文章编号：1674-2974（2022）04-0078-11 DOI：10.16339/j.cnki.hdxbzkb.2022271

基于混合生成网络的软件系统异常状态评估

杨宏宇[1,2†]，李译[2]，张良[3]
（1. 中国民航大学 安全科学与工程学院，天津 300300；
2. 中国民航大学 计算机科学与技术学院，天津 300300；
3. 亚利桑那大学 信息学院，图森 美国 AZ 85721）

摘 要：针对现有软件系统异常状态评估方法过度依赖数据标注、对时序数据的时间依赖性关注较低和系统异常状态难以量化等问题，提出一种基于混合生成网络的软件系统异常状态评估方法．首先，通过对长短期记忆网络（long short-term memory network，LSTM）与变分自动编码器（variational auto-encoder，VAE）的融合，设计一种LSTM-VAE混合生成网络，并以该网络为基础构建基于LSTM-VAE混合生成网络的系统异常状态检测模型，由LSTM对系统数据的时序特征进行提取并由VAE对系统数据的分布进行建模．然后，由LSTM-VAE异常状态检测模型处理系统关键特征参数，获取系统关键特征参数的异常度量值．最后，利用耦合度方法对传统的线性加权和方法进行优化，通过加权耦合度优化方法计算得到软件系统异常状态的量化值，从而实现对软件系统的异常状态评估．实验结果表明，本文模型对软件系统的异常时序数据具有较好的检测能力，其对系统异常状态的评估结果更为合理、有效．

关键词：软件系统；状态评估；长短期记忆网络；变分自动编码器；异常检测；耦合度
中图分类号：TP309 **文献标志码**：A

Evaluation of Software System Abnormal Status Based on Hybrid Generative Network

YANG Hongyu[1,2†], LI Yi[2], ZHANG Liang[3]
（1. College of Safety Science and Engineering, Civil Aviation University of China, Tianjin 300300, China;
2. College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China;
3. College of Information, University of Arizona, AZ 85721, USA）

Abstract: To solve the problems that the existing software system abnormal status evaluation methods over depend on data labeling and pay less attention to the time dependence of time-series data, and then it is difficult to quantify the software system abnormal status. Thus, a software system abnormal status evaluation method based on the hybrid generative network is proposed. Firstly, by combining the long short-term memory network (LSTM) and the variational auto-encoder (VAE), an anomaly detection model based on LSTM-VAE hybrid generative network is designed. The features of the system time-series data are extracted by LSTM and its distribution is modeled by VAE.

* **收稿日期**：2021-09-27
基金项目：国家自然科学基金民航联合研究基金项目（U1833107），Civil Aviation Joint Research Fund Project of National Natural Science（U1833107）
作者简介：杨宏宇（1969—），男，吉林长春人，中国民航大学教授，博士
† 通信联系人，E-mail：yhyxlx@hotmail.com

Then, the LSTM-VAE anomaly detection model detects the software system key feature parameters and obtains the anomaly metric value of system key feature parameters. Finally, the coupling degree method is used to optimize the linear weighted sum method. According to the weighted coupling degree method which is optimized, the software system abnormal status quantitative value is calculated, and the software system abnormal status is evaluated. The experimental results show that the proposed model has a better detection ability for the abnormal time-series data of the software system, and its system abnormal status evaluation result is more feasible and effective.

Key words: software system; status evaluation; long short-term memory network; variational auto-encoder; anomaly detection; coupling degree

软件系统作为社会生产方式和信息化发展的成果之一,正朝着复杂化的方向不断发展,系统一旦产生异常[1],将对软件系统的安全稳定运行造成影响[2].为克服软件系统异常解决方案中存在的盲目性和被动性,需要对软件系统进行及时、有效的状态评估.软件系统异常状态评估是从异常的角度对系统状态进行评估,分析异常事件对系统造成的危害程度,为制定科学合理的软件系统安全保障方案提供客观依据和基础支持.

现有系统状态评估方法主要包括基于数学模型、基于逻辑规则推理和基于神经网络的方法[3-5].基于数学模型的状态评估方法通过对影响系统正常运行的因素进行分析,建立影响系统运行因素与系统状态之间的对应关系.由于易受主观因素影响且实时性较差,基于数学模型的评估方法的评估结果不够理想,与实际情况偏差较大[6-7].基于逻辑规则推理的状态评估方法根据先验知识构建模型并使用逻辑规则推理方法对系统状态进行评估,凭借先验知识对状态指标设置阈值判断系统状态,使得评估结果主观性强[8-9].此外,由于软件系统运行过程中产生的状态信息量较大,基于逻辑规则推理的状态评估方法适应性较差.与基于数学模型和基于逻辑规则推理的状态评估方法相比,基于神经网络的状态评估方法由于具有高效和易拓展等特点应用更加广泛,该类方法利用训练数据集对特定模型进行训练后可以对系统状态进行分类,实现对系统状态的评估[10-12],常用的方法包括AVE[13]、深度挖掘[14]、卷积神经网络[15]等.基于神经网络的状态评估方法虽然对系统状态的评估效果较好,但该类方法的通用性和可扩展性较差,对系统状态的量化分类效果不佳.

针对上述问题,本文提出一种基于混合生成网络的软件系统异常状态评估方法.首先,通过对长短期记忆网络(long short-term memory network, LSTM)和变分自动编码器(variational auto-encoder, VAE)的融合,设计一个LSTM-VAE混合生成网络,以此为基础构建一种基于LSTM-VAE混合生成网络的系统异常状态检测模型.然后,采集系统关键特征参数数据,利用建立的LSTM-VAE异常状态检测模型对关键特征参数进行检测并获取其相应的异常度量值.最后,利用耦合度方法[16]对线性加权和方法进行优化,根据优化后得到的加权耦合度方法计算系统异常状态的量化值,实现对软件系统异常状态的量化评估.

