doi:10.3969/j.issn.1672-1292.2024.04.002

基于 RoBERTa 和超球体空间的日志异常检测研究

李小鹏[1,2],尹传环[1,2],钞　萌[3]

(1.北京交通大学计算机科学与技术学院,北京 100044)
(2.交通数据分析与挖掘北京市重点实验室,北京 100044)
(3.中国人寿保险股份有限公司上海数据中心,上海 201201)

[**摘要**]　通过监控和分析大量日志数据,日志异常检测能够及时识别入侵攻击、恶意操作等异常行为,是现代系统管理人员的一项关键工具. 针对标注数据稀少的问题,提出基于 RoBERTa 和超球体空间的无监督日志异常检测算法. 首先,为充分学习日志文本的语义特征,提出多层次语义提取网络,有效从多个层面学习日志的上下文信息. 先使用日志语料库对稳健优化的 BERT 预训练方法(robustly optimized BERT pretraining approach,RoBERTa)进行预训练,再使用 RoBERTa 和 Transformer 编码器分别在词语层面和句子层面挖掘日志条目的语义特征. 其次,为增加类差异和挖掘日志的正常模式,在特征空间引入超球体损失. 通过对模型不断优化,在仅使用正常样本进行训练的前提下,正常样本的特征表示能够聚集于超球体空间的中心,而异常样本则远离该中心,最终达到分离异常样本的目的. 最后,该模型在 HDFS 日志数据集和 BGL 日志数据集上分别取得了 0.94 和 0.93 的 $F1$ 分数,验证了该模型的有效性.

[**关键词**]　日志异常检测,稳健优化的 BERT 预训练方法,变换器,超球体空间
[**中图分类号**]TP391　[**文献标志码**]A　[**文章编号**]1672-1292(2024)04-0017-11

Study on Log Anomaly Detection Based on RoBERTa and Hypersphere Space

Li Xiaopeng[1,2],Yin ChuanHuan[1,2],Chao Meng[3]

(1.School of Computer Science and Technology,Beijing Jiaotong University,Beijing 100044,China)
(2.Beijing Key Lab of Traffic Data Analysis and Mining,Beijing 100044,China)
(3.China Life Insurance Company Shanghai Data Center,Shanghai 201201,China)

Abstract:By monitoring and analyzing large volumes of log data,log anomaly detection can promptly identify abnormal behaviors such as intrusions and malicious operations,making it a critical tool for modern system administrators. To address the issue of limited labeled data,this paper proposes an unsupervised log anomaly detection algorithm based on RoBERTa and hyperspherical space. Firstly,to fully capture the semantic features of log texts,a multi-level semantic extraction network is proposed to effectively learn the contextual information of logs from multiple perspectives. Specifically,the robustly optimized BERT pretraining approach(RoBERTa)is pretrained on a log corpus. And then both RoBERTa and Transformer encoders are used to extract semantic features of log entries at the word and sentence level,respectively. Additionally,to enhance class differentiation and uncover normal patterns in logs,hyperspherical loss is introduced in the feature space. By continuously optimizing the model and training with only normal samples,the feature representations of normal samples converge toward the center of the hyperspherical space,while anomalous samples are pushed away from the center,effectively separating the anomalies. The model achieved $F1$ scores of 0.94 and 0.93 on the HDFS and BGL log datasets,respectively,demonstrating its effectiveness.

Key words:logs anomaly detection,RoBERTa,transformer,hypersphere space

在计算机系统和网络应用中,日志数据是一种重要的信息资源,它被广泛用于系统监控、故障诊断和安全审计等任务. 而随着数字化技术的普及,越来越多的企业和组织依赖计算机系统和网络设备来支持

收稿日期:2024-05-12.
基金项目:国家自然科学基金项目(U23B2062).
通讯作者:尹传环,博士,副教授,研究方向:深度学习、网络安全、异常检测、数据挖掘. E-mail:chyin@ bjtu.edu.cn

业务运营,这导致了大量的日志数据积累. 随着时间的推移,这些日志数据的数量呈指数级增长,推动了市场对去人工化的自动日志异常检测的需求.近年来,研究者们已提出了许多方法. 传统的基于机器学习的检测方法受日志数据不平衡的影响,倾向于将所有系统日志都归类为正常日志,而对异常日志的检测准确率较低. 随着深度学习在自然语言处理(natural language processing,NLP)领域的成功,研究者们开始将日志数据视为文本序列,并利用深度神经网络进行文本分类,以实现日志异常检测[1].

针对日志数据标注困难以及日志文本的语义特征难以获取的问题,本研究采用无监督学习的方法,仅需要使用无标签的正常样本进行模型训练,此外,通过使用 Transformer 架构[2]获取文本的上下文信息,本研究能够提取具有丰富语义信息的嵌入表示. 基于以上设计,本研究提出了一个名为 LogBS 的基于 RoBERTa[3]的无监督异常检测模型. 该方法利用预训练的语言模型提取日志的语义嵌入向量,并通过 Transformer 架构和超球体体积最小化学习日志的潜在的正常模式,最后将不符合正常模式的日志判定为异常日志. 根据在 HDFS 数据集和 BGL 数据集上的实验结果,与其他利用长短期记忆网络(long short-term memory,LSTM)等模型的无监督方法相比实现了更高的检测性能[4-5].

1 相关工作

1.1 基于有监督学习的方法

LogRobust 使用 Drain 作为日志解析器删除日志中的参数信息[6-7],然后通过生成词向量和 TF-IDF[8]来获取日志的语义向量,最后使用基于注意力机制的双向长短期记忆网络(bidirectional long short-term memory,Bi-LSTM)对日志进行分类[9]. HitAnomaly 同样使用 Drain 提取日志模板[10],与 LogRobust 不同的是该方法没有舍弃参数信息,而是使用 Transformer 分别对日志模板和参数信息进行编码,然后使用注意力机制将两种特征结合起来,最后基于该融合特征对日志进行分类. NeuralLog 日志解析器可能会在解析的过程中丢失重要信息,因此使用简单预处理之后的原始日志信息进行分类[11]. 该方法通过 BERT(bidirectional encoder representations from transformers)提取日志的语义特征,然后基于注意力机制生成日志序列的向量表示,最后基于该表示对日志序列进行二分类[12]. Logsy 在模型的训练方法上有所不同[13],它抽出目标数据集中的正常日志数据并将其与不同系统上的异常日志数据混合,形成新的数据集用于训练. 这样的方法增强了模型对于未知日志的泛化效果.

