**Computer Science and Application** 计算机科学与应用, 2022, 12(3), 610-621
Published Online March 2022 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2022.123062

Hans汉斯

# 提取时空特征的无监督时间序列异常检测

**姜　昊[1,2]，郭文明[1,2,3]**

[1]北京邮电大学计算机学院(国家示范性软件学院)，北京
[2]可信分布式计算与服务教育部重点实验室，北京
[3]新疆工程学院信息工程学院，新疆　乌鲁木齐

收稿日期：2022年2月15日；录用日期：2022年3月11日；发布日期：2022年3月21日

## 摘　要

**为解决web应用程序及服务中的异常自动发现问题，针对互联网运维中常用监控指标的异常检测提出了一种基于深度挖掘时空特征的时间序列异常检测模型。考虑到web服务场景中异常发现的时效性要求，模型加强了编码器对空间信息的建模能力。模型使用基于VGG+Bi-LSTM的编码器用于挖掘时序数据中时空特征，使用全连接神经网络与Bi-LSTM构成的解码器重构输入数据。异常判定模块基于重构结果与原始输入的偏离程度计算异常得分与发现异常。这是一种无监督、不需要对异常数据进行分布假设，是纯数据驱动的方法。基于重构输入数据的方式使其拥有发现不可预见错误的能力。充分挖掘时空特征使模型能够及时准确地发现异常。实验结果表明，模型相较于目前常用的时间序列异常检测模型具有更高的异常识别准确率。在公开数据集上的实验结果表明，模型召回率提高6%，F1-score提高0.04。**

## 关键词

**AIOps，异常检测，时间序列，时空特征**

# Time Series Anomaly Detection Model Based on Deep Mining of Spatio-Temporal Features

**Hao Jiang[1,2], Wenming Guo[1,2,3]**

[1]School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing
[2]Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education, Beijing
[3]School of Information Engineering, Xinjiang Institute of Engineering, Urumqi Xinjiang

Received: Feb. 15th, 2022; accepted: Mar. 11th, 2022; published: Mar. 21st, 2022

文章引用: 姜昊, 郭文明. 提取时空特征的无监督时间序列异常检测[J]. 计算机科学与应用, 2022, 12(3): 610-621.
DOI: 10.12677/csa.2022.123062

## Abstract

**To solve the problem of automatic anomaly detection in web applications and services, a time series anomaly detection model based on deep mining of spatio-temporal features is proposed for the anomaly detection of commonly used monitoring indicators in Internet operation and maintenance. Taking into account the timeliness requirements of abnormal discovery in web service scenarios, the model strengthens the encoder's ability to model spatial information. The model uses an encoder based on VGG+Bi-LSTM to mine spatio-temporal features in time series data, and uses a decoder composed of a fully connected neural network and Bi-LSTM to reconstruct the input data. The abnormality determination module calculates the abnormality score and finds the abnormality based on the degree of deviation between the reconstruction result and the original input. This is an unsupervised method that makes no distribution assumption on the abnormal data, and is a purely data-driven method. Based on the way of reconstructing the input data, it has the ability to find unforeseen errors. Fully excavate spatio-temporal features so that the model can find anomalies in a timely and accurate manner. The experimental results show that the model in this paper has higher recognition accuracy than the current time series anomaly detection models. The experimental results on the public data set show that the model recall is increased by 6%, and the F1-score is increased by 0.04.**

## Keywords

**AIOps, Abnormal Detection, Time Series, Spatio-Temporal Features**



## 1. 引言

互联网公司为确保提供的服务稳定运行，需要密切监控 web 应用程序的各种 KPI [1]，从而及时发现服务的异常，给相关人员发送告警信息。智能运维(AIOps, artificial intelligence for IT operations)技术旨在通过人工智能和机器学习技术使 web 服务能够高效地构建和大规模运行。AIOps 可以帮助实现更高的服务质量和客户满意度、提高工程生产力和降低成本[2]。时间序列异常检测算法常被用来进行指标序列的异常自动发现，是 AIOps 技术中一种重要的算法。常用的监控指标包括业务指标(如请求数、响应率、平均响应时间、调用成功率等)与机器指标(CPU 使用率、内存/磁盘占有率等)。这些 KPI 的波动特征具有多样性。KPI 波动有表现为周期型的，有表现为稳定型的，也有表现为不稳定的，持续波动型的。

