**Computer Science and Application** 计算机科学与应用, 2022, 12(3), 755-767
Published Online March 2022 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2022.123077

Hans汉斯

# 基于GCN-LSTM的云计算服务异常检测算法

**石　林[1]，郭炜彬[1]，吴卓儒[2]**

[1]广东工业大学计算机学院，广东　广州
[2]广东工业大学自动化学院，广东　广州

收稿日期：2022年2月25日；录用日期：2022年3月24日；发布日期：2022年3月31日

## 摘　要

**随着云计算服务架构日渐庞大，运维人员需要对大量性能指标数据进行检测以确保云计算各系统和业务的可靠和稳定。为提高服务器异常检测准确率，本文提出了一种基于GCN和LSTM的云计算服务异常检测算法。首先，通过建立图模型描述云计算服务器的空间特征和属性，并通过GCN模型提取其空间信息；然后将不同时刻的空间信息构成时间序列输入到LSTM模型从中提取时间信息，使用训练好的GCN-LSTM模型对时间序列进行重建；最后使用基于Copula函数的方法对重构误差进行异常检测。本文使用了来自大数据批处理系统的MBD数据进行实验，实验结果表明我们提出的模型具有有效性。**

## 关键词

**云计算服务，异常检测，GCN，LSTM，时间序列，Copula**

# Cloud Computing Service Anomaly Detection Algorithm Based on GCN-LSTM

**Lin Shi[1], Weibin Guo[1], Zhuoru Wu[2]**

[1]School of Computer Science, Guangdong University of Technology, Guangzhou Guangdong
[2]School of Automation, Guangdong University of Technology, Guangzhou Guangdong

Received: Feb. 25th, 2022; accepted: Mar. 24th, 2022; published: Mar. 31st, 2022

## Abstract

**With the increasing scale of cloud computing service architecture, the operating personnel need to detect a large amount of performance data to ensure the reliability and stability of cloud computing systems and businesses. In order to improve the detection accuracy of the servers, this paper proposes a cloud computing service anomaly detection algorithm based on GCN-LSTM. This algorithm**

文章引用: 石林, 郭炜彬, 吴卓儒. 基于 GCN-LSTM 的云计算服务异常检测算法[J]. 计算机科学与应用, 2022, 12(3): 755-767. DOI: 10.12677/csa.2022.123077

**first describes the spatial features and attribute of cloud computing server by constructing a graph model, extracts its spatial information by GCN; then inputs the spatial information at different times into LSTM to extract time information, and uses the trained GCN-LSTM model to reconstruct the time series; the method based on Copula function is used to detect the anomaly of reconstruction error finally. In this paper, MBD data from big data batch processing system is used for experiments. The experimental results show that the model we proposed is effective.**

## Keywords

**Cloud Computing Service, Anomaly Detection, GCN, LSTM, Copula**



## 1. 引言

云计算服务，指的是基于互联网的相关服务的增加、使用和交互模式，通过互联网技术来提供可扩展以及可虚拟化的资源。但由于其服务架构日渐庞大，系统相对应的性能指标数据也会随之增多，都需要大量的运维人员来对系统环境进行部署和维护，González 等人[1]曾指出一个简单的异常可能会带来无法想象的连锁反应。

然而，异常目前还没有一个统一的定义，不同的应用场景和领域发生的异常也会有不同的定义。著名统计学家 Hawkins 等人所定义的异常值[2]是指偏离整体样本的观察值，是在所有样本中偏离大多数正常数据的极端值。经过长年的发展，异常检测的相关工作分成了不同类型。基于规则的异常检测方法主要包括 Cohen 等人[3]提出的 RIPPER 和 Quinlan [4]提出的 C4.5 算法等典型算法，该类异常检测算法具有算法简单等优点，但是需要专家经验和人工辅助，不能只依靠数据集中学习到的规律，并且规则也会随着数据集的变化而变化；基于统计的异常检测方法最早是由 Barnett 等人[5]提出，研究数据集服从于某种分布(如正态分布、二项式分布或泊松分布)或概率模型(如高斯模型、回归模型或直方图)，该类方法根据数据分布能够快速有效地找出异常数据，但其只适用于单变量离群检测，较难确定模型概率分布，检测效率较低，对于高维数据处理能力不足。

随着信息技术的不断提高，对数据采集的颗粒度越来越细，基于规则和统计的异常检测方法在正确率和效率上已无法满足需求。因此，基于机器学习的方法开始被提出，这类异常检测方法可以分为以下四类：1) 基于距离的异常检测。其代表是 Ramaswamy 等人[6]提出使用基于距离的方法解决异常检测问题，采用 K-最近邻(K-nearest neighbor, KNN)算法，这种方法无须了解数据分布，无须带标签的训练集，数据类型要求不高，但是由于其计算复杂度高，难以确定参数，在应用中也受到了一定的限制。2) 基于密度的异常检测。Breunig [7]等人提出的基于局部离群因子(local outlier factor, LOF)是其中一个经典的方法，通过离群强度概念量化异常程度使其异常检测效果较好，但是时间成本和复杂度随维数增加，同时参数设置较难。3) 基于概率统计的异常检测。Li Z 等人提出的 COPOD 算法[8]是一种基于统计概率的异常检测算法，优点在于运行开销小，速度快并且不需要设置参数；Zheng Li 等人[9]提出的 ECOD 算法是一种计算速度快，且适用于高维数据集的算法。4) 基于分类的异常检测：南京大学周志华教授团队提出了 iForest 算法[10]，这种基于树的算法凭借线性的时间复杂度和优秀的准确率已广泛运用在工业界的结构化数据中，但建树完毕后仍然有大量的维度没有被使用，所以不适合高维数据的异常检测。

近年来，深度学习成为人工智能和机器学习中极为重要的部分。通过对数据的特征表示和提取以及设定不同类型的异常分数，基于深度学习的异常检测更是取得了巨大的进展。2018 年图灵奖得主 Yoshua Bengio 教授团队提出的基于自编码器的异常检测方法[11]是许多深度异常检测模型的核心，Wang 等人提出的 DeepFD 方法[12]运用了自编码器、图嵌入等技术来检测二分图中的结构异常；Zhang 等人提出的城市时空异常检测方法[13]使用了时空特征神经网络来检测时空数据的异常；Zheng 等人提出的 OCAN 方法[14]运用了自编码器、LSTM 以及生成式对抗网络(GAN, Generative Adversarial Networks)等技术实现了时间序列异常检测；基于 LSTM 的异常检测方法 Deeplog [15]和 LogAnomaly [16]也在日志异常检测中被提出。尽管基于深度学习的异常检测算法不断涌现，但是当前的深度学习算法在一定程度上体现出来的是可解释性较差，算法结果的好坏依赖于距离定义的方法，即阈值的计算与选择比较困难，同时在训练模型的时候大量的模型会占用过多的计算资源。

