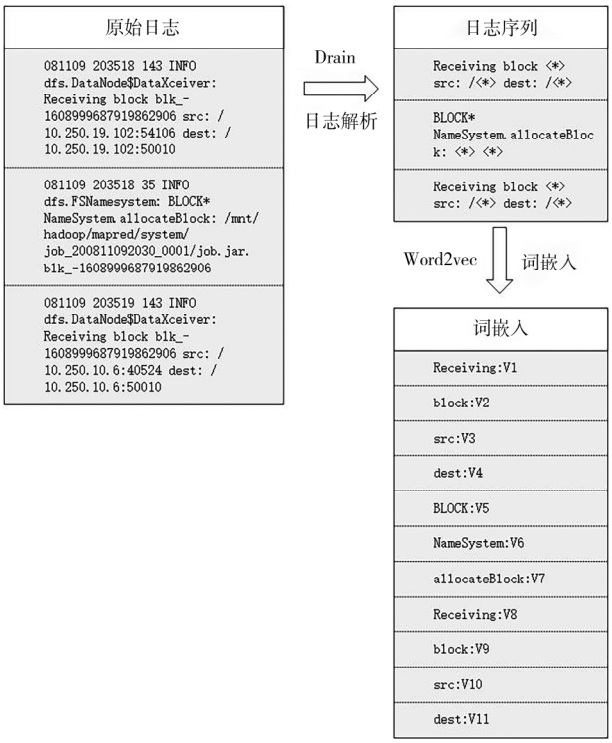

图 2　日志数据表征构建过程

Fig.2　Logs representation construction process

3.3　基于匹配平均算法的联邦学习

LogFTL 采用基于匹配平均的联邦学习算法实现全局模型的聚合[7].基于匹配平均的联邦学习算法通过对具有相似特征提取标志的隐藏元素进行匹配和均值计算,以分层方式构建共享全局模型.隐藏元素包括卷积层的通道、长短期记忆神经网络的隐藏状态、完全连接层的神经元等.考虑一个基本的循环神经网络(Recurrent Neural Network, RNN),$h_t=\sigma(h_{t-1}H+x_tW)$,其中 $H\in\mathbb{R}^{L\times L}$ 是隐藏权重到隐藏权重的置换不变量,L 是隐藏单元的数量,W 是隐藏权重的输入.为了说明 RNN 隐藏状态的置换不变性,我们注意到 h_t 的维数应该以与任何 t 相同的方式进行置换,因此,有

$$h_t=\sigma(h_{t-1}\Pi^TH\Pi+x_tW\Pi) \tag{1}$$

其中,Π 是 $L\times L$ 的置换矩阵.为了匹配循环神经网络 RNN,基本的子问题是将两个具有欧几里德相似性的客户端的隐藏权重对齐.这就需要在排列 Π 上对 $||\Pi^TH_j\Pi-H_{j'}||_2^2$ 进行最小化.利用置换不变性,联邦学习全局模型的隐藏权重能够被计算为 $H=\frac{1}{j}\sum_j\Pi_jH_h\Pi_j^T$. LSTM 具有多个单元状态,每个单元状态都具有隐藏到隐藏的单个单元状态和隐藏权重的输入.当计算置换矩阵时,我们将隐藏权重的输入堆叠成 $SD\times L$ 权重矩阵(S 是单元状态的数量;D 是输入维度;L 是隐藏状态的数量),然后通过式(2)平均所有权重.

$$\min_{\{\pi_{li}\}}\sum_{i=1}^{L}\sum_{j,l}min\,\pi_{li}^{j}c(w_{jl},\theta_i)\ s.t.\ \sum_i\pi_{li}^{j}=1\ \forall j,l;\sum_l\pi_{li}^{j}=1\ \forall i,j \tag{2}$$

其中 w_{jl} 表示在客户端 j 上学习的第 l 个神经元;θ_i 表示全局模型的第 i 个神经元;$c(\cdot,\cdot)$ 是两个神经元之间的平方欧式距离相似函数;π_{li}^{j} 是客户端 j 提供的置换矩阵.