随着微服务架构的发展和广泛应用，web 服务的规模变得更大，业务系统的复杂性也越来越高。这使得监控指标流量巨大、异常比例低、异常种类多样。以上特点对于互联网运维场景下的 KPI 异常检测算法提出了如下要求：异常检测算法应具备发现不可预见错误的能力；需要使用无监督算法，数据量巨大使得标注成本成倍提高，数据特征的频繁变化使标注价值降低；对于异常发现的时效性具有较高要求，应在引起用户感知之前发现异常，不然异常发现将失去意义。

为解决上述问题，本文提出了一种构建自动编码器重构输入数据的时间序列异常检测算法。该方法对于输入的时间序列数据进行重构，高重构误差可能意味着异常。这是因为该方法使用正常时间序列训练模型，与可预见的正常数据相比模型将无法很好地重建异常数据的不可预见的模式。这使得该方法拥

有发现不可预见错误的能力。模型使用滑动窗口的方式从时序数据中提取固定长度的输入序列。编码器使用 VGG (Visual Geometry Group)模块来提取时间序列数据中局部的空间特征、Bi-LSTM (Bi-directional Long Short-Term Memory)提取数据中的时间依赖关系。由全连接神经网络与 Bi-LSTM 构成的解码器重构输入的时间序列数据。使用重构后的 KPI 与原始 KPI 的残差用于检测异常。

对于异常发现的时效性，本文在实验结果分析中增加了如下设计：对于时间序列中的异常区间，如果异常检测算法在异常区间开始后的 T 个时间点内检测到了该异常区间，则认为整个区间都被检测成功，否则则视为整个区间都没有被检测成功。

本文的工作主要有三点贡献：

1) 使用无监督的异常检测方法使得该方法对于互联网运维领域的应用价值较高；

2) 提出了一种基于深度挖掘时空特征的异常检测算法使得模型能够及时地发现数据中的异常模式，且能够识别不可预见的异常；

3) 与目前常用的其他方法对比，本文提出的方法在识别准确率上有较大提升。在公开数据集上的实验结果表明，模型召回率提高 6%，F1-score 提高 0.04。

## 2. 相关工作

互联网运维的指标监控中，KPI 以数据流的形式实时的流经异常检测系统。在[3] [4]中，Sadik 详细介绍了流环境的特殊性以及在这种情况下异常值检测的困难。主要限制如下：数据不能扫描两次，算法的时间复杂度不能过高。这使得一些常用的异常检测方法不能被应用在数据量巨大的数据流上。如基于距离、基于最近邻或基于聚类的方法。

EGADS [5]和 Op-prentice [6]作为有监督的集成方法，使用用户反馈作为标签并使用传统检测器输出的异常评分作为特征来训练异常分类器。ALSR [7]使用标签筛选模型来更细粒度地分析和利用 KPI 的连续异常区间，并基于随机森林的再学习模型对检测到的异常域内的真/假阳性点进行重新分类，从而有效减少假阳性点的数量。EGADS、Opprentice 和 ALSR 均显示出令人鼓舞的结果，但它们严重依赖于良好的标签，这在大规模应用中通常不可行。Agarwal [8]使用高斯模型作为先验假设来检测多维数组中的异常。Eskin [9]提出了一种更通用的混合方法，模型基于多数分布和异常分布。然而这些基于统计的异常检测方法存在一个固有问题，即依赖于 Chandola 等在[10]中给出的假设：正常数据实例发生在随机模型的高概率区域，而异常发生在低概率区域。