上述方法中 LogRobust 和 HitAnomaly 没有避免日志解析器带来的信息丢失,NeuralLog 无法摆脱有监督学习需要大量异常样本的局限性,而 Logsy 在不同的数据集上的性能表现不一致,表现出较差的泛化能力.

1.2 基于无监督学习的方法

OCSVM 在日志异常检测中被广泛使用,该方法通过非线性变换将日志向量映射到超球体空间,最后识别到一个超平面将正常日志与异常日志分离[14]. LogCluster 通过聚类和频率特征挖掘可以从日志事件中发现频繁出现的异常事件[15]. 上述方法在本质上属于浅层的学习方法,不能充分捕获日志数据内在的语义特征.

DeepLog 是一个基于 LSTM 的深度神经网络模型[16]. 它使用日志解析器将日志分离为日志模板和参数信息两部分. 随后,该模型利用日志模板来检测系统的执行路径异常,同时利用参数信息来检测系统的参数异常. 然而该方法没有利用到日志的语义信息,仍有可改进之处.

LogBERT 是一个基于 BERT 的异常检测模型[17]. 它通过预测正常日志序列中被随机掩盖的日志来识别正常日志序列的模式,另外它以超球体体积最小化的方式进一步使得正常日志序列的表示更加紧凑. 最后将偏离正常模式的日志序列识别为异常. 该方法在计算日志序列表示时只是将日志直接映射到其日志模板的哈希值上,没有使用到日志的语义特征和日志的上下文信息,在未知日志数据上泛化效果较差.

2 多层次语义提取与超球体空间网络模型

本研究所提出的 LogBS 方法的模型框架如图 1 所示,其主要模块包括 3 个部分.

(1)数据预处理. 使用预定义的正则表达式对日志数据进行预处理,以删除日志中的标点符号等无用

信息. 然后通过时间窗口或会话窗口将预处理之后的日志记录划分成日志序列.

(2)日志序列语义表示. 针对预处理之后的日志序列,通过预训练语言模型 RoBERTa 和基于注意力机制的 Transformer 架构多层次提取日志序列的语义特征.

(3)异常检测. 通过将正常样本映射到超球体空间,并持续缩小超球体体积,最终可以学习到正常日志序列的潜在模式.

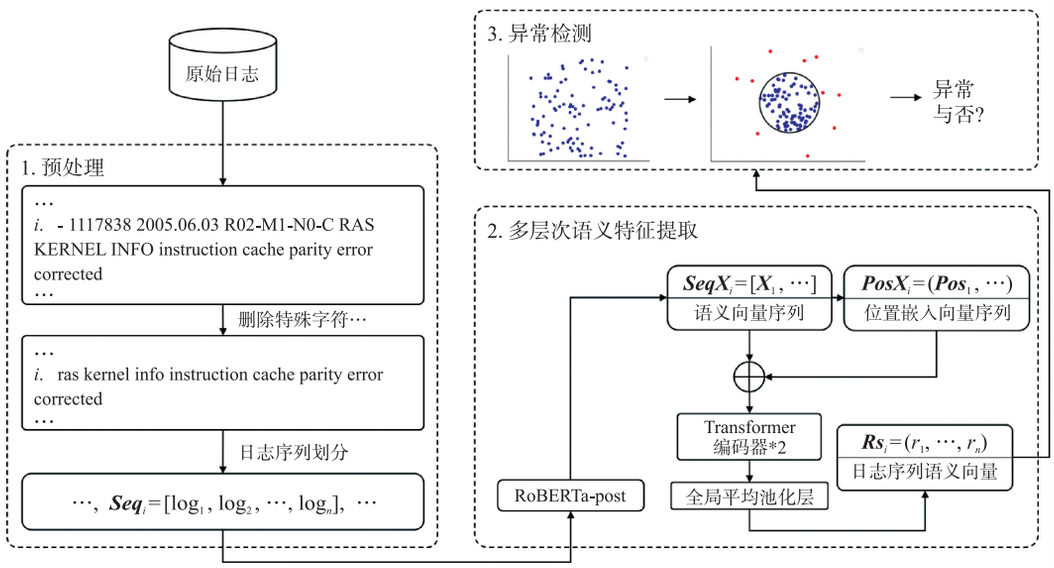

图 1　LogBS 的框架

Fig. 1　The architecture of LogBS

2.1　日志预处理

日志数据属于半结构化数据,包含了大量对异常检测无意义的信息,如系统名称、服务器节点号等,多种方法使用 Drain 等日志解析器对日志进行预处理,从而过滤掉这部分信息[6,10,16]. 然而有研究表明这会引入因语义误解而导致的日志解析错误[11]. 如图 2 所示,日志解析器将日志“database check enable”和“database check interrupt”解析为了同一个模板,而前者表示启用数据库检查,属于正常行为,后者表示数据库检查中断,属于异常行为. 这种类型的错误使得模型很难从日志内容上区分正常日志和异常日志.