用户端在本地训练时根据损失函数计算梯度并更新本地模型参数权重.最后服务器端聚合用户端上传的模型参数.我们使用 f_S 表示我们将要训练的全局模型,训练目标可以表示为:

$$arg\ \min_{\Theta}L=\sum_{i=1}^{n}l(y_i,f_S(x_i)) \tag{3}$$

其中,$l(\cdot,\cdot)$ 表示全局网络的损失;例如分类任务的交叉熵损失.$\{x_i,y_i\}_{i=1}^{n}$ 是所有样本和他们的标签.Θ 表示所有需要学习的参数,比如权重和偏差.在获得初始的全局模型后,它将被分发给所有用户,在我们的框架中,用户数据的分享和传输是

被禁止的. 在所有的用户模型训练完成后，用户模型的参数被上传到服务器上进行聚合对于参与者，其学习目标可以表示为：

$$arg\ \min_{\Theta_u} L_u = \sum_{i=1}^{n^u} l(y_i^u, f_u(x_i^u)) \tag{4}$$

3.4　基于长短期记忆网络的迁移学习

联邦学习能够打消参与者共享日志数据时发生数据隐私泄露问题的顾虑从而解决“数据孤岛”的问题，进而帮助我们使用不同来源不同类型的日志数据对模型进行训练.通过联邦训练得到的全局模型是对所有用户数据信息的聚合.

全局模型能够学习到更多不同应用系统类型的日志数据知识，但是不同类型应用系统的日志数据其结构和语法存在差异，反而可能导致本地模型对本地日志数据类型检测效果变差，因此需要通过迁移学习对本地模型进行重新训练以获得一个能够保证本地异常行为检测效果的本地模型. 为此我们设计了一个专门用于 LogFTL 框架的 LSTM 模型，并基于该模型实现迁移学习. LSTM 网络在日志异常检测方法中经常被使用. 在 LogFTL 框架的迁移学习过程中，客户端在接收到服务端传输回来的模型后使用本地日志序列对 LSTM 网络的全连接层进行重训练. 迁移学习的具体过程如图 3 所示.

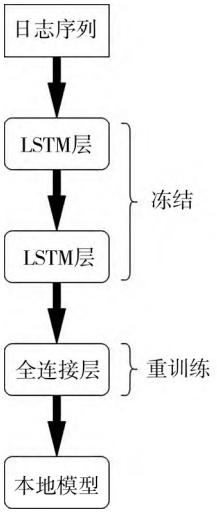

图 3　基于 LSTM 模型的迁移学习过程

Fig.3　Transfer learning process based on LSTM

4　实　验

4.1　基本设置

4.1.1　**数据集**　为了评估 LogFTL 的性能，我们使用了两个公共数据集进行了实验：(1)从 Hadoop 应用中收集的 Hadoop 应用数据集[14]；(2)从 Hadoop 文件系统中收集的 HDFS 数据集[6]. 我们对上述两个数据集进行数据预处理及日志表征构建. 使用 Drain 对日志数据进行模板解析并将其转化为向量序列，每个序列代表一个样本. 转化后 Hadoop 应用数据集共有 193 000 条数据，其中正常样本数 120 000 条，异常样本数 73 000 条；HDFS 数据集共有 3 833 000 条数据，其中正常样本数 3 725 000 条，异常样本数 108 000 条. 详细信息如表 1.