Hawkins [11]最早应用了自动编码器的思想，使用具有三层隐藏层的多层感知机神经网络，将输入变量作为输出标记训练模型，以重构误差为度量进行离群值检测。Sakurada [12]提出了自动编码器适用于时间序列数据得异常检测。证明了自动编码器可以正确学习时间序列数据的正常状态，当异常数据输入时，可以不同程度的激活。Chauhan [13]使用由 LSTM 单元组成的深度循环神经网络构建了一个心电图信号的预测模型，利用实际输入和预测结果的差异程度判定异常。Malhotra [14]提出了一种基于 LSTM 的编码解码器模型。通过监控传感器捕捉的代表机械行为和健康信息的时间序列数据来发现机械设备的运行异常。许浩文等[15]提出了一种基于变分自动编码器 VAE (Variational Auto-Encoder)的方法对周期性变化的 KPI 进行异常检测。TAnoGan [16]是一种基于生成对抗网络的异常检测方法，该方法可用于在少量数据点可用时检测时间序列中的异常。Hundman [17]等使用 LSTM 网络对多元 KPI 值进行预测，被用来发现航天设备的运行异常。亚苏等[18]提出了多元时间序列异常检测的随机递归神经网络，用于航天器，引擎等工业设备。工业设备的异常检测关注的是实体级别的异常检测，一个实体(如车辆的一个部件)往往有多个 KPI 监控它的运行，然而相关人员更关注是哪个设备发生了问题而不是哪个指标。戚琦等[19]通过 Gat-ed-CNN 及 GCN 组成的编码器来提取网络设备 KPI 的数据特征，随后使用基于卷积网络的解码器重

构原始 KPI。但是以上方法均未考虑挖掘时间序列数据上的局部空间特征，这使得上述方法均需要输入的序列足够长时才能够发现数据中的异常模式。当输入数据序列长度不足时，模型将无法提取数据的波动趋势，此时模型对于异常的辨别能力较弱。当异常发现的时效性要求较高时，上述方法不能胜任。

## 3. 研究方法

### 3.1. 模型设计

模型如图 1 所示，分为三个模块：预处理模块，编码－解码器模块与异常检测模块。预处理模块负责剔除无效数据，缺失值填补并对数据进行归一化处理。编码－解码器模块挖掘数据时空信息并根据挖掘的信息重构数据。异常检测模块根据重构数据与原始输入数据的偏理程度计算异常得分。