在本文中，我们提出了一种基于 GCN-LSTM 的云计算服务异常检测算法，联合提取云计算服务器的空间特征与时间特征来进行构造。首先通过图卷积神经网络来学习云计算服务器之间的空间信息，然后利用长短期记忆神经网络学习云计算服务器的时间序列信息。使用两者共同建立可重构时间序列重构模型，然后根据重构值和真实值的偏差，使用 COPOD [8]来训练误差并定义异常结果，最终构建云计算服务异常检测算法。该方法为一种无监督算法，使用时无需额外的超参数优化，同时可以体现一定的解释性。

## 2. 研究方法

### 2.1. 任务描述及算法框架

为了保证云计算服务的性能和可靠性，运营商需要持续监控系统的状态性能指标。其中的指标包括 CPU 使用率、每秒 I/O 请求数、网络吞吐量等，这些指标可以为了解云计算服务器的运行状态检测提供有效的参考信息。当服务器相关性能指标的观测值偏离大多数正常数据时，若运维人员没有有效检测并介入进行维护，表现到真实世界的可能是整个云计算服务器的网络故障告警风暴。因此，本文的异常检测任务便是充分利用相关信息，检测出云计算服务器异常的历史时间快照，并设计一个异常检测算法。

本文提出的异常检测方法首先通过建立图模型描述云计算服务器的空间特征和属性，并通过 GCN 模型提取其空间信息，然后将不同时刻的空间信息构成时间序列输入到 LSTM 模型从中提取时间信息，使用训练好的 GCN-LSTM 模型对时间序列进行重建。最后使用基于 Copula 函数的方法对重构误差进行尾端概率的判定和异常分数的确定并最后返回异常判断结果。所提出的算法框架图如图 1 所示，GCN-LSTM 序列重构模型表示将云服务器时长度为$n$的历史指标数据 $X_i$ 重构为 $X_i^p$；通过 $X_i$ 与 $X_i^p$ 相减的方式误差

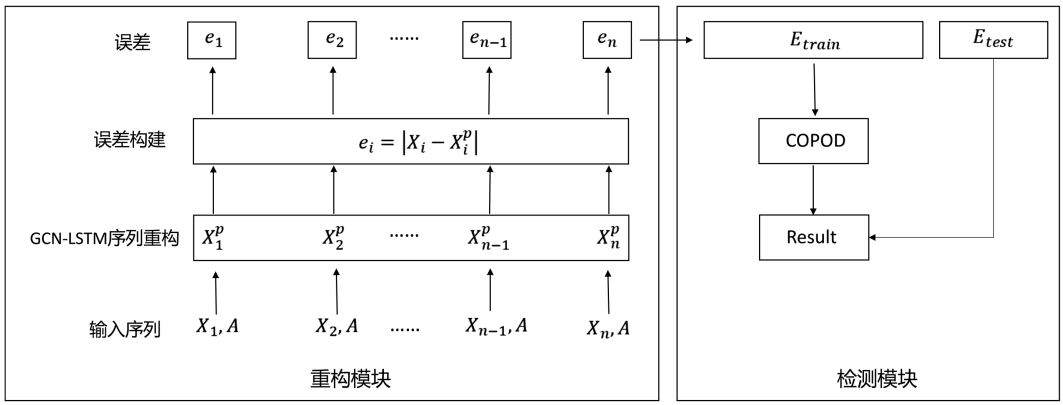


**Figure 1.** Algorithm framework
**图 1.** 算法框架

构建 $e_i$ 并将重构误差数据按照比例划分为 $E_{train}$ 和 $E_{test}$；使用 $E_{train}$ 数据集训练 COPOD 方法，获得异常阈值和异常分数后，使用 $E_{test}$ 数据集进行异常检测任务。

图 1 中各部分模块将在本章进行介绍。

## 2.2. 基础知识

### 2.2.1. 图卷积神经网络

图神经网络(Graph Neural Network, GNN)是近年来出现的面向图结构数据的深度学习技术。由于图神经网络处理的数据结构是图，在处理非欧几里得空间结构的数据有巨大的优势。在这其中，使用最广泛的是图卷积神经网络[17] [18] (Graph Convolutional Network, GCN)。

GCN 的目的就是利用卷积来提取非欧结构数据的空间信息以及属性信息，能够深入挖掘图模型中的特征规律，根据 Kipf 等人的定义[19]，给定一个无向图 $G=(V,E,A)$ 由节点集合 $V$ 和边集合 $E$ 构成，$A$ 为邻接矩阵，其中 $A\in R^{N\times N}$，输入变量 $X$ 和输出变量 $Y$，图卷积神经网络所采取的处理方式如式(1)，图卷积神经网络的前向传播公式如式(2)，其中 $\tilde{A}=A+I$，$I$ 为大小是 $N\times N$ 的单位矩阵；$\tilde{D}$ 为无向图的度矩阵；$H^{(l)}\in R^{N\times D}$ 表示第 $l$ 层的输出值；$W^{(l)}$ 表示第 $l$ 层的参数值；$\sigma$ 为激活函数。

$$f(X,A)=Y \tag{1}$$

$$H^{(l+1)}=f\left(H^{(l)},A\right)=\sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{2}$$

如图 2 所示，图卷积神经网络的本质就是将各个节点的特征与其有连接的节点的特征信息加权平均后传播到下一层，并随着层数的加深，每个节点可以聚合到的节点信息就更远，从而表示整个图模型的结构特征并进行下一步操作。