表 1　数据集概述

Tab.1　Datasets overview

数据集	应用系统类型	正常样本数	异常样本数
Hadoop	PageRank	120 000	73 000
HDFS	HDFS	3 725 000	108 000

4.1.2　**实验设置**　本文基于联邦迁移学习方法对应用系统日志进行异常检测.为模拟联邦学习场景，将 Hadoop 应用数据集根据 50%，30%，10%的比例随机划分为 3 部分，作为 3 个客户端的本地数据，以体现实际环境中各参与者日志样本量不同的情况，对应的客户端命名为 C_1、C_2、C_3，并将这些数据中的 70% 用做训练集，30% 用做测试集. 将 HDFS 数据集以相同比例划分，对应的客户端命名为 C_4，C_5，C_6. 两个数据集剩下的 10%数据混合后用来验证模型是否能够检测不同类型应用系统日志数据的异常样本，命名为 Server.最终获得对应 6 个客户端的子数据集和对应混合类型应用系统日志的子数据集.为方便统计和计算，删除了极少量的样本，其数量占总样本数不到 0.1%.实验数据集的具体划分情况如表 2 所示. 在 LogFTL 框架中，我们将 LSTM 网络设定为 2 层 128 个记忆单元的 LSTM 层和 1 层 192 个记忆单元的全连接层.

表 2　数据集划分情况

Tab.2　Datasets partition

客户端	数据来源	正常样本数	异常样本数
C_1	Hadoop	60 000	36 500
C_2	Hadoop	36 000	21 900
C_3	Hadoop	12 000	7 300
C_4	HDFS	1 862 500	54 000
C_5	HDFS	1 117 500	32 400
C_6	HDFS	372 500	10 800
Server	混合	384 500	136 900

4.2　实验评估指标

为评估检测效果,本文使用精度(*Precision*)、召回率(*Recall*)、F_1 分数(F_1-*Score*)和准确率(*Accuracy*)作为评估指标,其数学定义如下.

(1) 精度:正确判定的恶意样本数量与所有判定为恶意样本的数量之比.

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

(2) 召回率:正确判定的恶意样本数量与全部真正的恶意样本数量之比.

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

(3) F_1 分数:精度和召回率的加权调和平均值.

$$F_1\text{-}Score=\frac{2*Precision*Recall}{Precision+Recall}$$

(4) 准确率:正确判定良性和恶意样本的数量与所有样本数量之比,简写为 *Acc*.

$$Acc=\frac{TP+TN}{TP+TN+FP+FN}$$

4.3　联邦迁移学习框架效果评估实验

本文提出的基于联邦迁移学习的日志异常检测方法其核心目标是为了在数据独立存在于不同位置且因为隐私问题无法直接共享的前提下,利用联邦学习以隐私安全且模型参数加密的方式聚合出一个学习到不同客户端不同应用系统类型日志数据的全局模型,再将全局模型分发给参与者利用迁移学习强化其针对本地日志数据的检测能力.表 3 表明,所有客户端在经过联邦迁移学习之后在本地数据集的检测结果.

表 3　不同客户端模型本地检测结果

Tab.3　The detection results of different client models

客户端	准确率/%	精度/%	召回率/%	F_1 分数/%
C_1	90.72	97.29	88.85	92.88
C_2	88.60	90.79	90.87	90.83
C_3	86.42	81.30	96.29	88.16
C_4	92.80	93.12	99.44	96.17
C_5	91.57	91.78	99.51	95.49
C_6	90.26	90.49	99.59	94.82

上述实验表明,本文提出的联邦迁移学习算法,在检测客户端本地应用系统类型的日志数据的异常上有较好的表现,平均达到了 90.06% 的准确率.表 4 表明,所有客户端在经过联邦迁移学习之后在 Server 数据集的检测结果.

表 4　不同客户端模型在 Server 数据集上的评估结果

Tab.4　The detection results of different client models on Server dataset

客户端	准确率/%	精度/%	召回率/%	F_1 分数/%
C_1	88.51	88.51	95.59	91.91
C_2	88.62	89.56	94.72	92.07
C_3	86.77	85.66	95.97	90.52
C_4	89.35	89.60	95.68	92.54
C_5	86.74	85.79	95.79	90.52
C_6	89.21	88.45	96.63	92.36

通过对不同类型应用系统日志混合数据集 Server 的检测效果实验,也证明了在没有与其他客户端共享日志的条件下,联邦迁移学习让本地模型拥有检测其他类型应用系统日志异常数据异常的能力,平均达到了 88.2% 的准确率.