1　软件系统异常状态评估方法

1.1　方法设计

基于混合生成网络的软件系统异常状态评估由基于长短期记忆网络-变分自动编码器(LSTM-VAE)的系统异常状态检测模型(简称LSTM-VAE异常检测模型)构建和系统异常状态评估两部分组成,如图1所示.LSTM-VAE异常检测模型构建为系统异常状态评估提供系统异常状态检测模型、异常阈值和最大重构误差;系统异常状态评估利用已构建完成的异常检测模型获取系统关键特征参数的异常度量值,然后通过加权耦合度方法对系统异常状态的量化值进行计算,根据系统异常状态的量化值实现软件系统异常状态评估.本文方法中两个部分的处理过程设计如下:

1.1.1　基于LSTM-VAE的系统异常状态检测模型构建

首先,筛选出系统正常运行时序的关键特征参数历史时序数据,输入LSTM-VAE混合生成网络中进行训练,由训练完成后的LSTM-VAE混合生成网

络获取系统正常运行状态下的关键特征参数时序数据的长短期依赖关系和分布形式.

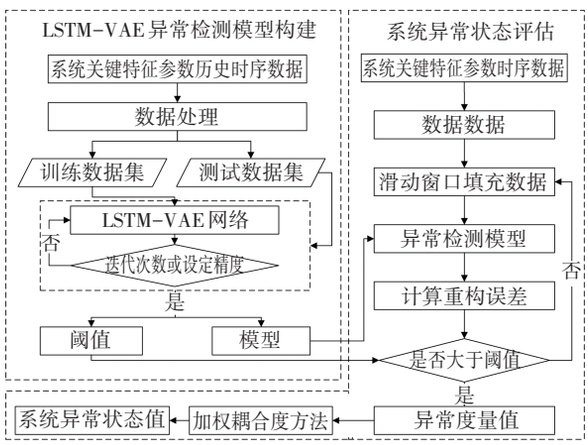


图1 软件系统异常状态评估方法

Fig.1 Software system abnormal status evaluation method

然后,将含有异常标注的关键特征参数的历史时序数据输入LSTM-VAE混合生成网络中,获取系统关键特征参数的重构误差,此时的重构误差表示关键特征参数偏离正常状态数据的分布程度.

最后,利用标注的异常信息与模型检测结果统计模型准确率与召回率,选择准确率等于召回率时的阈值为异常阈值,选择重构误差中最大的数值为最大重构误差.其中,异常阈值为该关键特征参数偏离正常状态下数据分布程度的下限,用于判断系统关键特征参数是否异常,最大重构误差为关键特征参数偏离正常状态下数据分布程度的上限.

1.1.2 系统异常状态评估

首先,采集系统关键特征参数的时序数据,利用已构建的LSTM-VAE异常检测模型对采集到的系统关键特征参数时序数据进行检测,获取系统各关键特征参数的异常度量值.

然后,依据AHP的权重评定原则,采用1-9标度法[17]确定系统各关键特征参数的相对重要度,对系统各关键特征参数进行权重赋值.

最后,利用加权耦合度方法计算系统异常状态的量化值,实现软件系统异常状态评估.

1.2 LSTM与GRU的对比分析

目前,采用门控机制的神经网络模型主要包括长短期记忆网络(LSTM)和门控循环单元(gated recurrent unit, GRU),LSTM和GRU的特点对比如表1所示.LSTM和GRU均是作为长、短期记忆的解决方案而提出的,两者都具有称为门的内部机制.不同之处在于:LSTM具有3个门控单元,而GRU相比LSTM少了一个门控单元,故从计算角度来看,GRU结构简单,效率更高.但是在数据集较大的情况下,与GRU相比,LSTM具有更强的表征能力[18].

表1 LSTM与GRU的比较

Tab.1 Comparison of LSTM and GRU

方法名称	优点	缺点
LSTM	参数多,表达更加灵活,对大数据集的适应性好	结构复杂,计算耗时,缺少正则化手段容易过拟合
GRU	结构简单,参数量少,减少过拟合的风险	数据集过少的情况下,泛化能力较弱

本文的研究是在数据集较大的场景下开展的,故选择使用LSTM.虽然LSTM牺牲了部分时间及计算的简便性,但LSTM对关键特征参数的表征能力更强也更灵活,对大数据环境的适应性更好.

2 基于混合生成网络的系统异常状态检测模型

2.1 变分自动编码器和长短期记忆网络

VAE的基本结构如图2所示.作为经典的无监督异常检测方法,VAE在模型训练时无须大量标注数据,可通过辨识输入数据与重构输出数据间的差异来达到异常检测的目的.首先,由编码器对输入样本$X=\{x_k|k=1, 2, \cdots, n\}$中的元素$X_k$进行拟合,使其服从均值为$u$和方差为$\sigma$的正态分布.然后,通过对所得正态分布进行采样得到隐变量$Z=\{z_k|k=1, 2, \cdots, n\}$,其中,元素$Z_k$服从均值为0和方差为1的标准正态分布.最后,由解码器对隐变量Z进行解码,生成输出样本$Y=\{y_k|k=1, 2, \cdots, n\}$,通过计算$X$与$Y$之间的均方误差获得输入样本与输出样本之间的重构误差:

$$u(t)=\sum_{i=1}^{r}h_i(z(t))F_i(t) \tag{1}$$

LSTM通过记忆单元和门机制提取时序数据内部的长短期依赖关系,其神经元结构如图3所示.LSTM的遗忘门和输入门控制单元C_t的输入,输出门控制单元C_t的输出.其中,遗忘门和输入门分别控制上一时刻的状态C_{t-1}和当前时刻的状态C_t^*输入记忆单元C_t中,输出门控制当前记忆单元C_t输入当前时刻的输出h_t中.