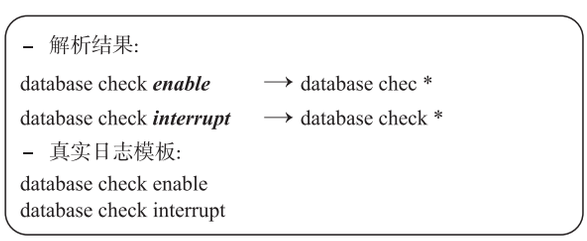

图 2　日志解析错误示例

Fig. 2　Example of log parsing error

为了避免以上错误,LogBS 方法放弃了传统的日志解析器方法,采用了一种新的预处理策略,旨在删除日志中的时间戳等参数信息,保留其他语义文本. 具体而言,该方法首先基于驼峰命名将日志内容中的组合词拆分为子词;然后将大写字母转换为对应的小写字母,并使用常见的分隔符(例如空格、逗号等)将日志内容拆分为多个标记;最后,将带有数字的标记以及所有符号从中删除,并使用空格将剩余的标记拼接为日志消息. 例如,对于原始日志数据“R24-M0-N1-C:J13-U11 RAS KERNEL INFO 162 double hummer alignment exceptions”,经过预处理后的日志内容为“ras kernel info double hummer alignment exceptions”.

通过预处理,LogBS 方法能够更好地捕获日志消息中的关键信息,从而提高模型的检测性能和准确性. 除此之外,还需要将日志划分为序列,因为系统异常可能表现为某个时间点发生的特定事件,或是一系列事件的持续变化. 通过划分日志序列,模型能够捕获到日志的顺序信息和上下文信息,进而更加细致地分析和理解系统行为的演变过程. 这对于日志异常检测来说是有必要的.

通常,划分日志序列的方法包括会话窗口、时间窗口和固定窗口. 会话窗口是基于会话 ID 或其他相关特征进行划分的,这意味着每个日志序列代表了一个完整的会话过程. 时间窗口则按照时间顺序将所有日志划分为窗口,但在相同的时间间隔内可能包含不同数量的日志. 固定窗口方法将日志数据划分为

包含相同数量日志的日志序列.

针对不同数据集,本研究使用不同的划分方法. 对于以日志序列为异常样本的数据集 HDFS,采用会话窗口划分,而对于以单条日志为异常样本的数据集 BGL,则使用固定窗口进行划分.

在获得包含有效语义文本的日志数据后,将这些数据输入到多层次特征提取模块,深入挖掘其语义特征. 然后,在超球体空间中聚集正常类的语义特征,以扩大正常类与异常类的整体特征差异.

2.2　多层次语义特征提取模块

预处理之后的日志数据包含了丰富的语义信息,如日志级别和事件信息等,这些信息是区分正常日志和异常日志的关键. 合理的日志序列表示应当能够准确地捕获到不同序列之间的差异,确保语义相异的日志序列具有相互远离的序列表示. 近年来,RoBERTa 和 Transformer 编码器凭借其卓越的文本表征能力和序列建模能力,在 NLP 领域取得了巨大成功,为各种文本相关任务提供了强有力的解决方案.

RoBERTa 是一个在 NLP 领域被广泛使用的预训练语言模型. 通过在大规模的文本语料上进行预训练,该模型学习了大量的语言知识. 这种预训练使得 RoBERTa 在各种任务中能够更好地理解和表征文本. 其次,RoBERTa 使用 Transformer 架构中的自注意力机制,能够有效地捕获文本中词语之间的依赖关系,从而提高对文本语义的理解能力. 另外,RoBERTa 是一个深层次的模型,由多个 Transformer 编码器层组成. 每个编码器层都能够对文本进行多层次的抽象表示,从字符级别到句子级别进行语义特征的提取,从而使得 RoBERTa 能够获取丰富的文本特征.

Transformer 编码器是 Transformer 架构的核心组件之一,常被用于将输入序列编码成高质量的表示. 它由多个相同的层堆叠而成,每个层都包含两个子层:自注意力机制子层和全连接前馈神经网络子层. 自注意力机制子层负责捕捉输入序列中各个位置之间的依赖关系,使得每个位置都能够同时考虑到整个输入序列的信息. 全连接前馈神经网络子层则通过多层神经网络对每个位置的特征进行非线性转换和映射,增强模型的表征能力. 这种精确的文本表征能力使得 Transformer 编码器可以被用于各种序列学习任务,如语言建模、文本分类、机器翻译等. 此外,由于 Transformer 编码器采用了自注意力机制和残差连接等关键技术,使得它能够处理长序列和捕捉长程依赖关系,因此在 NLP 领域取得了广泛的成功应用[18].

综上所述,本文通过将 RoBERTa 和 Transformer 编码器结合,分别从日志和日志序列的层次关注文本中的上下文信息,提取具有良好语义表征的嵌入向量. 该提取过程如图 3 所示.