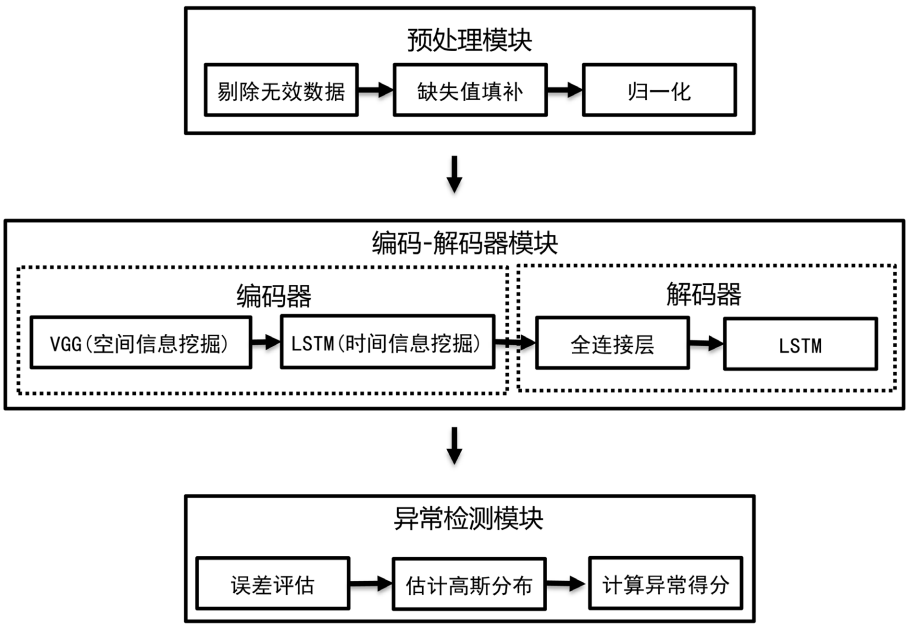


**Figure 1.** Model design
**图 1.** 模型设计

自动编码器由顺序连接的编码器和解码器网络组成。它将解码器的目标设置为等于编码器的输入。编码器学习输入的压缩表示，即数据特征或潜在分布。解码器从压缩后的表示中重建目标。重构误差既是原始输入数据和重构数据之间的差异。在训练期间，自动编码器将最小化重构误差作为目标函数。在[11] [12]中证明了这种方法在异常检测领域中的有效性。这种结合深度学习的重构输入数据的方法相比于之前的异常检测方法[5] [6] [7] [8] [9]的优势是：是一种无监督方法，不需要有标注的数据；不需要先验的异常知识；可以发现数据中不可预见的错误。

### 3.2. 空间信息挖掘

预处理模块使用滑动窗口的方法从时序数据中提取固定长度的时间序列作为模型输入。为能够在异常区间的前 T 个时间点内及时的检测出异常，编码器通过挖掘空间特征的方式增加模型对短序列上异常的辨别能力。时间序列数据是一种符合欧几里得空间的一维线性数据，可以使用 CNN (Convolutional Neural Network，卷积神经网络)提取空间特征。

卷积神经网络使用由卷积层和池化层构成的特征抽取器来进行自动特征提取，最常见的是使用二维

的卷积层来处理图像数据。CNN 中的卷积本质上就是利用一个共享参数的过滤器(kernel)，通过计算中心点以及相邻像素点的加权和来构成特征图，实现空间特征的提取，其中加权系数就是卷积核的权重系数。就像使用二维卷积核提取图像特征那样，使用一维卷积核的 CNN 也可以自动提取 KPI 数据中的局部空间特征，如图 2 中红框圈出的部分：

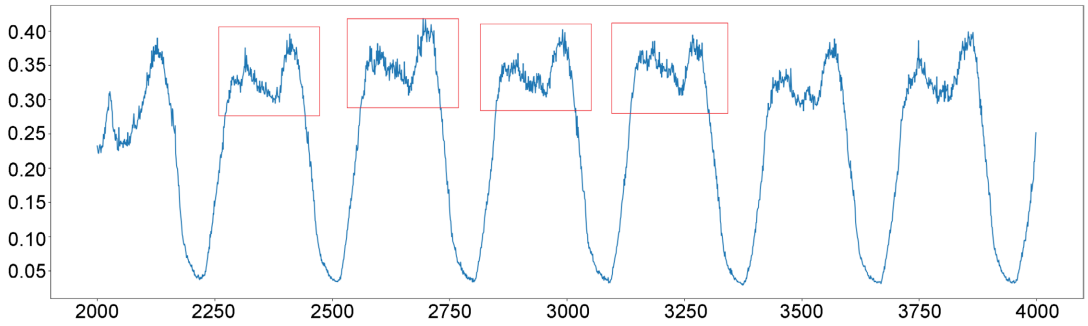


**Figure 2.** Local spatial features in KPI
**图 2.** KPI 中的局部空间特征

为应对时序数据中的复杂波动特征，模型使用 VGG 网络用于提取空间特征中的复杂模式。VGG 网络中连续使用 3 个相同的填充为 1，窗口形状为 3 * 1 的卷积层后接上一个步幅为 2、窗口形状为 2 * 1 的最大池化层。卷积层保持输入序列的长度不变，而池化层则对其减半。VGG 网络在保证具有相同感受野的条件下通过提升网络深度充分提取了 KPI 序列中的空间特征，并建立了长度为输入序列长度 1/8 的低维表征，降低了时间信息挖掘层的输入规模。

### 3.3. 时间信息挖掘

RNN (Recurrent Neural Network，循环神经网络)与早期的密集深度神经网络相比，它并非刚性地记忆所有固定长度的序列，而是通过隐藏状态来存储之前时间步的信息，降低了网络中参数数量的同时还保持了较好的对时间信息的建模能力。LSTM (Long Short-Term Memory，长短期记忆)则在 RNN 的基础上增加了门控机制。由于引入了以上下文为条件的加权自循环，LSTM 已被证明可以提高保持长期依赖记忆的能力。它们能够处理高复杂性、时间或顺序数据的能力确保了它们在自然语言处理(NLP)、文本分类、语音识别和时间序列预测等领域的广泛应用。[13] [14] [17]展示了 LSTM 在时间序列异常检测领域中的应用。