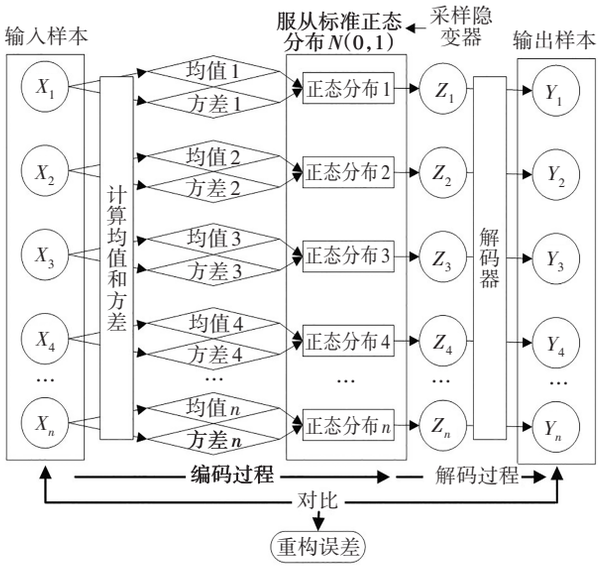


图 2　VAE 基本结构

Fig.2　Structure of VAE

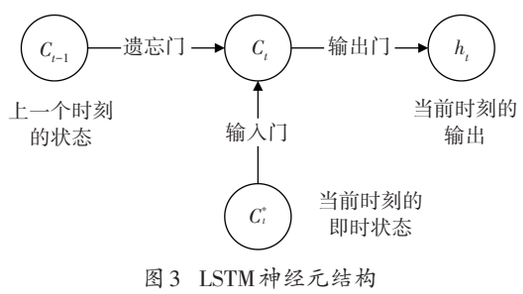


图 3　LSTM 神经元结构

Fig.3　Structure of LSTM neurons

由上文分析可知,VAE 和 LSTM 分别具有鲜明的特点,如表 2 所示.VAE 的训练无须大量标注数据并且采样过程具有随机性,有助于提高模型的泛化能力.但是,VAE 对系统时序数据的时序特征不敏感,无法获取并表征系统时序数据内部的长短期依赖关系.由于系统关键特征参数数据具备明显的时序特征,LSTM 可以通过遗忘门、输入门和输出门控制记忆单元的状态,使其能够获取并表征系统时序数据内部的长短期依赖关系.

表 2　VAE 和 LSTM 的特点

Tab. 2　Characteristic of VAE and LSTM

名称	优点	缺点
VAE	不需要大量异常标注的关键特征参数数据训练模型;采样过程具有随机性,能够提高模型的泛化能力	对时序数据内部长短期依赖关系缺少关注
LSTM	可以获取时序数据内部的长短期依赖关系	结构复杂,计算耗时,缺少正则化手段容易过拟合

2.2　基于 LSTM-VAE 的混合生成网络设计

为解决现有异常检测方法在训练过程中需要大量标注数据和缺少对时序数据内部长短期依赖关系关注的问题,本文将 LSTM 与 VAE 融合,设计一个 LSTM-VAE 网络(图 4).在该网络中,用 LSTM 神经元对 VAE 的编码层和解码层中的神经元进行替换,即采用 LSTM 对输入时序数据内部的长短期依赖关系进行提取,通过 VAE 的变分推理对系统时序数据的分布进行建模.

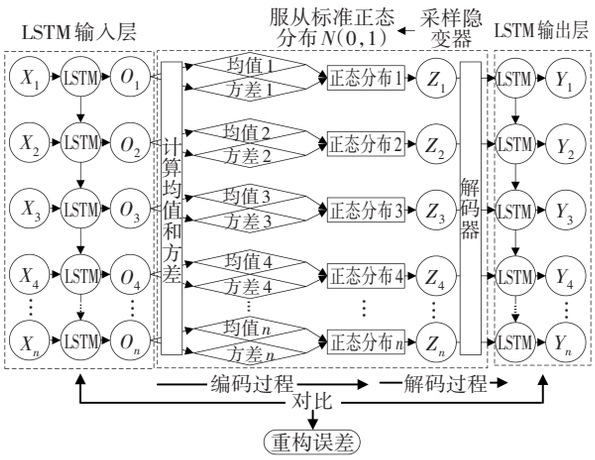


图 4　LSTM-VAE 网络结构

Fig.4　Structure of LSTM-VAE network

与单一的 LSTM 或 VAE 相比,LSTM-VAE 网络可以对系统关键特征参数时序数据的长短期依赖关系进行提取,无须大量标注的关键特征参数数据训练模型,同时模型在隐变量学习过程中利用采样的随机性起到正则化作用,可以防止网络过拟合,使得 LSTM-VAE 混合生成网络在提取时序数据特征和提高模型泛化能力方面更具优势.

在 LSTM-VAE 网络模型中,首先,输入序列为 $X=\{x_k|k=1, 2, \cdots, n\}$,经 LSTM 输入层解码后实现对输入序列 X 时间依赖性的学习,从而建立一个有低维隐层空间的序列 $O=\{O_k|k=1, 2, \cdots, n\}$.然后,将获得的序列 $O=\{O_k|k=1, 2, \cdots, n\}$ 输入 VAE 中进行编码,得到序列 O 的隐层空间样本序列 $Z=\{z_k|k=1, 2, \cdots, n\}$.最后,样本序列 Z 再通过解码网络与 LSTM 输出层进行拟合,生成重构的输出序列 $Y=\{y_k|k=1, 2, \cdots, n\}$.

2.3　基于混合生成网络的系统异常状态检测模型构建

以本文设计的 LSTM-VAE 混合生成网络为基础,构建一个系统异常状态检测模型.该异常检测模型的构建步骤设计如下.

输入:系统关键特征参数的历史时序数据.

输出:异常检测模型.

步骤1:选取与系统状态相关的关键特征参数.

选择系统CPU利用率、内存利用率、磁盘利用率和网卡吞吐率等特征参数作为系统状态异常检测和评估系统异常状态的关键特征参数,获取系统各关键特征参数的历史时序数据.

步骤2:系统关键特征参数预处理.

对系统关键特征参数进行归一化处理,抑制取值范围差异对训练产生的负面影响,

$$x=(x^{*}-x_{\min})/(x_{\max}-x_{\min}) \tag{2}$$

式中:x表示经过归一化之后的关键特征参数值,x^{*}表示初始关键特征参数值,$x_{\max}$表示关键特征参数中的最大值,$x_{\min}$表示关键特征参数中的最小值.

步骤3:数据集划分.

将系统关键特征参数历史时序数据划分为训练集与测试集,训练集中不含异常标签的数据,测试集中含有异常标签的数据.

步骤4:模型训练.