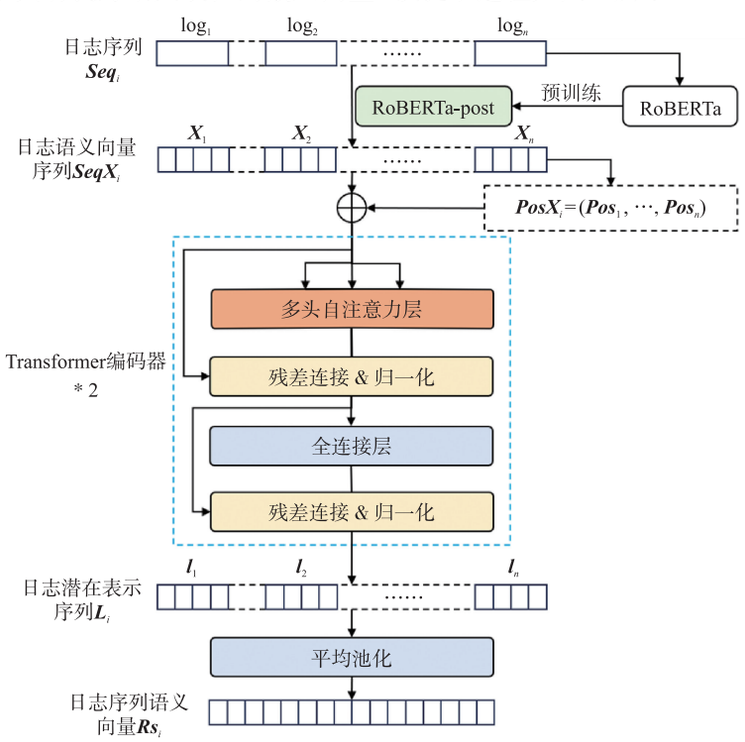

图 3　多层次语义特征提取模块

Fig. 3　Multi-level semantic feature extraction module

2.2.1 RoBERTa 预训练阶段

尽管 RoBERTa 已经在大规模语料上进行了预训练,但由于其使用的语料库与日志文本在结构、术语等方面存在明显差异,因此它在对日志的理解方面可能存在一定的偏差. 为了解决这一问题,本方法利用日志数据集对 RoBERTa 进行了无监督的掩码语言模型(masked language model,MLM)训练,具体来说,对于每一条输入的日志,随机选择其中 15%的单词进行掩盖,而对于每一个被选择的词,有 80%的概率将其替换为特殊的“[Mask]”标记,10%的概率将其替换为语料库中随机选择的其他词,另有 10%的概率保持不变,然后让模型预测这些被掩蔽的单词,从而学习上下文信息. 这种训练方法使得 RoBERTa 能够更好地理解日志文本的特点,从而提高其在日志异常检测任务中的泛化能力. 本文将经过预训练的 RoBERTa 模型保存为模型实例 RoBERTa-post,以便后续用于生成日志语义向量.

2.2.2 日志语义挖掘

给定日志序列 $\boldsymbol{Seq}_i=[\log_1,\log_2,\cdots,\log_n]$,对于序列 $\boldsymbol{Seq}_i$ 中的任意一条日志 $\log_j$,RoBERTa-*post* 会在其首尾分别添加“<*s*>”和“</*s*>”标记,然后将其截断或填充到统一长度,最后对处理后的日志 $\log_j=[<s>,tok_1,tok_2,\cdots,pad,</s>]$编码. 如图 4 所示,给定包含 V 个词汇的语料库以及任意包含 n 个文本标记的日志条目,首先,RoBERTa-post 使用嵌入矩阵 $\boldsymbol{E}_{\text{token}}\subset\mathbf{R}^{V*d}$将文本标记转化为初始嵌入向量 $\boldsymbol{E}\in\mathbf{R}^{n*d}$. 然后,使用三角函数编码初始嵌入向量的位置信息,生成位置嵌入 $\boldsymbol{T}\in\mathbf{R}^{n*d}$并将其与 $\boldsymbol{E}$ 相加作为编码层的输入. 其次,使用包含 12 层 Transformer 层的编码器处理向量表示,以注意力机制关注每个单词的上下文,计算其上下文表示,得到日志的语义嵌入向量 $\boldsymbol{S}\in\mathbf{R}^{n*d}$.

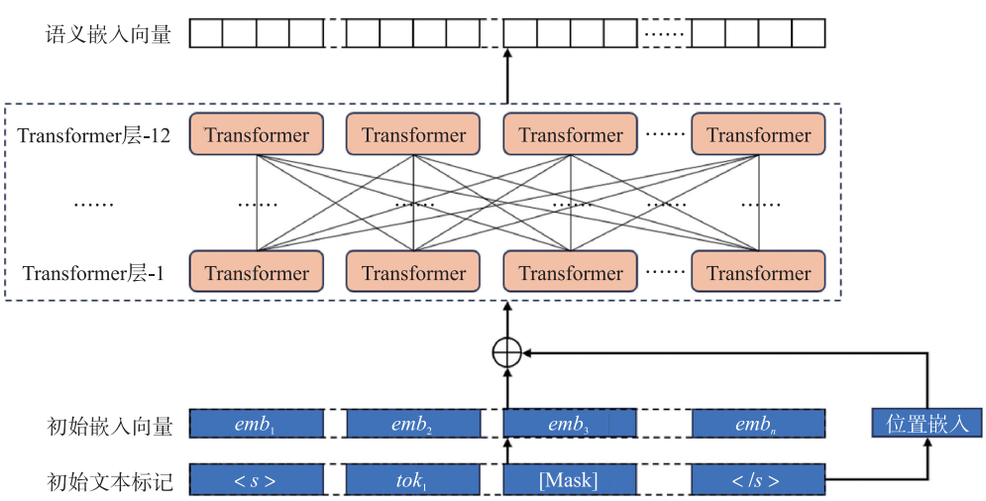

图 4 日志语义挖掘流程图

Fig. 4 Flow chart of log semantic mining

上述过程中位置嵌入向量 $\boldsymbol{Pos}_{(i,j)}$ 为

$$\boldsymbol{Pos}_{(i,j)}=\begin{cases}\sin(w_j\cdot i), & j=2k,\\ \cos(w_j\cdot i), & j=2k+1.\end{cases}\tag{1}$$

其中,i 是元素在对应序列中的位置.

$$w_j=\frac{1}{10\ 000^{\frac{2\cdot j}{d}}},\quad j=0,1,\cdots,\frac{d}{2}-1.\tag{2}$$

其中,d 是待编码向量的嵌入向量维度数,j 是位置编码 $\boldsymbol{Pos}_i$ 中的元素下标.

2.2.3 日志序列语义挖掘

本研究使用基于注意力机制的 Transformer 架构对日志语义向量序列进行合并,主要过程可分为位置编码和语义编码两个阶段.

(1)位置编码. 除了语义信息之外,日志之间的顺序也是重要特征之一,不同的顺序意味着不同的系统状态变化过程. 而上文中的日志语义向量并不包含顺序信息,因此,LogBS 通过三角函数计算与日志语义向量 $\boldsymbol{X}_i$ 维度相同的位置编码 $\boldsymbol{Pos}_i$[2]. 其计算形式如式(1)所示.