为充分提取时序数据中的时间特征，模型参考了 Bi-LSTM 的设计思想，在解码器中使用隐藏层单元的逆序来重构输入数据。LSTM 编码器学习输入时间序列的固定长度向量表示，LSTM 解码器使用此表示的当前隐藏层状态和前一时间步预测的值重建时间序列。给定输入序列：

$$X = \{x_1, x_2, \cdots, x_L\} \tag{1}$$

使用 $h_E^i$ 表示编码器在 $t_i$ 时刻的隐藏状态，其中 $i \in \{1, 2, \cdots, L\}$。编码器和解码器联合训练以逆序重建时间序列。编码器的最终状态 $h_E^L$ 用作解码器的初始状态。LSTM 解码器层之上的线性层用于重构目标。在训练过程中，解码器使用 $x_i$ 作为输入获得状态 $h_D^{i-1}$，然后重构目标 $x_{i-1}$ 对应的 $x'_{i-1}$。在计算过程中，将重构值 $x'_i$ 输入到解码器以获得 $h_D^{i-1}$ 并重构 $x'_{i-1}$。该模型经过训练以最小化式(2)。其中 $\boldsymbol{S_N}$ 是一组正常的训练序列。

$$\sum_{X \in s_N} \sum_{i=1}^{L} \left\| x_i - x'_i \right\|^2 \tag{2}$$

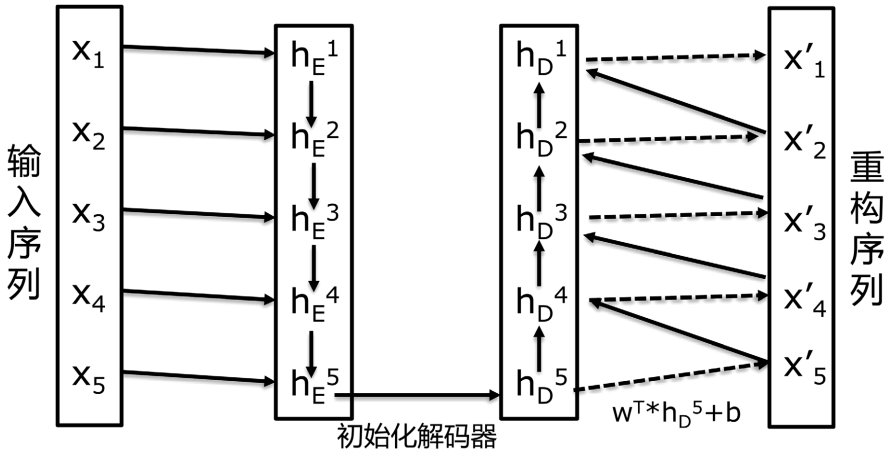


**Figure 3.** Time information mining module
**图 3.** 时间信息挖掘模块

图 3 描述了 LSTM 编码器－解码器重建模型中 $L = 5$ 的序列的计算步骤。$t_i$ 时刻的编码器隐藏层状态 $h_E^i$ 来源于 $t_i$ 时刻的输入值 $x_i$ 和 $t_{i-1}$ 时刻的编码器隐藏层状态 $h_E^{i-1}$。编码器的最终状态 $h_E^5$ 作为解码器的初始状态 $h_D^5$，即 $h_D^5 = h_E^5$。解码器的顶部使用大小为 $c$*1 的权重矩阵 $w$ 和偏置向量为 $b$ ($b \in R_m$)的线性层来计算式 $x_5'$。如式(3)所示：