利用训练集对LSTM-VAE混合生成网络进行训练.采取滑动时间窗口法对关键特征参数的训练集序列进行子序列提取,假设时间窗口长度为l,关键特征参数序列长度为L,则可以从中提取$L-l+1$个子序列.假设$\{x_1, x_2, \cdots, x_l\}$为系统CPU利用率训练集中部分运行数据序列片段,其中,x_i为i时刻系统CPU利用率,l为序列片段长度,LSTM-VAE混合生成网络对$\{x_1, x_2, \cdots, x_l\}$进行网络重构后的输出序列为$\{y_1, y_2, \cdots, y_l\}$,采用均方根误差函数计算得到输入序列$\{x_1, x_2, \cdots, x_l\}$与重构输出序列$\{y_1, y_2, \cdots, y_l\}$之间的重构误差$e$

$$e=\mathrm{RMSE}(x,y)=\sqrt{\frac{1}{k}\sum_{i=1}^{l}(x_i-y_i)^2} \tag{3}$$

由混合生成网络输出的重构误差集合E为

$$E_{\mathrm{cpu}}=\{e_1,e_2,\cdots,e_{\mathrm{m}}\} \tag{4}$$

式中:$m=L-l+1$,m为CPU利用率训练集中数据序列片段个数,L为CPU利用率训练集中数据序列长度,l为序列片段长度也即滑动时间窗口长度.

同样地,采用相同方法,得到系统的内存利用率、磁盘利用率和网卡吞吐率等关键特征参数的重构训练误差集合.当关键特征参数重构训练误差集合的所有重构误差均达到设定的精度要求或模型达到设定的迭代次数时,结束训练.

步骤5:异常阈值选择与最大重构误差.

将含有标签的系统各关键特征参数测试集输入已训练完成的LSTM-VAE异常检测模型中.$\{X_1, X_2, \cdots, X_L\}$为系统CPU利用率测试集中部分运行数据序列片段,其对应的标注信息序列片段为$\{B_1, B_2, \cdots, B_L\}$,其中,$X_i$为$i$时刻系统CPU利用率数值,$B_i$为测试集中对应序列$X_i$的标注信息($B_i$=1表示$i$时刻该系统CPU利用率数值异常,$B_i$=0表示该时刻系统CPU利用率数值无异常),$L$为序列片段长度.

首先,将CPU利用率测试集数据输入训练完成后的LSTM-VAE异常检测模型中获取相应的重构输出序列,由公式(3)和公式(4)计算得到该关键特征参数测试集的重构误差集合$E_{\mathrm{cpu}}=\{e_1, e_2, \cdots, e_m\}$.

然后,定义CPU利用率阈值ζ,将CPU利用率测试集重构误差集合E_{cpu}中第i个数据序列片段的重构误差e_i与CPU利用率阈值ζ进行比较,若$e_i<\zeta$,则表示异常检测模型判定该序列片段无异常,用C_i=0表示;若$e_i\geqslant\zeta$,则表示异常检测模型判定该序列片段异常,用C_i=1表示.为评估模型性能,引入准确率和召回率指标.

准确率P:CPU利用率测试集中检测为异常的序列片段中标注为异常的序列片段的比例.

$$P=\frac{\sum_{i=1}^{m}[C_i=B_i=1]}{\sum_{i=1}^{m}[C_i=B_i=1]+\sum_{i=1}^{m}[C_i=1\cap B_i=0]},\quad C_i=\begin{cases}1, e_i\geqslant\zeta\\0, e_i<\zeta\end{cases} \tag{5}$$

召回率R:CPU利用率测试集中标注为异常的序列中检测为异常的序列片段的比例.

$$R=\frac{\sum_{i=1}^{m}[C_i=B_i=1]}{\sum_{i=1}^{m}[C_i=B_i=1]+\sum_{i=1}^{m}[C_i=0\cap B_i=1]},\quad C_i=\begin{cases}1, e_i\geqslant\zeta\\0, e_i<\zeta\end{cases} \tag{6}$$

式中:$\sum_{i=1}^{m}[C_i=B_i=1]$为CPU利用率测试集中检测为异常的异常标注序列片段个数,$\sum_{i=1}^{m}[C_i=1\cap B_i=0]$为CPU利用率测试集中检测为异常但标注为正常的序列片段个数,$\sum_{i=1}^{m}[C_i=0\cap B_i=1]$为CPU利用率测试集中检测为正常但标注为异常的序列片段个数.

最后,由准确率P和召回率R的定义可知,准确率P随阈值ζ的增大而增大,召回率R随阈值ζ增大而减小.当准确率P等于召回率R时,模型性能最佳.因此,定义此时的阈值ζ为CPU利用率的异常阈值

ζ^*,其中,CPU 利用率的异常阈值ζ^*表示该关键特征参数偏离正常状态下数据分布程度的下限.同时,定义该关键特征参数的最大重构误差e_{cpu}为

$$e_{\text{cpu}} = \max|T_{\text{cpu}}| = \max_{1 \leqslant i \leqslant m}|e_i| \tag{7}$$

式中:e_{cpu}为 CPU 利用率的最大重构误差,表示该关键特征参数偏离正常状态下数据分布程度的上限.

同样地,采用相同方法,得到系统的内存利用率、磁盘利用率和网卡吞吐率等关键特征参数的异常阈值和最大重构误差.

3　系统异常状态评估

在完成LSTM-VAE异常检测模型构建并获得系统各关键特征参数的异常阈值和最大重构误差后,将系统各关键特征参数的时序数据输入LSTM-VAE异常检测模型中,获取系统各关键特征参数的异常度量值.然后对系统各关键特征参数的权重进行赋值.最后利用耦合度方法对线性加权和方法进行优化,通过加权耦合度优化方法计算得到系统异常状态值.

系统异常状态评估部分的具体步骤设计如下:

输入:各关键特征参数的时序数据.

输出:系统异常状态值.

步骤1:获取各关键特征参数的异常度量值.

将系统关键特征参数时序数据输入LSTM-VAE异常检测模型中,计算输入序列与输出序列之间的重构误差.当重构误差小于异常阈值时,重新获取时序数据进行异常检测;否则,判断该关键特征参数为异常.然后,基于已获得的重构误差、异常阈值与最大重构误差,计算该关键特征参数的异常度量值

$$I_i = \begin{cases} 0 & (e_i \leqslant \zeta_i^*) \\ \dfrac{\left|e_i - \zeta_i^*\right|}{e_{\max} - \zeta_i^*} & (e_{\max} > e_i > \zeta_i^*) \\ 1 & (e_i \geqslant e_{\max}) \end{cases} \tag{8}$$

式中:I_i为系统第i个关键特征参数的异常度量值,e_i为该关键特征参数此时的重构误差,ζ_i^*为系统第i个关键特征参数的异常阈值,$e_{\max}$为系统第i个关键特征参数的最大重构误差.

步骤2:各关键特征参数权重赋值.