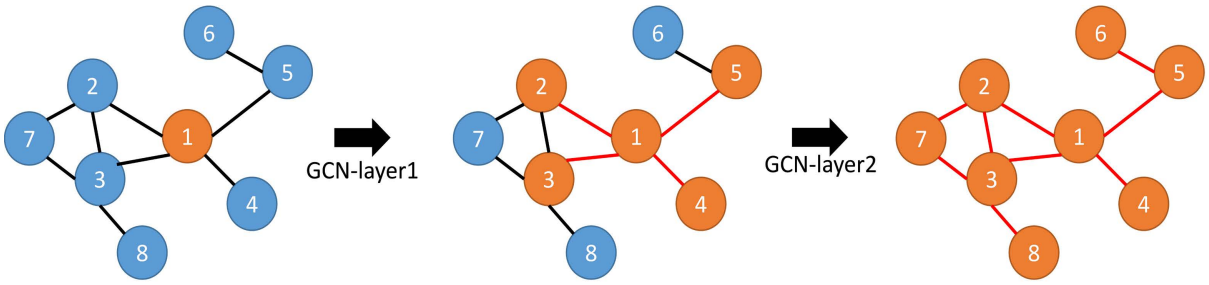


**Figure 2.** Schematic diagram of GCN spatial feature extraction
**图 2.** GCN 空间特征提取示意图

### 2.2.2. 长短期记忆神经网络

基于长短期记忆神经网络(Long Short-Term Memory, LSTM)是循环神经网络(Recurrent Neural Network, RNN)的一种，由于 RNN 在训练时都会将自身提取的信息传递到下一个单元，当该链条的长度累积到一定程度时候便会出现信息的消失。LSTM 的出现为的就是解决长序列训练过程中梯度消失和梯度爆炸的问题，相比起常规的 RNN，LSTM 能够在更长的序列中有更好的效果，LSTM 内部模块示意图如图 3 所示。

LSTM 内部采用了 3 种门控机制删除和增加神经元的信息，其分别为：遗忘门 $f$，输入门 $i$ 和输出门 $o$ ($t$ 表示该 LSTM 单元在 $t$ 时刻)。

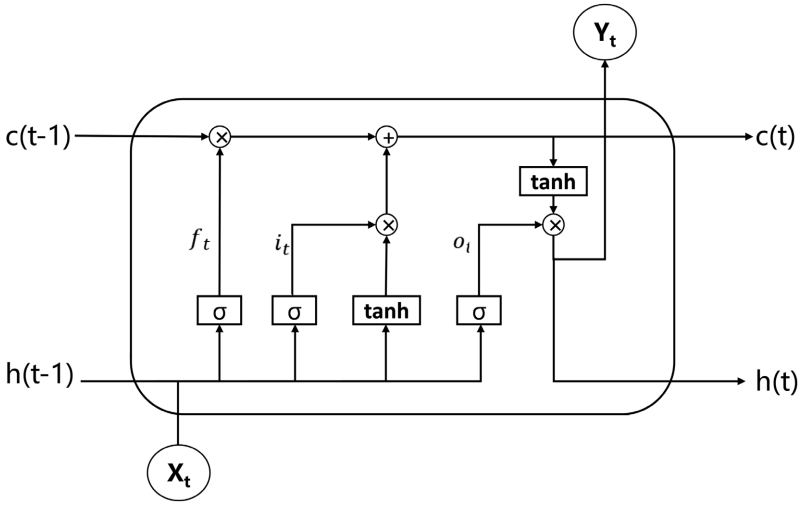


**Figure 3.** Schematic diagram of LSTM internal module
**图 3.** LSTM 内部模块示意图

遗忘门是用来控制当前神经元需要剔除哪些信息，遗忘门的具体计算过程如式(3)：

$$f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right) \tag{3}$$

输入门则是用来控制当前神经元接受的信息有多少可以保留到当前的单元状态中，输入门的状态更新计算过程如下式：

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{4}$$

$$\tilde{C}_t = \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) \tag{5}$$

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{6}$$

输出门控制的是当前神经元能够输出多少信息到下一个时刻，输出门的状态更新计算过程如下式：

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{7}$$

$$h_t = o_t \times \tanh\left(C_t\right) \tag{8}$$

上述式子中，$\sigma$ 表示的是神经网络的激活函数 Sigmoid 函数，会根据输入将变量映射成[0, 1]之间的向量；$\tilde{C}_t$ 表示的是候选神经元中的信息；$W_f$，$W_i$，$W_o$，$W_c$ 表示的是 LSTM 神经元状态更新计算过程中权重；$b_f$，$b_i$，$b_o$，$b_c$ 表示的是 LSTM 神经元状态更新计算过程中偏置。

LSTM 用以捕捉时间信息，就是相比普通的 RNN，LSTM 在面向含有未知长度的时间序列的学习中十分有效，其具有维持长时记忆的能力。因此本文选用 LSTM 作为子模块。

## 2.3. GCN-LSTM 序列重构方法

各级服务器之间不仅有网络层级拓扑结构的交互信息，每个服务器自身的运行状态性能指标也会随着时间的变化而产生变化。因此，整个云计算服务器架构需要抽象成图模型，各个子服务器被抽象成为图的节点，各服务器内部的状态性能指标数据抽象为每个节点的特征向量，各个服务器之间的连接关系抽象为图的边。与此相同，图模型中节点的运行状态除了自身状态以外还受其他节点状态的影响，该图模型的建立为描述云计算服务器的空间特征和属性提供了分析工具。

云计算服务器内部在时域上都经历着使用率、吞吐率、读取速度、写入速度等要素属性的变化，而

这些要素又受到地理环境、气候环境、人类生活、生产规律的影响而体现出来一定的规律性和周期性。因此，通过对过去的云服务器性能指标历史数据进行分析和计算可以对未来云服务器的运行状态进行评估。云服务器的客观运行规律也为时序算法提供了有力的理论支撑。

基于以上分析，本节提出了一种融合了空间信息和时间信息的时间序列重构模型：GCN-LSTM 模型。GCN-LSTM 模型由图卷积神经网络和长短期记忆神经网络两部分组成，其内部模块连接示意图如图 4 所示。一方面，图卷积神经网络对建立的图数据模型提取其内部的信息，解析其拓扑结构，并提取其空间特征。另一方面，将图卷积神经网络对不同时刻的所提取的特征依时间序列的方式输入到长短期记忆神经网络当中学习时间特征。