$$x_5' = w^{\mathrm{T}} * h_D^5 + b \tag{3}$$

其中 $c$ 为LSTM编码器中隐藏层单元的数量。解码器使用 $h_D^i$ 来重构 $x_i'$ 的值和获取下一个隐藏层状态 $h_D^{i-1}$ 的值。

### 3.4. 异常评分与判定

将正常时间序列划分为四个集合：训练集 $\boldsymbol{S_N}$，验证集 $\boldsymbol{V_{N1}}$，$\boldsymbol{V_{N2}}$ 和测试集 $\boldsymbol{T_N}$，异常时间序列划分为：验证集 $\boldsymbol{V_A}$，测试集 $\boldsymbol{T_A}$。使用 $\boldsymbol{S_N}$ 训练模型，$loss_1$ 为一步重构的模型输出与 $\boldsymbol{V_{N1}}$ 的均方损失，$loss_2$ 为多步重构的模型输出与 $\boldsymbol{V_{N1}}$ 的均方损失，$loss_3$ 为一步重构与多步重构的均方损失。损失函数 $loss$ 如式(4)所示：

$$loss = loss_1 + loss_2 + loss_3 \tag{4}$$

然后使用得到的模型在验证集 $\boldsymbol{V_{N2}} + \boldsymbol{V_A}$ 上计算每个点的异常得分。对于集合中*(i-1)*时刻 KPI 的值对 $i$ 时刻 KPI 的值的重构误差为式(5)

$$e_{i(i-1)} = \left| x_i - x'_{i(i-1)} \right| \tag{5}$$

其中 $x_i$ 为 KPI 序列中 $i$ 时刻的实际值，$x'_{i(i-1)}$ 为模型以 $x_{i-1}$ 为输入对 $x_i$ 的重建值。假设 *prediction_window_size* 的值为 $w$，对于输入值 $x_i$ 的重建误差向量 $\boldsymbol{error}_i$ 是一个长度为 $w$ 的向量：

$$\boldsymbol{error}_i = \left\{ e_{i(i-w)}, \cdots, e_{i(i-2)}, e_{i(i-1)} \right\} \tag{6}$$

基于最大似然评估 $N(\mu, \Sigma)$ 估计每个时刻 KPI 值的正态分布，得到每个点的异常得分为：

$$score\ a(i) = (error_i - \mu)^{\mathrm{T}} \Sigma^{-1} (error_i - \mu) \tag{7}$$

给定阈值 $\tau$，如果 $x_i$ 点的异常得分 $score\ a(i) > \tau$ 则判定 $x_i$ 为异常点，否则判定 $x_i$ 为正常点。在验证集 $\boldsymbol{V_{N2}} + \boldsymbol{V_A}$ 上通过最大化 F1-score 来确定判定异常的阈值 $\tau$。最后在测试集 $\boldsymbol{T_N} + \boldsymbol{T_A}$ 上得出整个异常检测模型的准确率 P，回归率 R 和 F1-score。

## 4. 实验

### 4.1. 实验数据集与数据预处理

实验将在两个不同的数据集上对模型进行验证。其中数据集 1 来源为某金融公司的内部系统，并结合其 error 日志，异常告警信息，错误事件记录对验证集进行异常标注。数据集 2 为公开数据集，是 2018 年 Aiops 挑战赛决赛使用的 KPI 数据集，其中包含了 29 个 KPI 数据，序列长度在 8000~150,000 不等。

数据预处理主要涉及缺失值填补，剔除无效、重复数据，平滑噪声数据及规范化处理各个特征数据及非数值数据。最终做数据归一化处理，将数据按照一定比例进行缩放，使数据映射到特定的空间里面，去除数据的单位限制，便于不同单位或量级的指标能够进行比较和加权。在数据集 1 上使用的归一化方法有线性函数归一化(Min-Max scaling)、0 均值标准化(Z-score standardization)、均值绝对偏差标准化、对数变换、sigmoid 函数等。

### 4.2. 实验结果及分析

实验包含三个部分。1) 验证性实验：验证模型的有效性。2) 对比实验：与目前常用的各类异常检测方法进行对比，探究模型效果及优缺点。3) 参数影响：探究不同参数下模型效果的差异。