通过将日志语义向量 $\boldsymbol{X}_i$ 与对应的 $\boldsymbol{Pos}_i$ 的和输入 Transformer 编码器,模型能够捕获到日志在日志序列中的位置信息,进而能够分辨日志组成相同,但顺序不同的日志序列,这对于检测日志的顺序异常是不可或缺的.

(2)Transformer 编码器. 如图 3 所示,Transformer 编码器主要包含一个多头自注意力层和一个前馈神经网络层,层间的数据处理包括 Dropout 正则化、残差连接和层归一化.

多头自注意力机制是自注意力机制的一种扩展模式,它通过引入多个注意力头,同时关注日志文本序列中的多个位置,以多种注意力机制挖掘日志在日志序列中的上下文信息. 该机制的实现步骤如下:

① 通过 3 个不同的线性变换将输入序列 $\boldsymbol{Seq}_X$ 变换为查询、键和值矩阵,分别用 $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$ 表示,其计算过程为

$$\boldsymbol{Q}=\boldsymbol{Seq}_X\boldsymbol{W}_Q,\boldsymbol{K}=\boldsymbol{Seq}_X\boldsymbol{W}_K,\boldsymbol{V}=\boldsymbol{Seq}_X\boldsymbol{W}_V. \tag{3}$$

② 单独计算每个注意力头的注意力得分. 其中,第 i 个注意力头的注意力分数 head_i 为

$$\mathrm{head}_i=\mathrm{Attention}(\boldsymbol{QW}_{Q_i},\boldsymbol{KW}_{K_i},\boldsymbol{VW}_{V_i})=\mathrm{softmax}\left(\frac{\boldsymbol{QW}_{Q_i}(\boldsymbol{KW}_{K_i})^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{VW}_{V_i}. \tag{4}$$

其中,Attention( )代表注意力得分的计算过程,softmax( )用于将给定的向量转换为一个概率分布,首先,softmax( )会对每个输入数值取指数,然后将每个指数值除以所有指数值的总和. $\boldsymbol{W}_{Q_i}$,$\boldsymbol{W}_{K_i}$和 $\boldsymbol{W}_{V_i}$分别是第 i 个头的查询、键和值的权重矩阵,d_k 是键的维度.

③ 将所有头的输出拼接起来并使用线性变换将结果转换到与输入序列 $\boldsymbol{Seq}_X$ 相同的维度,该计算过程为

$$\mathrm{MultiHead}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\mathrm{Concat}(\mathrm{head}_1,\mathrm{head}_2,\cdots,\mathrm{head}_n)\boldsymbol{W}_O. \tag{5}$$

其中,Concat( )表示将多个张量沿 1 维进行拼接,MultiHead( )代表该计算过程,$\boldsymbol{W}_O$ 是 1 个线性变换矩阵.

④ 使用残差连接和层归一化处理多头自注意力层的输出,缓解梯度消失和梯度爆炸的问题并稳定训练过程,

$$\begin{aligned}&\mathrm{LayerNorm}(x)=\frac{x-\mu}{\sigma+\varepsilon}\gamma+\beta,\\&\mathrm{Output}(x)=\mathrm{LayerNorm}(\boldsymbol{Seq}_X+\mathrm{MultiHead}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})).\end{aligned} \tag{6}$$

其中,LayerNorm( )代表层归一化处理,Output( )是多头自注意力层的输出经处理后的输出,μ 是输入 x 的均值,σ 是输入 x 的标准差,ε 是防止除零的 1 个小常数,γ 和 β 是可学习的缩放和偏移参数.

在多头自注意力层之后,本研究使用一个由两层全连接层和 ReLU 激活函数组成的前馈神经网络对特征进行进一步变换和组合,其计算过程为

$$\mathrm{Output}(x)=\mathrm{Mean}(\mathrm{Dropout}(\mathrm{ReLU}(\boldsymbol{W}^{\mathrm{T}}\mathrm{Output}+b))). \tag{7}$$

其中,Mean( )代表平均池化,Dropout( )代表随机选择网络中的某些神经元,并将它们的输出设为 0,$\boldsymbol{W}$ 和 b 分别是神经网络的权重矩阵及其偏置. 该过程有助于模型捕捉更复杂的特征和模式,增强模型的表达能力. 最后,通过 Dropout 和平均池化得到日志序列的语义表示.

2.3　异常检测

按照上述步骤,LogBS 充分地挖掘了日志序列中的语义特征,得到的日志序列语义向量能够反映日志序列之间的语义差异. 进一步地,我们期望正常日志序列和异常日志序列的语义向量能够尽可能地差异化. 这种差异化的语义向量表示对于日志异常检测至关重要,因为它能够帮助模型更准确地区分正常行为和异常行为,从而更有效地发现潜在的异常模式和问题,提高日志异常检测模型的性能和可靠性. 受单分类支持向量机(one-class support vector machine,OCSVM)[19]的启发,本方法将日志序列的语义向量映射到超球体空间中,并通过语义向量到超球体中心的均方误差训练语义特征提取器.

训练损失为

$$\mathcal{L}_s=\frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{Rs}_i-c\|^2+\frac{\lambda}{2}\sum_i\|\theta_i\|_F^2, \tag{8}$$

$$c=\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{Rs}_i. \tag{9}$$

其中,n 是正常日志序列的个数,$\boldsymbol{Rs}_i$ 是第 i 个正常日志序列的语义向量,$\mathcal{L}_s$ 的第二项是 L2 正则化项,超球体的中心 c 是所有正常日志序列语义向量的平均值.

经过多轮迭代训练,正常日志序列的语义向量会逐渐靠拢中心,而异常日志序列的语义向量会逐渐远离中心. 最终,通过比较日志序列向量到中心的距离是否超过预先设定的阈值 γ,可判断日志序列是否异常.

为了确定阈值 γ,本方法从目标数据集中抽取了少量的正常日志和异常日志作为验证集. 在验证集中,将所有正常数据的特征平均值作为 γ 的下界 $\gamma_{\min}$,所有异常数据的特征平均值作为 γ 的上界 $\gamma_{\max}$. 然后以 AUC 为标准在区间[$\gamma_{\min}$,$\gamma_{\max}$]内查找最优阈值,使得模型在验证集上取得最高的 AUC. 最终将最优阈值 γ_{best}用于测试集的测试.