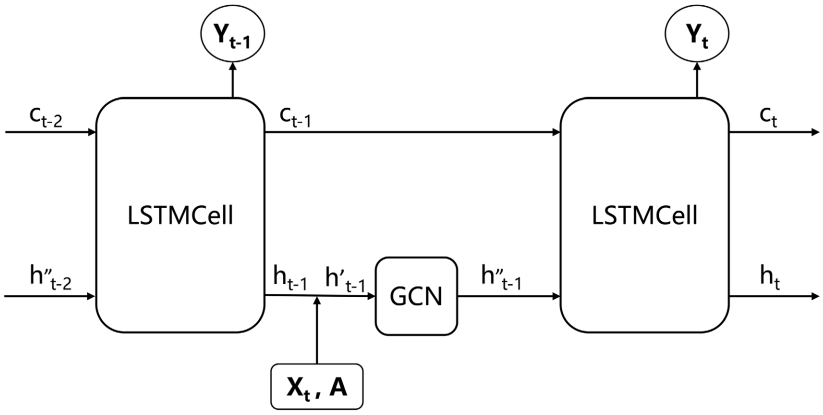


**Figure 4.** Schematic diagram of GCN-LSTM internal module connection
**图 4.** GCN-LSTM 内部模块连接示意图

GCN-LSTM 模型面向云计算服务器中的性能指标数据的操作流程可以简单描述为：通过云服务器历史指标数据时长为 $s$ 的数据 $[X_{t-s},\cdots,X_{t-1}]$ 和图模型的邻接矩阵 $A$，来推测下一个时刻 $t$ 的性能指标，设 $X_t$ 表示$t$时刻云服务器指标数据，$F$ 为 GCN-LSTM 模型处理方法，即可得式(9)：

$$X_t = F\left([X_{t-s},\cdots,X_{t-1}],A\right) \tag{9}$$

将云服务器的历史性能指标序列 $X=[X_1,\cdots,X_n]$ 输入到 GCN-LSTM 模型中，生成重构序列 $X^P=\left[X_1^P,\cdots,X_n^P\right]$，误差序列 $E=[e_1,\cdots,e_n]$，其中 $e_i=X_i-X_i^P$。将集合 $E$ 按照时间顺序分 $E_{train}$ 为和 $E_{test}$ 两部分，并分别作为 COPOD 方法的训练集以及测试集。

## 2.4. COPOD 异常检测方法

异常检测的目的是对不符合预期模式或大多数数据的分布的项目、数据、事件或观测值的识别。从最容易理解的角度出发就是看该数值距离平均值有多远，给定一组一维数据满足高斯分布，若某数据离均值 2 个或者 3 个标准差以外的数值就可以简单地被认为是异常。然而在现实世界中，该方法并不是常常有效的，例如大多数数据并不是一维的，而是有多个维度的。同时这多个维度的数据并非是相互独立的，也就意味着联合分布的建模会变得十分困难，更意味着无法简单地使用 3sigma 准则(又称为拉依达准则)来判定异常，否则会忽略了多个维度数据之间的关联性，使得模型过于盲目和随意。

为了解决该问题，Li 等人提出了的基于 Copula 的异常检测方法[8] (Copula-Based Outlier Detection, COPOD)来估算各数据维度之间联合分布的尾端概率。其中 Copula 函数理论由 Sklar [20]在 1996 年率先提出，可以将一个多元联合分布分解为数个边缘分布以及 Copula 函数，而该函数可以确定变量间的相关性。即设 $F$ 为一个 $N$ 维联合分布函数，其边缘分布为 $F_1,F_2,\cdots,F_n$，则存在一个 Copula 函数使得式(10)成立。

$$F\left(x_1,x_2,\cdots,x_n\right)=\text{Copula}\left(F_1\left(x_1\right),F_2\left(x_2\right),\cdots,F_n\left(x_n\right)\right) \tag{10}$$

若 $F_1\left(x_1\right),F_2\left(x_2\right),\cdots,F_n\left(x_n\right)$ 连续，则存在唯一的 Copula 函数。经过多年发展，Copula 理论已成为了解决高维随机变量联合概率分布问题的有效手段，由于 Copula 函数可以对各边缘分布进行一定程度上的相关性的总结，COPOD 可以对异常是哪些维度造成的提供一些可解释性，例如运维人员可以直接找到造成异常最多的维度，进行深入分析。

COPOD 是一种基于 Copula 的异常检测方法，给定 $d$ 维数据集 $E=\left(e_{1,i},e_{2,i},\cdots,e_{d,i}\right)$，其中 $i$ 表示当前的时间快照，算法分三步进行操作：

1) 面向每一个维度，按照如式(11)以及式(12)使用非参数方法估计左尾部以及右尾部经验累积联合分布(left/right tail Empirical CDF)。

$$\hat{F}_d\left(e\right)=\frac{1}{n}\sum\nolimits_1^n\left(e_i\le e\right) \tag{11}$$

$$\hat{\hat{F}}_d\left(e\right)=\frac{1}{n}\sum\nolimits_1^n\left(-e_i\le -e\right) \tag{12}$$

此时，异常可能出现在分布的左边，也可能出现在分布的右边。不同情况下，使用不同方向的 ECDF 会得到不一样的结果，此时还需要计算偏度系数(Skewness coefficient)。偏度系数是统计数据分布偏斜方向和程度的度量，是统计数据分布非对称(左右不一致)程度的数字特征。当分布左右对称时，偏度系数为 0；当偏度系数大于 0 时，该分布为右偏，异常点更倾向于落在分布的右侧；当偏度系数小于 0 时，该分布左偏，异常点更倾向于落在分布的左侧。偏度系数的计算方式如式(13)。

$$b_i=\frac{\frac{1}{n}\sum\nolimits_{i=1}^n\left(e_i-\overline{e_i}\right)^3}{\sqrt{\frac{1}{n-1}\sum\nolimits_{i=1}^n\left(e_i-\overline{e_i}\right)^2}^3} \tag{13}$$