依据AHP的权重评定原则,采用1-9标度法确定系统各关键特征参数的相对重要度,构造决策矩阵$\boldsymbol{M}=(m_{ij})_{n\times n}$,决策矩阵$\boldsymbol{M}$为

$$\boldsymbol{M} = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nn} \end{bmatrix} \tag{9}$$

式中:n为系统关键特征参数的个数,元素m_{ij}表示第i个关键特征参数与第j个关键特征参数的重要程度之比,当$i=j$时,$m_{ij}=1$.

计算决策矩阵$\boldsymbol{M}$的最大特征根$\lambda_{\max}$及其特征向量$W=(w_1, w_2 \cdots, w_n)$,

$$\lambda_{\max} = \sum_{i=1}^{n} \frac{(WV)_i}{nv_i} \tag{10}$$

式中:$v_i = n(\prod_{j=1}^{n} m_{ij})^{1/2}$,$V=(v_1, v_2, \cdots, v_{\text{n}})^{\text{T}}$,$i,j=1,2,\cdots,n$.

最大特征值$\lambda_{\max}$对应的特征向量W即为各关键特征参数的权重集合$(w_1, w_2 \cdots, w_n)$.

步骤3:改进加权耦合度方法计算系统异常状态值.

线性加权和方法仅强调变量之间的独立性,不考虑变量间数值差异与相互作用问题,耦合度方法常用于表示变量间数值差异与相互作用.软件系统作为一个整体,其不同组件及关键特征参数之间的相互作用可能影响系统的正常运行.忽略不同组件及关键特征参数之间的差异和相互作用将导致系统状态评估时结果出现较大的差异.因此,本文在线性加权和方法的基础上融入耦合度方法计算系统异常状态值,即在量化系统异常状态时关注不同关键特征参数异常度量值间的差异给量化结果带来的影响.

首先,基于步骤1获取的系统各关键特征参数的异常度量值I_i,计算系统耦合度H

$$H = n\frac{\sqrt[n]{(1-I_1)\cdots(1-I_n)}}{(1-I_1)+\cdots+(1-I_n)} \tag{11}$$

然后,利用系统各关键特征参数权重w_i和系统各关键特征参数的异常度量值I_i,计算系统关键特征参数异常度量值的线性加权和S

$$S = \sum_{i=1}^{n} w_i I_i \tag{12}$$

最后,基于系统耦合度H和系统关键特征参数异常度量值的线性加权和S,计算系统异常状态值S_{a}

$$S_{\text{a}} = S^H \tag{13}$$

其中,系统耦合度H与系统关键特征参数异常度量值的线性加权和S的值均在[0,1]范围内.

根据指数函数的定义可知,式(13)在[0,1]范围

内为单调递减函数,在S一定的条件下,H越小,则S_a越大;同样地,在H一定的条件下,S越大,则S_a越大.因此,根据该方法计算得到的系统异常状态量化结果符合本文的评估思路,即在计算系统异常状态值时关注不同特征参数的异常度量值之间的差异对评估结果造成的影响.

4 实验与结果分析

4.1 实验数据

实验数据集采用百度公司联合清华大学公开的运维数据集,选择其中的CPU利用率、内存利用率、磁盘利用率与网卡吞吐率这四项关键特征参数进行异常检测验证实验.数据集中每个关键特征参数均包含连续4周的数据,时间间隔为5 min.在实验中,将各关键特征参数的前三周数据划分为训练集,将最后一周数据划分为测试集,其中,训练集中数据均为系统正常运行时序数据且无异常数据,测试集中含有异常数据且异常数据已经标注.部分关键特征参数的数据格式如表3所示.

表3 关键特征参数数据格式

Tab. 3 Data formats for key feature parameters

KPI ID	TimeStamp	Value	Label
1	1469894400	0.216052574	0
1	1469894700	0.226608083	0
1	1469895000	0.218362752	0
…	…	…	…
1	1470577800	0.289618012	1

4.2 LSTM-VAE混合生成网络配置及模型训练

LSTM-VAE混合生成网络由LSTM和VAE构成,输入向量维数等于输出向量维数,具体网络的配置如表4所示.

表4 LSTM-VAE网络配置

Tab. 4 Configuration of LSTM-VAE network

参数名称	参数值/名称	参数名称	参数值/名称
训练迭代次数	50	激活函数	sigmoid
优化算法	PMSProp算法	学习率	0.01
输入层神经元个数	32	输出层神经元个数	32

在训练过程中,各关键特征参数的训练误差与迭代次数关系如图5所示.由图5可见,各关键特征参数的训练误差随着迭代次数的增加而迅速减小.其中,各关键特征参数的误差曲线在经过35次迭代后趋于平稳,表明模型达到较好的收敛效果,能够对系统正常的关键特征参数时序数据的长短期依赖关系和数据分布形式进行学习并实现数据重构.