3 实验验证与结果分析

LogBS 对上文提出的日志异常检测方法进行实验评估,并对结果进行详细分析.

3.1 数据集介绍

本文在以下两个大规模的公共日志数据集上评估了 LogBS 方法的性能.

(1)数据集 HDFS[4]

数据集 HDFS 源自基准工作负载下的分布式系统,其中的数据根据自定义规则被手动标记为正常或异常. 该数据集共包含 1 175 629 条日志消息,涵盖 49 个日志模板以及 53 种日志异常类型. 此外,以“block id”为标志,该数据集中的日志被组织为了 558 223 个正常日志会话和 16 838 个异常日志会话. 相比于互相独立的单条日志,日志会话代表了完整的系统执行路径,有利于模型学习完整的上下文信息.

(2)数据集 BGL[4]

数据集 BGL 来源于 Lawrence Livermore 国家实验室的超级计算机系统. 该数据集同样被手动标注了标签,其中包含 348 460 条异常日志和 4 399 503 条正常日志. 与 HDFS 数据集不同的是,该数据集以单条日志作为标注的基本单位,无法追踪系统的执行路径. 因此,在日志序列划分时通常基于时间戳对该数据集进行滑动窗口或固定窗口划分. 若划分后的日志序列中包含异常日志则将该序列视为异常日志序列.

本文将数据集 BGL 中日志消息长度小于 5 的日志删除,然后进行窗口大小为 20,步长为 4 的固定窗口划分,此外,对数据集 HDFS 进行会话窗口划分. 划分后的数据集的分布如表 1 所示.

表 1 数据集的分布

Table 1 The distribution of dataset

数据集	类型	序列数量
BGL	正常	270 804
	异常	31 143
HDFS	正常	558 223
	异常	16 838

3.2 评估方法

本研究使用 3 个评价指标来衡量日志异常检测方法的性能:精确率 P、召回率 R 以及分数 $F1$.

$$P=\frac{TP}{TP+FP}, \tag{10}$$

$$R=\frac{TP}{TP+FN}, \tag{11}$$

$$F1=\frac{2\times(P\times R)}{P+R}. \tag{12}$$

式中,TP 表示真正例,即模型正确地预测出的异常日志的数量. FP 表示假正例,即模型将正常日志错误地预测为异常日志的数量. FN 表示假负例,即模型漏检的异常日志的数量.

基于以上定义,精确率 P 代表模型预测出的异常日志序列中真实异常日志序列所占的比例,召回率 R 代表异常日志序列中被模型正确识别的比例,而 $F1$ 分数则综合考虑了这两个指标,代表了模型的整体性能.

3.3 实验设置

(1)实验环境介绍

所有实验均在固定的实验平台上运行,其环境参数如表 2 所示.

(2)数据集配置

对于两个数据集,在预训练阶段,随机抽取其中 100 000 条日志用于 RoBERTa 的 MLM 训练. 在训练阶段,分别取其中 70%的正常日志序列作为训练集,取 9 000 条正常日志序列和 3 000 条异常日志序列作为验证集,其余的数据都被用作测试集.

(3)模型配置

文中所提出的模型的语义向量维度和位置嵌入向量维度均为 768,Transformer 编码器中多头自注意力层的注意力头为 12 个,隐藏层维度为 1 024,平均池化的操作维度在第 2 维. 文中采用自适应学习率优化算法 Adam(adaptive moment estimation),学习率初始值是 1×10^{-5},权重衰减值为 1×10^{-4},训练轮次设置为 100,早停轮次设置为 10,batchsize 设置为 128.

表 2　实验平台配置详情

Table 2　Configuration details of experimental platform

配置项	配置详情
操作系统	Ubuntu 20.04.6 LTS
CPU 信息	Intel(R)Xeon(R)Platinum 8358P CPU@2.60GHz
GPU 型号	NVIDIA GeForce RTX 3090
Python	3.9.18
Pytorch	1.12.0+cu113
Anaconda	23.7.4

3.4　对比实验

如表 3 所示,本研究在相同数据集下将 LogBS 与 2 类基准方法进行了对比,第一类是有监督方法,包括 LogRobust[6] 和 NeuralLog[11],训练集中的数据带有标签,第二类是无监督方法,包括 PCA[20]、OCSVM[14]、DeepLog[16]、LogAnomaly[21] 和 LogBERT[17],训练集中的数据不带标签. 其中,PCA 是一种主成分分析方法,该方法通过将日志序列的数量特征映射到低维空间来识别异常序列. LogAnomaly 是一个基于 LSTM 的模型,该模型通过 template2vec 技术将日志模板转化为语义向量,进而使用 LSTM 对其进行分类. 此外,使用粗体和下划线标记了有监督方法和无监督方法中最高的 $F1$ 分数,使用粗体标记了第二高的 $F1$ 分数.

表 3　总体性能比较

Table 3　Comparison of overall performance

模型类型	数据集模型	数据集 HDFS			数据集 BGL			平均		
		P	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1
有监督	LogRobust	0.98	1.00	**0.99**	0.62	0.96	**0.75**	0.80	0.98	**0.87**
	NeuralLog	0.96	1.00	**0.98**	0.98	0.98	**0.98**	0.97	0.99	**0.98**
无监督	PCA	0.06	1.00	0.11	0.09	0.98	0.16	0.08	0.99	0.14
	OCSVM	0.03	1.00	0.06	0.01	0.12	0.02	0.02	0.56	0.04
	DeepLog	0.88	0.69	0.77	0.90	0.83	0.86	0.89	0.76	0.82
	LogAnomaly	0.94	0.40	0.56	0.73	0.76	0.74	0.84	0.58	0.65
	LogBERT	0.87	0.78	**0.82**	0.89	0.92	**0.90**	0.88	0.85	**0.86**
	LogBS(ours)	0.95	0.93	**0.94**	0.96	0.90	**0.93**	0.96	0.92	**0.94**

在有监督方法中,LogRobust 和 NeuralLog 的测试数据来自 NeuralLog 的原论文[11]. 在无监督方法中,除了本研究提出的方法 LogBS,其他方法的测试数据均来自于 LogBERT 的原论文[17].