2) 计算每一个时间快照 $e_i$ 的 empirical copula 观测值，计算过程如式(14)和式(15)所示。于是，可分别得到 $\hat{U}_{d,i}=\hat{F}_d\left(e_{d,i}\right)$ 以及 $\hat{V}_{d,i}=\hat{\hat{F}}_d\left(e_{d,i}\right)$，此时根据 $b_d$ 的值确定偏度系数的 empirical copula 观测值 $\hat{W}_{d,i}$，若 $b_d\ge 0$，则 $\hat{W}_{d,i}=\hat{V}_{d,i}$ 反之 $\hat{W}_{d,i}=\hat{U}_{d,i}$。

$$\left(\hat{U}_{1,i},\hat{U}_{2,i},\cdots,\hat{U}_{d,i}\right)=\left(\hat{F}_1\left(e_{1,i}\right),\hat{F}_2\left(e_{2,i}\right),\cdots,\hat{F}_d\left(e_{d,i}\right)\right) \tag{14}$$

$$\left(\hat{V}_{1,i},\hat{V}_{2,i},\cdots,\hat{V}_{d,i}\right)=\left(\hat{\hat{F}}_1\left(e_{1,i}\right),\hat{\hat{F}}_2\left(e_{2,i}\right),\cdots,\hat{\hat{F}}_d\left(e_{d,i}\right)\right) \tag{15}$$

3) 计算 $e_i$ 的左侧、右侧以及偏度的尾部概率，计算过程如下式：

$$p_l=-\sum\nolimits_{j=1}^d\log\left(\hat{U}_{j,i}\right) \tag{16}$$

$$p_r=-\sum\nolimits_{j=1}^d\log\left(\hat{V}_{j,i}\right) \tag{17}$$

$$p_s=-\sum\nolimits_{j=1}^d\log\left(\hat{W}_{j,i}\right) \tag{18}$$

当尾部概率越小，它的负对数就越大，所以如果一个点的尾部概率小，会更容易被认定为异常值，因此该时间快照下输出异常分数如式(19)：

$$O\left(e_i\right)=\max\left\{p_l,p_r,p_s\right\} \tag{19}$$

阈值可以按照不同领域不同行业自行设置，针对本文方法并按照经验总结，如式(20)所示，异常的阈值

$O_{Thershold}(e_i)$由数据集总体异常率$\alpha$的分位数决定，percentile 函数表示计算分析各时间节点按由大到小排序后异常分数的百分比数值点。当$O(e_i) > O_{Thershold}(e_i)$时，当前时间快照为异常，反之为正常，并将结果输出。

$$O_{Thershold}(e_i) = \text{percentile}(O(e_i), 1-\alpha) \tag{20}$$

## 3. 实验评估与分析

### 3.1. 实验数据集

本实验采用 MBD 数据集，该数据集来自运行大数据批处理系统的环境，其中一个包含 1 个主节点和 4 个从节点。监控并收集的数据包括每个节点的 26 个指标，包括 CPU 空闲、CPUI/O 等待、CPU 软件、CPU 系统、CPU 系统、CPU 用户、每秒等待、磁盘 I/O 进程、磁盘使用百分比、磁盘读取速度、磁盘写入速度、内核熵、负载等，其图模型如图 5 所示。R740-3-1、R740-3-2、R740-3-3、R740-3-4 以及 R740-3-5 表示位于不同地点的云服务器；服务器之间的连线表示其间的交互情况；服务器保存着的过去运行情况的数据指标，上述三点分别可用于构建图模型的节点、边以及属性，以实现图模型内空间特征的提取。

云服务器的指标数据总共收集了 3 天，数据集共有 8640 条数据(观测值)，数据观测时间从 2020.3.25 0:00:00 开始到 2020.3.27 23:59:30，以 30 秒为一个观测单元记录 1 条数据。该数据集被不规则地注入随机参数来模拟应用程序故障，标签值 label 代表异常标记，1 表示异常数据，0 表示正常数据。

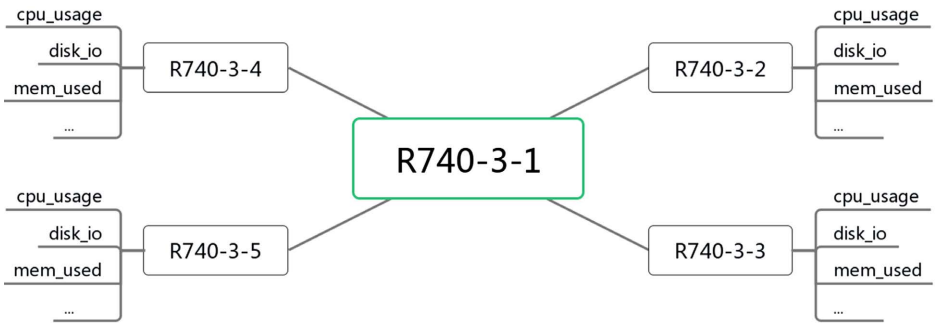


**Figure 5.** Schematic diagram of MBD data set topology
**图 5.** MBD 数据集拓扑结构示意图

### 3.2. 实验评估

本实验使用基于 python3 环境的 Pytorch 开源深度学习框架实现图卷积神经网络和长短期记忆网络进行设计，本文实验采用 PC 机，Microsoft Windows 10 操作系统，16GB 内存，处理器为 AMD Ryzen 5 4600U with Radeon Graphics 2.10 GHz。本文将前两天的数据作为模型的训练集，将第三天的数据作为模型的测试集。

在检测算法中，模型的评估方法使用混淆矩阵及其衍生的各项性能指标，如表 1：

**Table 1.** Confusion matrix
**表 1.** 混淆矩阵

| 混淆矩阵 | | 样本数 | |
|---|---|---|---|
| | | Positive | Negative |
| 检测数 | Positive | TP | FP |
| | Negative | FN | TN |

本文中将使用准确率(Accuracy)、精确率(Precision)、召回率(Recall)、AUC 值作为异常检测评估指标，其计算公式如以下各式：

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{21}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{20}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{21}$$

ROC 曲线(Receiver operating characteristic curve)，即受试者工作特征曲线，主要用来评价对二分类算法的效果，以及寻找最佳的指标临界值使得分类效果最好。该曲线由真阳性率(TPR, True positive rate)为横坐标，假阳性概率(FPR, False positive rate)为纵坐标共同绘制而成，曲线越往左上角则效果越好。

AUC 值(Area Under ROC Cure)被定义为 ROC 曲线下与坐标轴围成的面积，取值范围一般在 0.5 和 1 之间，这是因为 ROC 曲线一般在 $y = x$ 上方，其评价标准为：当 AUC 值越接近 1.0，该异常检测方法真实性越高。

### 3.3. 实验结果与分析

实验中采用了多种在异常检测领域有所建树的方法作为对比实验，包括基于距离的 KNN 方法，基于密度的 LOF [21]、COF [22]方法，基于树的 iForest 方法，基于概率的 ECOD 方法以及基于深度学习的 AutoEncoder 方法。同时，我们还对基模型 LSTM 的方法进行了实验对照，以便于对模型效果更好的评估。以下为各基准方法的介绍：