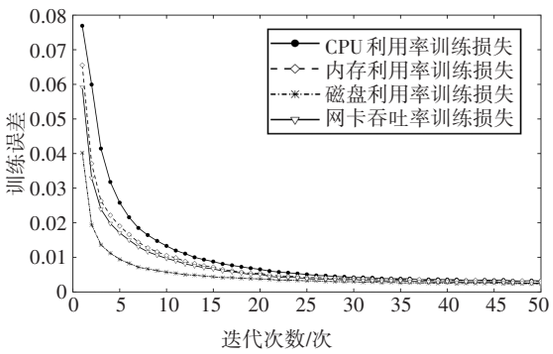


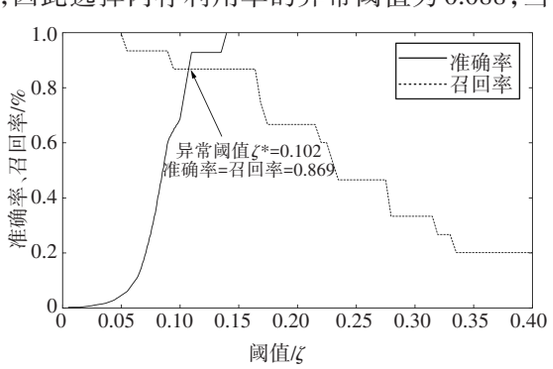
图5 训练误差与迭代次数

Fig.5 Diagram of training error and number of iterations

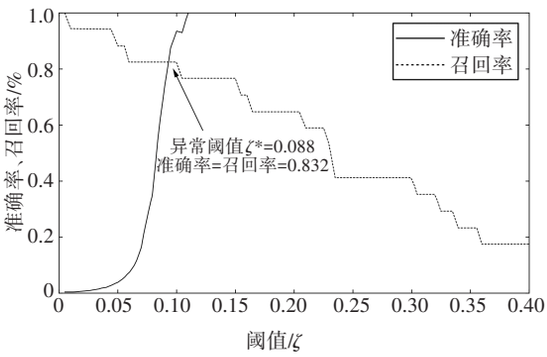
4.3 异常阈值选择与评估效果

在实验中,通过遍历所有的可取阈值,确定异常阈值.可取阈值范围为0到该关键特征参数的最大重构误差.设置初始阈值为0,若关键特征参数的重构误差大于0,将其视为异常;否则,首先计算阈值为0时的准确率和召回率,然后增加阈值并计算对应的准确率和召回率,当阈值设置为该关键特征参数的最大重构误差时结束.利用训练完成后的LSTM-VAE异常检测模型对系统各关键特征参数的测试集进行异常检测,得到相应的阈值与准确率、召回率的变化关系,如图6所示.

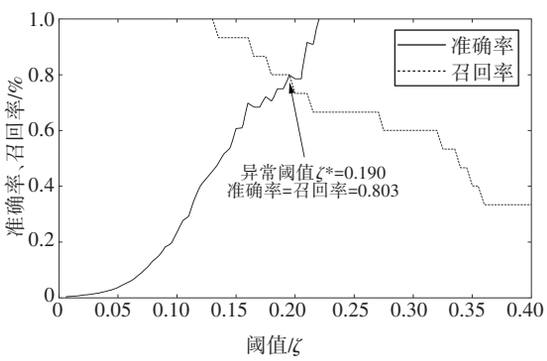
由图6可见,当CPU利用率的异常阈值为0.102时,CPU利用率的准确率与召回率相等,因此选择CPU利用率的异常阈值为0.102;当内存利用率的异常阈值为0.088时,内存利用率的准确率与召回率相等,因此选择内存利用率的异常阈值为0.088;当磁



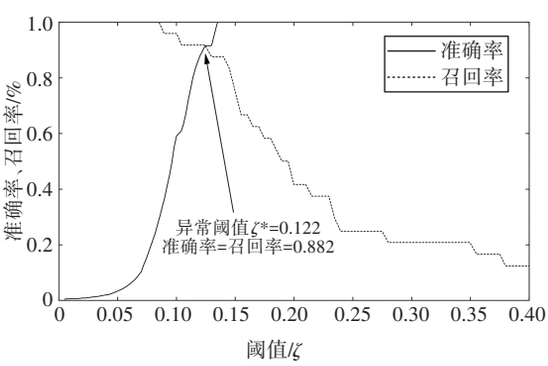
(a)CPU利用率阈值与准确率、召回率关系图



(b)内存利用率阈值与准确率、召回率关系图



(c)磁盘利用率阈值与准确率、召回率关系图



(d)网卡吞吐率阈值与准确率、召回率关系图

图6　各关键特征参数的阈值与准确率、召回率关系图

Fig.6　Diagram of relationship between threshold values and accuracy rate, recall rate

盘利用率的异常阈值为0.190时,磁盘利用率的准确率与召回率相等,因此选择磁盘利用率的异常阈值为0.190;当网卡吞吐率的异常阈值为0.122时,网卡吞吐率的准确率与召回率相等,因此选择网卡吞吐率的异常阈值为0.122.

基于系统各关键特征参数测试集,计算得到各关键特征参数对应的最大重构误差,结果如表5所示.

为评价本文异常检测模型的效果,分别采用VAE、AE、LSTM-AE和LSTM-VAE网络得到不同关键特征参数评估的$F1$-score值,$F1\text{-score}=(2\times P\times R)/(P+R)$.各模型的$F1$-score值如表6所示.

表5　各关键特征参数的最大重构误差

Tab. 5　Maximum reconstruction error of each key feature parameter

	CPU利用率	内存利用率	磁盘利用率	网卡吞吐率
最大重构误差	0.523	0.496	0.674	0.562

表6　不同网络在不同关键特征参数测试集中的$F1$-score值

Tab. 6　$F1$-score values of different models in different key feature parameter test sets

网络名称	CPU利用率	内存利用率	磁盘利用率	网卡吞吐率
LSTM-VAE	0.869	0.823	0.803	0.882
VAE	0.784	0.810	0.733	0.752
LSTM-AE	0.738	0.764	0.693	0.713
AE	0.673	0.719	0.627	0.663

由表6可见,针对测试集中的CPU利用率、内存利用率、磁盘利用率和网卡吞吐率4个关键特征参数测试数据,LSTM-VAE网络模型的$F1$-score值均优于其他模型.与VAE、AE模型相比,LSTM-VAE模型利用LSTM神经网络的时序特征提取能力,能够有效挖掘关键特征参数时序数据内部的长短期依赖关系,显著提升异常检测精度;同时,LSTM-VAE网络可以利用VAE的隐变量空间,减少LSTM-AE网络中神经网络的过拟合对异常检测效果的影响,也有助于提升异常检测效果.

4.4　滑动时间窗口长度确定

由于滑动时间窗口长度L的取值会影响LSTM-VAE模型的异常检测效果,为了验证滑动时间窗口长度对LSTM-VAE模型的异常检测效果的影响并确定最佳的滑动时间窗口长度值,有必要进行滑动时间窗口长度影响实验.

在实验中,选取不同窗口长度L的值并计算其对应的$F1$-score值,选择窗口长度值从长度4到20依次增加2个单位长度进行实验.选择最高$F1$-score值所对应的窗口长度作为LSTM-VAE模型的滑动时间窗口长度L的值,不同滑动时间窗口长度L与对应的$F1$-score值如图7所示.

由图7可见,当滑动时间窗口长度L为12时,LSTM-VAE模型对各关键特征参数测试数据的$F1$-

score值均大于其他滑动时间窗口长度对应的F1-score值,表明滑动时间窗口长度为12时,LSTM-VAE模型对系统状态的异常检测效果最佳,所以,在LSTM-VAE模型中,将滑动时间窗口长度L的值设定为12.