为了进一步评估 LogFTL 检测混合类型应用系统日志的效果,我们将其与 4 种有监督的基于日志的异常检测方法,包括线性回归、SVM、决策树和 CNN 模型以及 4 种无监督的方法,包括 PCA、LogCluster、LogAnomaly、DeepLog 进行对比.图 4 记录了使用不同方法在 Server 数据集上的测试结果.实验结果表明,我们的方法通过聚合位于不同客户端的数据,能够比传统的有监督学习方法和无监督学习方法学习到更多异常模式.在本地数据样本不足的情况下,本文方法的准确率要高于其他检测方法,达到了 88.25%,证明了 LogFTL 能够在不影响隐私安全的前提下,将处于不同客户端的数据以联邦学习的方式进行聚合.

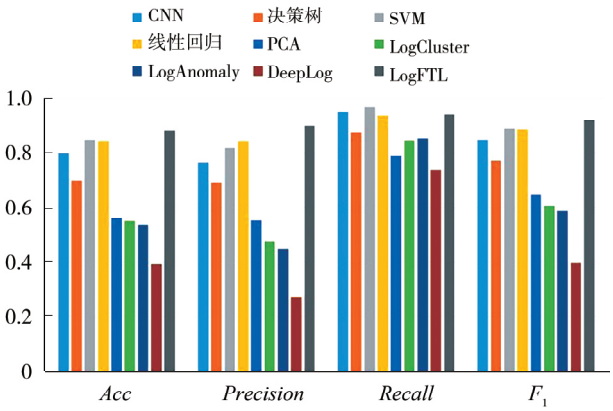

图 4　不同方法在 Server 数据集上的检测结果

Fig. 4　The detection results of different methods on Server dataset

4.4　迁移学习效果评估实验

为了证明 LogFTL 框架迁移学习的有效性，我们比较了使用了迁移学习和没有迁移学习的 LogFTL 的效果，其结果如表 5 所示. 显然，使用迁移学习确实提高了 LogFTL 框架检测异常的准确率.

表 5　迁移学习效果评估实验结果

Tab.5　The evaluation results of transfer learning

客户端名称	LogFTL/%	LogFTL-NoTransfer/%
C_1	90.72	85.67
C_2	88.60	86.05
C_3	86.42	82.47
C_4	92.80	87.41
C_5	91.57	83.38
C_6	90.26	86.91

5　结　论

本文分析了现有日志数据异常检测中存在的隐私泄露风险，提出了一种基于联邦迁移学习的应用系统日志的异常检测方法. 该方法基于长短期记忆网络构建基础模型，在保证参与者数据隐私的前提下，通过联邦学习框架解决了日志数据的“数据孤岛”问题和不同应用系统类型日志的结构和语法差异问题，并采用迁移学习的方法解决联邦学习全局模型本地化后检测效果降低的情况. 我们在两种数据集下进行了多场景的实验，证明了 LogFTL 方法在联邦学习的场景下，不仅具有最好的模型性能，还能够保证隐私和数据的安全.

参考文献：

[1] Chen J, Zhang S, He X, *et al*. How incidental are the incidents ?: characterizing and prioritizing incidents for largescale online service systems [C]// Proceedings of the 35 th IEEE/ACM International Conference on Automated Soft-ware Engineering. New York: ACM, 2020: 373.

[2] 张颖君，刘尚奇，杨牧，等. 基于日志的异常检测技术综述 [J]. 网络与信息安全学报，2020，6：1.

[3] 黄纬，黄晓华，张源，等. 基于 Git 日志的即时软件质量分析框架[J].吉林大学学报：理学版，2022，60：135.

[4] Xu W, Huang L, Fox A,*et al*. Detecting large-scale system problems by mining console logs [C]//Proceedings of the ACM SIGOPS 22nd Symposium on Operating systems Principles. New York: ACM, 2009: 117.

[5] Du M, Li F, Zheng G, *et al*. Deeplog: Anomaly detection and diagnosis from system logs through deep learning [C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2017: 1285.