1) KNN (K-Nearest Neighbor)：基于距离的方法，对数据集分布、标签以及类型要求不高，K 取值为 5；

2) LOF (Local outlier factor)：基于密度的方法，若某点密度越低越可能被认定是异常点；

3) COF (Connectivity based Outlier Factor)：基于局部密度的方法，通过最短路径求出局部密度；

4) iForest (Isolation Forest)：采用构造多个决策树的方式进行异常检测，通过树的高度确定异常值；

5) ECOD (Empirical-Cumulative-distribution-based Outlier Detection)：通过计算经验累积分布确定落在尾部的异常值；

6) AutoEncoder：利用编码器和解码器的之间的序列重构误差确定异常，该方法异常点服从不同的分布，从而无法将异常数据较好还原。

实验采用 3.1 的数据集进行，分别对各类基准方法进行对比，评估方式如(3.2)各式，结果如下方图 6、图 7、图 8，表 2 所示：

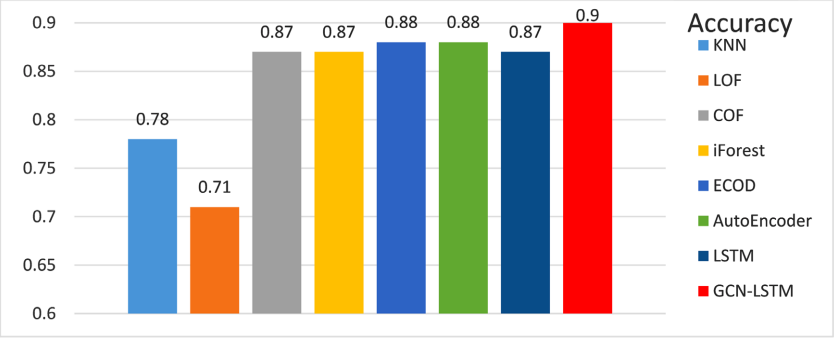


**Figure 6.** Experimental results of Accuracy
**图 6.** 准确率(Accuracy)实验结果

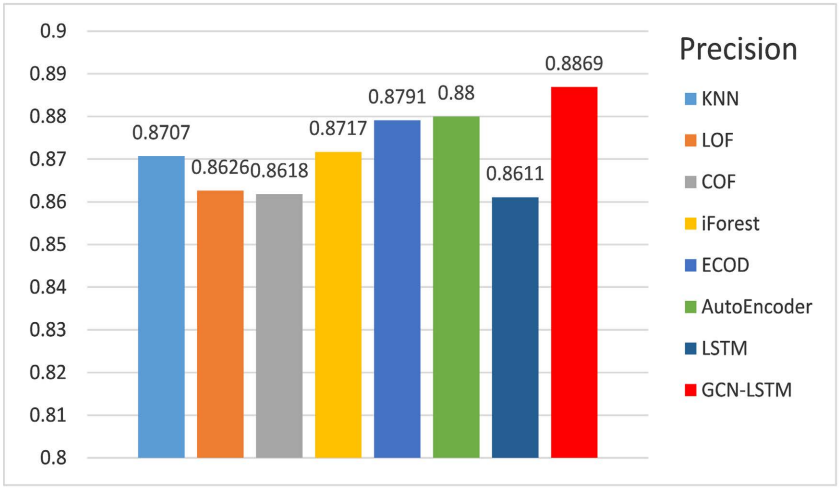


**Figure 7.** Experimental results of Precision
**图 7.** 精确率(Precision)实验结果

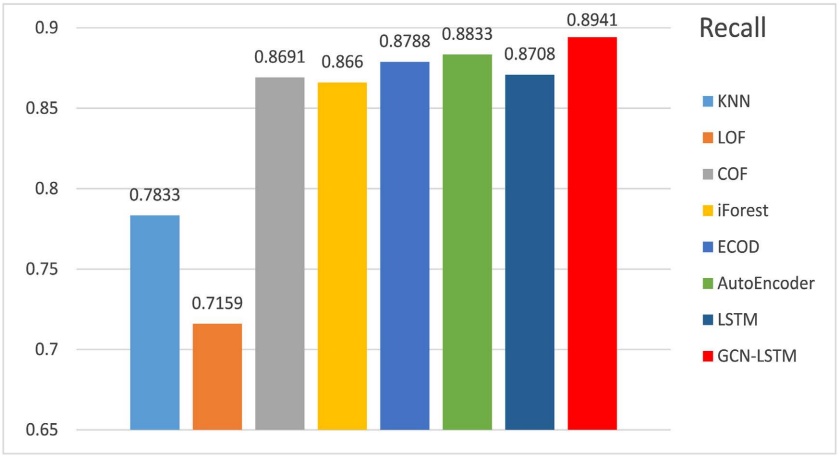


**Figure 8.** Experimental results of Recall
**图 8.** 召回率(Recall)实验结果

**Table 2.** Comparison of AUC values
**表 2.** AUC 值对比

| 方法 | KNN | LOF | COF | iForest | ECOD | AutoEncoder | LSTM | **GCN-LSTM** |
|---|---|---|---|---|---|---|---|---|
| AUC | 0.583 | 0.4993 | 0.5032 | 0.6272 | 0.6745 | 0.6641 | 0.7127 | **0.7329** |

从实验结果可以得出，本文提出的 GCN-LSTM-COPOD 异常检测方法在 MBD 数据集上，准确率(Accuracy)、精确率(Precision)、召回率(Recall)、AUC 值等都优于上述方法。除了准确率、精确率和召回率都有一定程度的提高以外，相比较起其他方法，本文提出的方法可以获得更好 AUC 值，相比起其他方法提高了大约 27%~40%，意味着本方法在面向异常的有效程度和检测价值得到了提高。如图 9 所示，ROC 曲线可以很好地展现出各类方法对异常的检测能力，展示本文提出的于图卷积神经网络的云服务时序异常检测算法优于其他类型方法，试验准确性高。另外，相比起基模型 LSTM，GCN-LSTM 在获取空间信息上有着更好的适用性，使得模型在训练时可以学习到更加充足信息，在序列重构时，可以将空间信息和时间信息有效重现，此时的构建误差可以获得更好的效果。