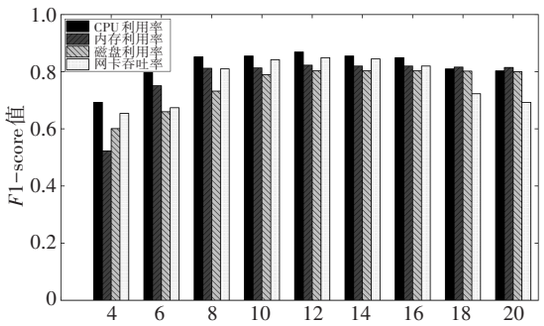


图7 滑动时间窗口长度与$F1$-score值

Fig.7 Sliding time window lengths and the corresponding $F1$-score values

4.5 异常状态评估

4.5.1 关键特征参数异常度量值与关键特征参数赋权

在确定系统各关键特征参数的异常阈值和最大重构误差后,为保证实验结果具有可比性,选取10组关键特征参数异常时序数据组成测试集,用本文模型的方法对该数据集进行检测并计算关键特征参数的重构误差,由公式(7)计算测试集中关键特征参数的异常度量值,由公式(11)计算各测试集的系统耦合度,测试集中各关键特征参数的异常度量值如图8所示,测试集中各关键特征参数的系统耦合度如图9所示.

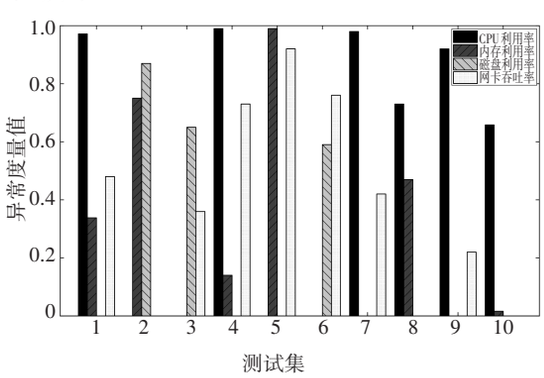


图8 测试集中各关键特征参数的异常度量值

Fig.8 Key feature parameters anomaly metric value in the test group

采用加权耦合度方法计算系统异常状态值时,需要对系统各关键特征参数进行赋权.利用AHP方法对系统关键特征参数进行主观赋权,根据AHP权重评定原则,采用1-9标度法确定系统各关键特征参数的相对重要度,建立决策矩阵如表7所示.

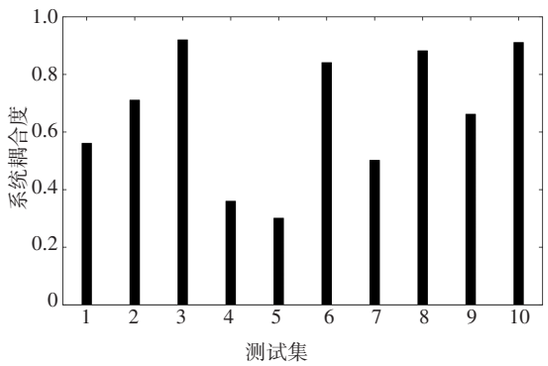


图9 各测试集的系统耦合度

Fig.9 System coupling degree of each test group

表7 决策矩阵

Tab. 7 Decision matrix

关键特征参数	CPU利用率	内存利用率	磁盘利用率	网卡吞吐率
CPU利用率	1	2	3	3
内存利用率	1/2	1	3	3
磁盘利用率	1/3	1/3	1	1
网卡吞吐率	1/3	1/3	1	1

计算得到决策矩阵的最大特征根及其对应的特征向量,如表8所示.

表8 最大特征值及其特征向量

Tab. 8 Maximum eigenvalue and its eigenvector

最大特征值	特征向量(w_1, w_2, w_3, w_4)
4.0606	(0.443 5,0.312 1,0.122 2,0.122 2)

由表8可见,CPU利用率权重为0.443 5,内存利用率权重为0.312 1,磁盘利用率权重为0.122 2,网卡吞吐率权重为0.122 2.

4.5.2 系统异常状态评估

由4.5.1节的实验过程得到测试集中各关键特征参数的异常度量值、各测试集的系统耦合度和相应的关键特征参数权重,由公式(13)计算各测试集的系统关键特征参数异常状态值.在系统异常状态评估阶段,使用线性加权和方法、TOPSIS方法和本文方法计算得到系统的异常状态结果,如图10所示.

由图10可见,在测试集4、5和7中,与线性加权和方法和TOPSIS方法得到的异常状态值相比,本文方法得到的异常状态值差异较为明显.同时由图8可见,在测试集4、5和7中,多个特征参数的异常度量值较高,且不同特征参数的异常度量值之间差异较大.同时由图9可见,与其他测试集的系统耦合度值相比,测试集4、5和7的系统耦合度值较小,在系统特征参数异常度量值的线性加权和一定的条件下,耦合度值越小的测试集,其系统异常状态值越大.因此,与其他方法计算得到的异常状态值相比,本文方法计算得到的测试集4、5和7的异常状态值差异较为明显.

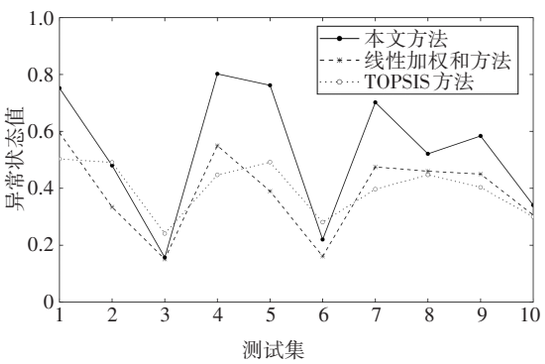


图10　实验结果对比图

Fig.10　Comparison of experimental results

同样地,在测试集3、6和10中,与线性加权和方法和TOPSIS方法得到的异常状态值相比,本文方法得到的异常状态值差异较小.同时由图8可见,在测试集3、6和10中,其总体特征参数的异常度量值较低,且不同特征参数的异常度量值之间差异较小.同时由图9可见,与其他测试集的系统耦合度值相比,测试集3、6和10的系统耦合度值较大,在系统特征参数异常度量值的线性加权和一定的条件下,耦合度越大的测试集,其系统异常状态值越接近特征参数异常度量值的线性加权和.因此,与其他方法计算得到的异常状态值相比,本文方法计算得到的测试集3、6和10的异常状态值差异较小.