[6] Karun A K, Chitharanjan K. A review on hadoop——HDFS infrastructure extensions [C]// Proceedings of the 2013 IEEE Conference on Information & Communication Technologies. Thuckalay: IEEE, 2013: 132.

[7] Chen R, Zhang S, Li D, *et al*. Logtransfer: Cross-system log anomaly detection for software systems with transfer learning [C]//Proceedings of the 2020 IEEE 31st Internation-al Symposium on Software Reliability Engineering. Coimbra: IEEE, 2020: 37.

[8] Wang H, Yurochkin M, Sun Y, *et al*. Federated learning with matched averaging [EB/OL]. [2022-02-15]. https://arxiv.org/pdf/2002.06440.pdf.

[9] He S, Zhu J, He P, *et al*. Experience report: System log analysis for anomaly detection [C]//2016 IEEE 27th International Symposium on Software Reliability Engineering. Ottawa: IEEE, 2016: 207.

[10] Zhu J, He S, Liu J, *et al*. Tools and benchmarks for autom-ated log parsing [C]//Proceedings of the 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice. Montreal: IEEE, 2019: 121.

[11] He P, Zhu J, Zheng Z, *et al*. Drain: an online log parsing approach with fixed depth tree [C]//Proceedings of the 2017 IEEE International Conference on Web Services.Montreal: IEEE, 2017: 33.

[12] 闫力，夏伟.基于机器学习的日志异常检测综述[J]. 计算机系统应用，2022，31：57.

[13] Chen M, Zheng A X, Lloyd J, *et al*. Failure diagnosis using decision trees [C]//Proceedings of the International Conference on Autonomic Computing. New York: IEEE, 2004: 36.

[14] Liang Y, Zhang Y, Xiong H, *et al*. Failure prediction in ib-m bluegene/l event logs [C]//Proceedings of the 7 th IEEE International Conference on Data Mining. Omaha: IEEE, 2007: 583.

[15] Lin Q, Zhang H, Lou J G, *et al*. Log clustering based problem identification for online service systems [C]//Proceedings of the 2016 IEEE/ACM 38th International Conference on Software Engineering Companion. Austin Texas: ACM, 2016: 102.

[16] Zhang X, Xu Y, Lin Q, *et al*. Robust log-based anomaly detection on unstable log data [C]//Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. Tallinn Estonia: ACM, 2019: 807.

[17] Meng W, Liu Y, Zhu Y, *et al*. LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs [C]//Proceedings of the 2019 International Joint Conferences on Artificial Intelligence.Macao: IJCAI, 2019: 4739.

[18] Konecny J, McMahan H B, Ramage D, *et al*. Federated optimization: Distributed machine learning for on-device intelligence [EB/OL].[2016-08-08]. https://arxiv.org/pdf/1610.02527.pdf.

[19] Guo Y, Wu Y, Zhu Y, *et al*. Anomaly detection using distributed log data: a lightweight federated learning approach [C]//Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN). Shenzhen: IEEE, 2021: 1.

[20] Xu W, Huang L, Fox A, *et al*. Online system problem detection by mining patterns of console logs [C]//Proceedings of the 9 th IEEE International Conference on Data Mining. [S.l.]: IEEE, 2009: 588.

[21] Mikolov T, Chen K, Corrado G, *et al*. Efficient estimation of word representations in vector space [EB/OL]. [2013-06-16]. https://arxiv. org/pdf/1301.3781.pdf.

[22] Wang J, Tang Y, He S, *et al*. LogEvent2vec: LogEvent-to-vector based anomaly detection for large-scale logs in internet of things [J]. Sensors, 2020, 20: 2451.

引用本文格式:

中　文: 曾闽川, 方勇, 许益家. 基于联邦迁移学习的应用系统日志异常检测研究[J]. 四川大学学报: 自然科学版, 2023, 60: 033002.

英　文: Zeng M C, Fang Y, Xu Y J. Research on application system log anomaly detection based on federated transfer learning [J]. J Sichuan Univ: Nat Sci Ed, 2023, 60: 033002.