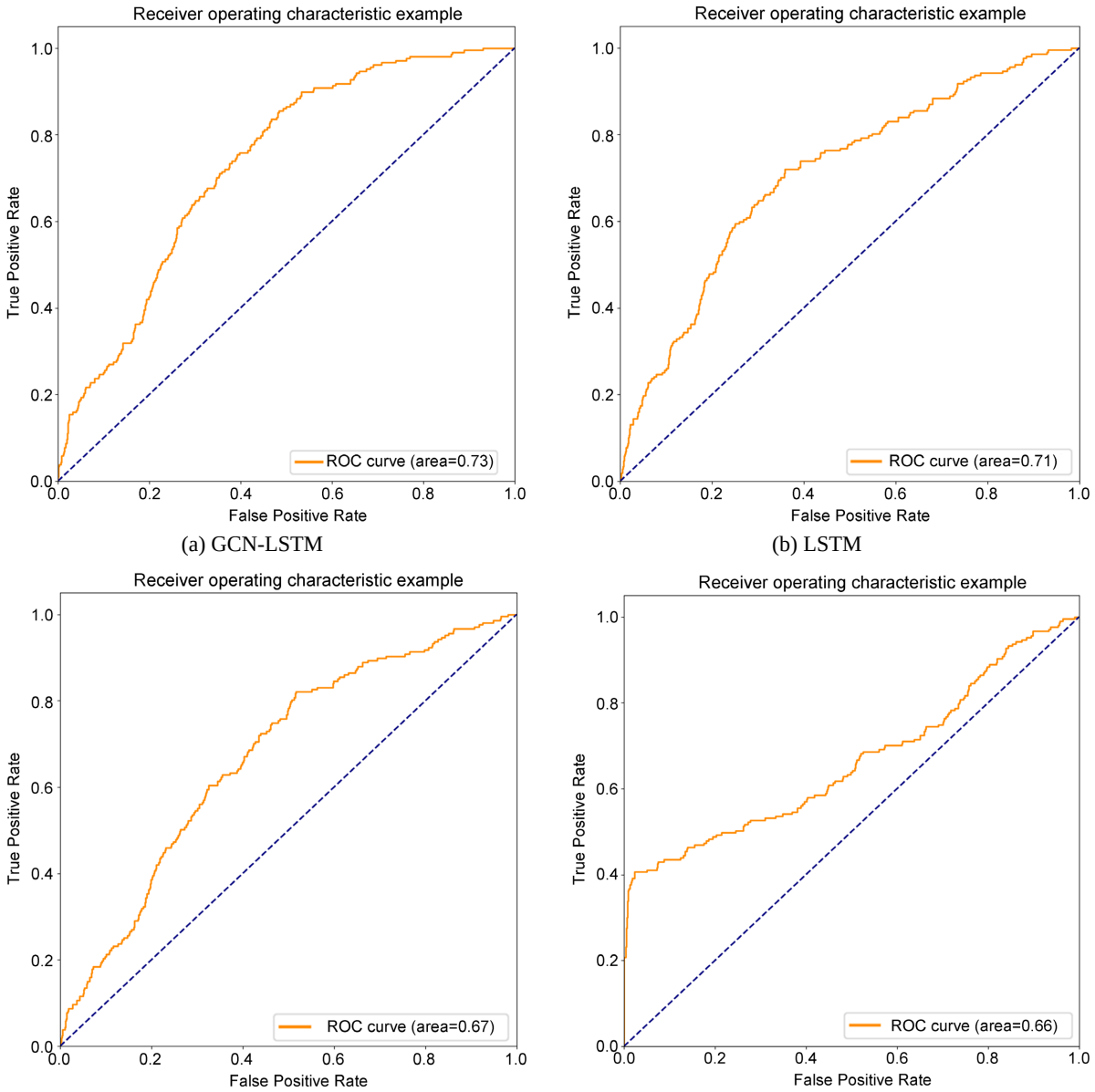


**Figure 9.** ROC curves of each method (excerpt ranking the top four)
**图 9.** 各方法 ROC 曲线(节选排名前四)

在可解释性方面，本文使用的异常检测方法可以将一个多元联合分布分解为数个边缘分布以及 Copula 函数，此时可以针对不同的边缘分布进行分析，查看其哪个分布的异常得分对异常结果有最大的贡献，从而让该方法可解释。如图 10 所示，这是 MBD 数据集中处于异常状态时最后 20 个维度的异常得分示意图，其中数字 14 的维度拥有远超其他维度的异常得分，该维度表示的数据为建立 TCP 连接的等待时间，因此，意味着此时的异常极有可能是 TCP 建立时引发的。

## 4. 结论

本文提出了一种基于 GCN-LSTM 的云计算服务异常检测算法，面向带有时空特征数据的云服务器异常检测问题，首先采用图卷积神经网络提取云服务器之间的空间特征，然后用不同时刻提取到的时间信

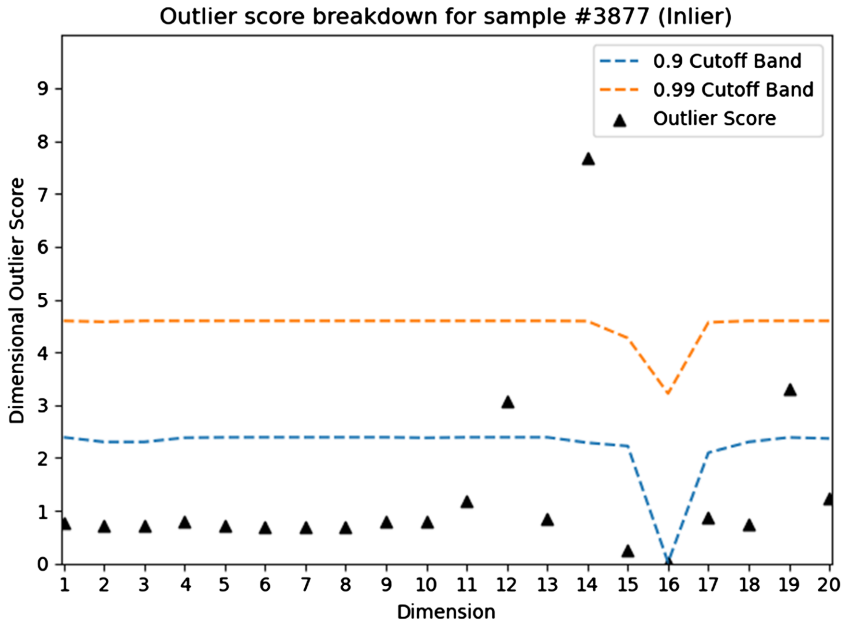


**Figure 10.** Schematic diagram of dimension anomaly score
**图 10.** 维度异常得分示意图

息输入到长短期记忆网络中对云服务器性能指标进行重构，利用 COPOD 对重构误差进行异常检测。实验结果表明本文提出的异常检测算法要比其他类型异常检测算法更优异，同时减少了阈值的选择和计算，并为造成异常的原因带来了可解释性。未来，将对该模型进一步优化，研究更优秀的数据异常检测的处理方法，提高模型的各项效果以及指标。