从表 3 可得到 3 点结论:

(1)LogRobust 在 HDFS 数据集上表现优异,达到了 0.99 的 $F1$ 分数,但在 BGL 数据集上表现一般,$F1$ 分数仅达到了 0.75. 与之相比,NeuralLog 在两个数据集上性能较好,均获得了 0.98 的 $F1$ 分数.

这些结果表明,基于 Transformer 架构的 NeuralLog 相较于使用 FastText 和 TF-IDF 的 LogRobust 在挖掘日志的语义信息上具备优势. 然而,考虑到在计算机系统中异常日志样本的收集相对困难,以有监督学习为基础的模型在实际情境下通常呈现出较差的泛化性能.

(2)在基于无监督学习的模型中,传统机器学习方法(如 PCA,OCSVM)在处理复杂、高维数据以及需要捕捉非线性关系的任务上存在一些局限性,在性能上要低于基于深度学习的方法(如 DeepLog,LogAnomaly,LogBERT,LogBS).

PCA 和 OCSVM 在召回率上表现出色,但它们的精确率较低,这表明它们倾向于将大量的正常日志误判为异常日志,从而无法有效区分正常日志和异常日志.

LogBERT 在 HDFS 数据集和 BGL 数据集上分别取得了 0.82 和 0.90 的 $F1$ 分数,表现较为突出. 这表明基于 BERT 的 LogBERT 能够充分地学习日志序列的上下文信息. 然而,由于 LogBERT 直接使用日志模

板的模板 ID 作为一条日志的表示,并没有利用到日志记录本身的语义信息,这使得该模型在整体表现上逊色于 LogBS.

DeepLog 在两个数据集上表现较为均衡,平均 $F1$ 分数达到了 0.82. LogAnomaly 虽然在 HDFS 数据集上具备较高的精确率,但是其召回率相对较低,说明该模型漏判了许多异常日志.

总之,LogBS 取得了最高的平均 $F1$ 分数,充分验证了基于 RoBERTa 和 Transformer 架构的多层次语义特征提取模块的有效性.

(3)日志异常检测领域的实用价值在于将模型部署到实际系统中进行异常检测. 由于日志数据通常是由大量正常日志和少量异常日志构成的,这种数据的极度不平衡对模型提出了严峻挑战. 传统的基于有监督学习的方法往往忽略了这一事实,它们可能在利用大量公共数据进行训练时取得了良好的性能表现,却很难在实际应用中获得理想的效果. 本文提出的 LogBS 方法在训练的过程中只使用正常日志数据和少量的异常日志数据,却仍然具有较好的检测性能,甚至在平均 $F1$ 分数上比有监督模型 LogRobust 高 0.07,这说明了 LogBS 在现实应用中具备较高的实用性.

3.5 消融实验

本文提出的 LogBS 方法利用预训练后的 RoBERTa 和 Transformer 架构学习日志文本的语义特征,并将其应用于日志异常检测,取得了良好的性能. 为验证 RoBERTa 模型在语义提取上的优势以及 RoBERTa 预训练的有效性,设计了消融实验:(1)模型在不同的日志向量表示方法下的检测性能. (2)模型在 RoBERTa 未经预训练和经过预训练的情况下的性能.

(1)为了探究 RoBERTa 模型在日志语义提取上的性能,在不进行预训练的前提下,采用 4 种不同的方法提取日志的向量表示,并对模型的性能进行了比较. 检测结果如表 4 所示.

表 4 通过不同的日志向量表示方法产生的检测结果

Table 4 Detection results generated by different log vector representation methods

方法	数据集 HDFS			数据集 BGL		
	P	*R*	*F*1	*P*	*R*	*F*1
日志模板 ID	0.75	0.69	0.72	0.78	0.72	0.75
Word2Vec	0.86	0.77	0.81	0.75	0.81	0.78
BERT	0.90	0.86	0.88	0.91	0.87	0.89
RoBERTa(ours)	**0.90**	**0.93**	**0.91**	**0.93**	**0.88**	**0.90**

从表 4 中的实验结果可以看出,使用 RoBERTa 进行日志向量表示时,即使未进行预训练,模型仍然取得了最佳的检测效果. 这可能是因为日志模板 ID 只是日志模板的数字编号,不包含日志的语义信息,使得模型的训练目标仅是将一组无意义的数字映射到对应标签,导致鲁棒性较差,也无法泛用至未知的日志数据.

Word2Vec 通过浅层的神经网络架构生成静态词向量,为每个词语分配一个唯一的向量,捕捉了词语的语义信息. 在 HDFS 和 BGL 数据集上,基于 Word2Vec 的模型比基于日志模板 ID 的模型在 $F1$ 分数上分别提高了 0.09 和 0.03. 然而,这种方式无法处理词的多义性,从而忽略了一定的日志上下文信息.

BERT 能够通过词语的上下文动态调整词向量,因此能够区分多义词,相比于 Word2Vec 鲁棒性更好. RoBERTa 在训练规模和架构等方面对 BERT 进行了一系列优化,包括使用更大规模的训练数据、采用动态掩码策略、增加训练批次大小以及进行超参数优化等. 这些改进使其在各种自然语言处理任务上表现更加鲁棒和强大. 基于 RoBERTa 的模型比基于 BERT 的模型在两个数据集上的 $F1$ 分数分别提高了 0.03 和 0.01.