上述结果的原因在于,本文方法在计算系统异常状态值时考虑了系统耦合度因素.由于耦合度能够反映系统各关键特征参数异常度量值间的差异,但不能反映系统异常状态值大小,因此本文方法在计算系统异常状态值时,在线性加权和的基础上融入耦合度因素,即在计算系统异常状态值时能够关注不同关键特征参数异常度量值之间的差异,故本文方法得到的结果更为合理.

5　结　论

针对现有软件系统异常状态评估方法过度依赖数据标注、对时序数据的时间依赖性关注较低和系统异常状态难以量化等问题,本文提出一种基于混合生成网络的软件系统异常状态评估方法.首先,针对异常检测方法过度依赖数据标注和对时序数据的时间依赖性关注较低等问题,设计一种基于LSTM-VAE混合生成网络的异常检测模型,解决异常检测模型应用场景受限和准确率较低的问题.然后,利用LSTM-VAE异常检测模型对系统关键特征参数进行检测并对其异常度量值进行计算,为后续系统异常状态评估提供可靠的数据支撑.最后,通过加权耦合度优化方法计算系统异常状态值,解决传统软件系统状态评估方法难以对系统异常状态进行量化的问题.实验结果表明,本文方法对系统异常时序数据的时间特征更为敏感,评估结果也更为合理、有效.

由于本文在系统关键特征参数权重赋值过程中存在一定的主观因素,可能导致软件系统异常状态的评估结果随评估主体的不同而改变,下一步将重点研究采用主客观相互结合[19]的赋权方法以减少主观因素对软件系统异常状态评估的影响.

参考文献

[1] HUCH F, GOLAGHA M, PETROVSKA A, *et al*. Machine learning-based run-time anomaly detection in software systems: an industrial evaluation [C]//2018 IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE).2018:13-18.

[2] CHEN H Y,TU S S,ZHAO C Y,*et al*. Provenance cloud security auditing system based on log analysis [C]//2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS).2016:155-159.

[3] HE J L, SHI Z K. Performance evaluation and information systems based on software requirements analysis—A case study of China [C]//2010 6th International Conference on Advanced Information Management and Service (IMS). IEEE,2010:122-126.

[4] YIN L,ZHU B. Study on supply chain information systems performance evaluation based on fuzzy AHP [C]//2010 International Conference on Information, Networking and Automation (ICINA). IEEE, 2010:223-226.

[5] BROWN A,TUOR A,HUTCHINSON B,*et al*. Recurrent neural network attention mechanisms for interpretable system log anomaly detection [C]//Proceedings of the First Workshop on Ma-

chine Learning for Computing Systems. New York: ACM, 2018: 1-8.

[6] JIA Y Y, WU H Y, JIANG D X. A hierarchical framework of security situation assessment for information system[C]//2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery.2015:23-28.

[7] WANG D Q, LU Y M, GAN J F. An information security evaluation method based on entropy theory and improved TOPSIS[C]// 2017 IEEE Second International Conference on Data Science in Cyberspace (DSC).2017:595-600.

[8] 杨卓群,金芝. 面向环境与需求不确定性的系统自适应决策[J]. 计算机研究与发展,2018,55(5):1014-1033.
YANG Z Q, JIN Z. Self-adaptive decision making under uncertainty in environment and requirements[J]. Journal of Computer Research and Development, 2018, 55(5): 1014-1033. (In Chinese)

[9] ALRAJEH D, KRAMER J, RUSSO A, *et al*. Learning operational requirements from goal models[C]//2009 IEEE 31st International Conference on Software Engineering. Vancouver, BC, Canada: IEEE, 2009:265-275.

[10] 丁小欧,于晟健,王沐贤,等. 基于相关性分析的工业时序数据异常检测[J]. 软件学报,2020,31(3):726-747.
DING X O, YU S J, WANG M X, *et al*. Anomaly detection on industrial time series based on correlation analysis[J]. Journal of Software, 2020, 31(3): 726-747. (In Chinese)

[11] 杨宏宇,王峰岩. 基于无监督多源数据特征解析的网络威胁态势评估[J]. 通信学报,2020,41(2):143-154.
YANG H Y, WANG F Y. Network threat situation assessment based on unsupervised multi-source data feature analysis[J]. Journal on Communications, 2020, 41(2): 143-154. (In Chinese)

[12] KUMARAGE T, RANATHUNGA S, KURUPPU C, *et al*. Generative adversarial networks (GAN) based anomaly detection in industrial software systems [C]//2019 Moratuwa Engineering Research Conference (MERCon). Moratuwa, Sri Lanka: IEEE, 2019:43-48.

[13] 张圣林,林潇霏,孙永谦,等. 基于深度学习的无监督KPI异常检测[J]. 数据与计算发展前沿,2020,2(3):87-100.
ZHANG S L, LIN X F, SUN Y Q, *et al*. Research on unsupervised KPI anomaly detection based on deep learning[J]. Frontiers of Data & Computing, 2020, 2(3): 87-100. (In Chinese)

[14] 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. New York: ACM, 2017: 1285-1298.

[15] LU S Y, WEI X, LI Y D, *et al*. Detecting anomaly in big data system logs using convolutional neural network[C]//2018 IEEE 16th Intl Conference on Dependable, Autonomic and Secure Computing. Athens, Greece: IEEE, 2018: 151-158.

[16] LIU Y P, LEI S, LIU W. Research on coupling degree model and application of IT resources and IT application capabilities[C]// The 26th Chinese Control and Decision Conference (2014 CCDC). Changsha: IEEE, 2014: 140-144.

[17] 姚远,潘传幸,张铮,等. 多样化软件系统量化评估方法[J]. 通信学报,2020,41(3):120-125.
YAO Y, PAN C X, ZHANG Z, *et al*. Method of quantitative assessment for diversified software system[J]. Journal on Communications, 2020, 41(3): 120-125. (In Chinese)

[18] YANG S D, YU X Y, ZHOU Y. LSTM and GRU neural network performance comparison study: taking yelp review dataset as an example[C]//2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI). Shanghai: IEEE, 2020:98-101.

[19] WANG B H, ZHANG S. A subjective and objective integration approach of determining weights for trustworthy measurement[J]. IEEE Access, 2018, 6: 25829-25835.