## 参考文献

[1] González, S., Sedano J., Herrero, Á., Baruque, B. and Corchado, E. (2011) Testing Ensembles for Intrusion Detection: On the Identification of Mutated Network Scans. In: Herrero, Á. and Corchado, E., Eds., *Computational Intelligence in Security for Information System*, Springer, Berlin, Heidelberg, 109-117. https://doi.org/10.1007/978-3-642-21323-6_14

[2] Atkinson, A.C. (2018) Review: Identification of Outliers. by D. M. Hawkins. *Biometrics*, **37**, 860-861. https://doi.org/10.2307/2530182

[3] Cohen, W.W. (1995) Fast Effective Rule Induction. *Proceedings of the* 12*th International Conference on Machine Learning*, Tahoe City, 9-12 July 1995, 115-123. https://doi.org/10.1016/B978-1-55860-377-6.50023-2

[4] Quinlan, J. (2014) RC4.5: Programs for Machine Learning. Morgan Kaufmann, Cambridge.

[5] Barnett, V., Lewis, T. and Abeles, F. (1994) Outliers in Statistical Data. 3rd Edition, Wiley, Chichester.

[6] Ramaswamy, S., Rastogi, R. and Shim, K. (2000) Efficient Algorithms for Mining Outliers from Large Data Sets. *ACM SIGMOD Record*, **29**, 427-438. https://doi.org/10.1145/335191.335437

[7] Breunig, M.M., Kriegel, H.P., Ng, R.T. and Sander, J. (1999) Optics-of: Identifying Local Outliers. *Proceedings of European Conference on Principles of Data Mining and Knowledge Discovery*, Prague, 15-18 September 1999, 262-270. https://doi.org/10.1007/978-3-540-48247-5_28

[8] Li, Z., Zhao, Y., Botta, N., Ionescu, C. and Hu, X. (2020) COPOD: Copula-Based Outlier Detection. 2020 *IEEE International Conference on Data Mining*, Sorrento, 17-20 November 2020, 1118-1123. https://doi.org/10.1109/ICDM50108.2020.00135

[9] Li, Z., Zhao, Y., Hu, X., Botta, N., Ionescu, C. and Chen, G.H. (2022) ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions. *IEEE Transactions on Knowledge and Data Engineering*. arXiv preprint arXiv:2201.00382. https://doi.org/10.1109/TKDE.2022.3159580

[10] Fei, T.L., Kai, M.T. and Zhou, Z.H. (2008) Isolation Forest. 2008 *IEEE International Conference on Data Mining*, Pisa, 19 January 2009, 413-422.

[11] Pascal, V., Larochelle, H., Bengio, Y. and Manzagol, P.-A. (2008) Extracting and Composing Robust Features with Denoising Autoencoders. *Proceedings of the* 25*th International Conference on Machine Learning*, Helsinki, 5-9 July 2008, 1096-1103. https://doi.org/10.1145/1390156.1390294

[12] Wang, H., Zhou, C., Jia, W., Dang, W., Zhu, X. and Wang, J. (2018) Deep Structure Learning for Fraud Detection. 2018 *IEEE International Conference on Data Mining* (*ICDM*), Singapore, 17-20 November 2018, 567-576. https://doi.org/10.1109/ICDM.2018.00072

[13] Zhang, M., Li, T., Shi, H., Li, Y. and Hui, P. (2019) A Decomposition Approach for Urban Anomaly Detection across Spatiotemporal Data. 28*th International Joint Conference on Artificial Intelligence* (*IJCAI*-19), Macao (China), 10-19 August 2019, 6043-6049. https://doi.org/10.24963/ijcai.2019/837

[14] Zheng, P., Yuan, S., Wu, X., Li, J. and Lu, A. (2018) One-Class Adversarial Nets for Fraud Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, **33**, 1286-1293. https://doi.org/10.1609/aaai.v33i01.33011286

[15] Du, M., Li, F., Zheng, G. and Srikumar, V. (2017) DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Dallas, 30 October-3 November 2017, 1285-1298. https://doi.org/10.1145/3133956.3134015

[16] Meng, W., Liu, Y., Zhu, Y., Zhang, S., Pei, D., Liu, Y., et al. (2020) LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs. *Proceedings of the* 28*th International Joint Conference on Artificial Intelligence*, Macao (China), 10-16 August 2019, 4739-4745. https://doi.org/10.24963/ijcai.2019/658

[17] Defferrard, M., Bresson, X. and Vandergheynst, P. (2016) Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. arXiv:1606.09375.

[18] Henaff, M., Bruna, J. and Lecun, Y. (2015) Deep Convolutional Networks on Graph-Structured Data. arXiv:1506.05163.

[19] Kip, F.T.N. and Welling, M. (2016) Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907.

[20] Sklar, A. (1996) Random Variables, Distribution Functions, and Copulas—A Personal Look Backward and Forward. Institute of Mathematical Statistics, Beachwood. https://doi.org/10.1214/lnms/1215452606

[21] Breunig, M.M., Kriegel, H.P., Ng, R.T. and Sander, J. (2000) LOF: Identifying Density-Based Local Outliers. *Proceedings of the* 2000 *ACM SIGMOD international conference on Management of Data*, Dallas, 15-18 May 2000, 93-104. https://doi.org/10.1145/342009.335388

[22] Tang, J., Chen, Z., Fu, A. and Cheung, D.W.-L. (2002) Enhancing Effectiveness of Outlier Detections for Low Density Patterns. Advances in Knowledge Discovery and Data Mining. *Proceedings of the* 6*th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, Taipei, 6-8 May 2002, 535-548.