(2)为了观察 RoBERTa 预训练对模型检测性能的影响,在表 5 中比较了模型在 RoBERTa 预训练前后的检测效果,“×”表示未对 RoBERTa 进行预训练,而“√”则与其相反. 结果表明,将 RoBERTa 在日志语料库上进行预训练提升了模型对于日志文本的理解能力,提高了 RoBERTa 提取日志语义特征的效果.

具体来看,预训练后的 RoBERTa 模型在两个数据集上的 $F1$ 分数均提高了 0.03. 这表明预训练帮助模型更有效地学习到日志中的特征模式,使其能够更准确地解析和分类不同类型的日志信息,增强了模型的鲁棒性和泛化能力.

表 5　对 RoBERTa 预训练与否的模型检测性能对比

Table 5　Comparison of model detection performance with or without RoBERTa pre-training

预训练	数据集 HDFS			数据集 BGL			平均		
	P	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1
×	0.90	0.93	**0.91**	0.93	0.88	**0.90**	0.92	0.91	**0.91**
√	0.95	0.93	**0.94**	0.96	0.90	**0.93**	0.96	0.92	**0.94**

此外,这一结果还表明,针对特定领域进行预训练,可以显著提升语言模型在该领域任务中的表现. 通过在相关领域的大规模语料库上进行预训练,模型能够积累更多专业知识,从而在后续任务中展现出更强的理解和处理能力. 这对于日志分析这样的专业领域尤为重要,因为日志信息具有独特的格式和内容特征,通用模型难以直接适用. 预训练使得 RoBERTa 能够更好地适应这些特定特征,从而在实际应用中取得更优异的性能.

4　结论

针对日志数据标注困难以及难以获取高质量的日志语义特征的问题,本文提出了无监督日志异常检测方法 LogBS. 该方法在模型训练阶段仅需无标签的正常日志,在测试阶段仅需较少的标注数据作为验证集以确定分离正常日志序列和异常日志序列的决策边界. 此外,通过使用预训练的 RoBERTa 语言模型学习日志的语义特征,并通过 Transformer 架构以注意力机制挖掘日志条目在日志序列中的上下文信息,该方法能够从多个层次充分挖掘日志文本的语义信息. 最后,采用超球体损失对模型进行优化,学习日志的正常模式,将不符合正常模式的日志判定为异常日志. 通过真实数据集上的实验验证了该方法的有效性和泛化能力.

[参考文献](References)

[1] LE V H,ZHANG H. Log-based anomaly detection with deep learning:How far are we? [J]. arXiv Preprint arXiv:2202.04301,2022.

[2] VASWANI A,SHAZEER N,PARMAR N,et al. Attention is all you need[J]. 31st Conference on Neural Information Processing Systems. Long Beach,CA,USA,2017.

[3] LIU Y,OTT M,GOYAL N,et al. Roberta:A robustly optimized bert pretraining approach[J]. arXiv Preprint arXiv:1907.11692,2019.

[4] ZHU J M,HE S L,HE P J,et al. Loghub:A large collection of system log datasets for AI-drive log analytics[C]//2023 IEEE 34th International Symposium on Software Reliability Engineering. Florence,Italy,2023.

[5] HOCHREITER S,SCHMIDHUBER J. Long short-term memory[J]. Neural Computation,1997,9(8):1735-1780.

[6] 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,2019.

[7] HE P,ZHU J,ZHENG Z,et al. Drain:An online log parsing approach with fixed depth tree[C]//2017 IEEE International Conference on Web Services. Honolulu,HI,USA:IEEE,2017.

[8] SALTON G,BUCKLEY C. Term-weighting approaches in automatic text retrieval[J]. Information Processing & Management,1988,24(5):513-523.

[9] HUANG Z,XU W,YU K. Bidirectional LSTM-CRF models for sequence tagging[J]. arXiv Preprint arXiv:1508.01991,2015.

[10] HUANG S,LIU Y,FUNG C,et al. Hitanomaly:Hierarchical transformers for anomaly detection in system log[J]. IEEE Transactions on Network and Service Management,2020,17(4):2064-2076.

[11] LE V H,ZHANG H. Log-based anomaly detection without log parsing[C]//2021 36th IEEE/ACM International Conference on Automated Software Engineering. Melbourne,Australia:IEEE,2021.

[12] DEVLIN J,CHANG M W,LEE K,et al. BERT:Pre-training of deep bidirectional transformers for language understanding[J]. arXiv Preprint arXiv:1810.04805,2018.

[13] NEDELKOSKI S,BOGATINOVSKI J,ACKER A,et al. Self-attentive classification-based anomaly detection in unstructured

logs[C]//2020 IEEE International Conference on Data Mining. Sorrento,Italy,2020.

[14] WANG Y,WONG J,MINER A. Anomaly intrusion detection using one class SVM[C]//Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop. West Point,NY,USA:IEEE,2004.

[15] VAARANDI R,PIHELGAS M. Logcluster-A data clustering and pattern mining algorithm for event logs[C]//2015 11th International Conference on Network and Service Management. Barcelona,Spain,2015.

[16] DU M,LI F F,ZHENG G N,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. Dallas,Texas,USA,2017.

[17] GUO H X,YUAN S L,WU X T. LogBERT:Log anomaly detection via BERT[C]//2021 International Joint Conference on Neural Networks. Shenzhen,China,2021.

[18] GILLIOZ A,CASAS J,MUGELLINI E,et al. Overview of the transformer-based models for NLP tasks[C]//2020 15th Conference on Computer Science and Information Systems. Sofia,Bulgaria,2020.

[19] SHIN H J,EOM D H,KIM S S. One-class support vector machines—an application in machine fault detection and classification[J]. Computers & Industrial Engineering,2005,48(2):395-408.

[20] 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. Big Sky,Montana,USA,2009.

[21] MENG W B,LIU Y,ZHU Y C,et al. Loganomaly:Unsupervised detection of sequential and quantitative anomalies in unstructured logs[C]//IJCAI. Macau,China,2019.

[责任编辑:陈　庆]