#### 4.2.1. 验证实验

模型在数据集 2 上的实验结果如表 1：

**Table 1.** Experimental results of the model on data set 2
**表 1.** 模型在数据集 2 上的实验结果

| 精确率(P) | 召回率(R) | F1-score |
|---|---|---|
| 0.74 | 0.91 | 0.82 |

对于数据集中周期型波动的 KPI，模型能够充分挖掘其时空信息学得其数据分布。当异常数据输入时，由于数据分布发生变化，模型不能重构出异常数据，从而能够有效的发现周期型 KPI 中的异常数据。如图 4、图 5 所示，对于序列中的异常部分，模型能够在异常区间的开始部分及时给出较高的异常得分。

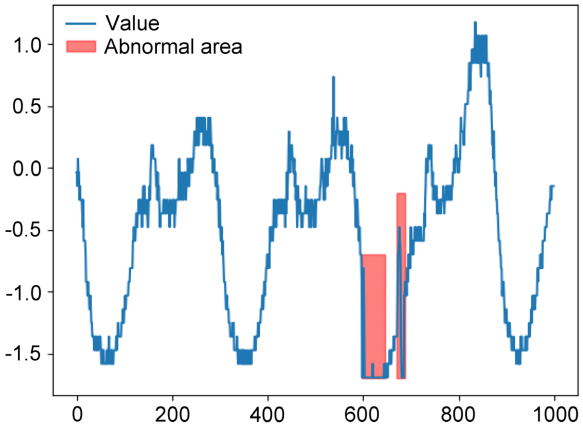


**Figure 4.** Periodic KPI sequence
**图 4.** 周期型 KPI 序列

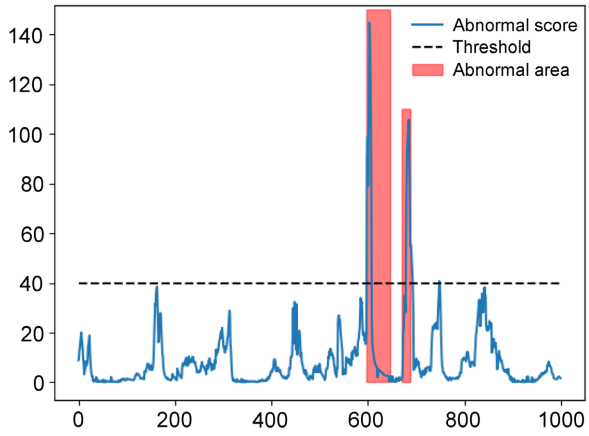


**Figure 5.** Periodic KPI abnormal score
**图 5.** 周期型 KPI 异常得分

对于波动型 KPI 数据，即使是正常数据，也没有一个固定的分布，其数据分布不断变化，异常部分的分布与正常数据差异性不大。所以当异常数据输入时，模型的重构结果与原始输入差别不大，故模型对于此类数据的异常部分分辨能力较差。如图 6、图 7 所示：

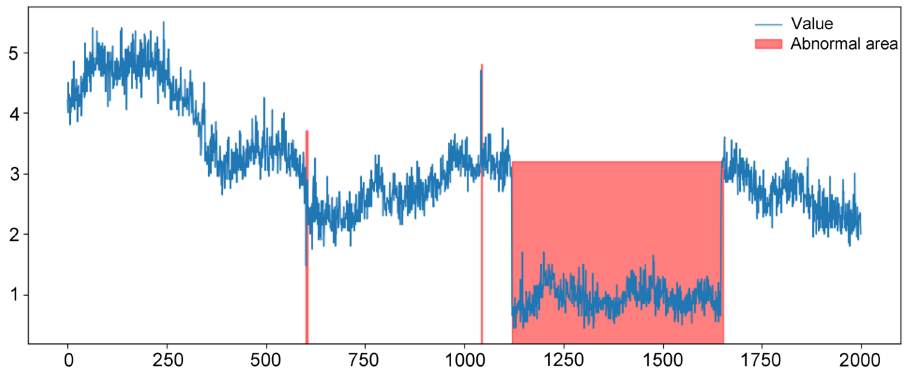


**Figure 6.** Volatility KPI sequence 1
**图 6.** 波动型 KPI 序列 1

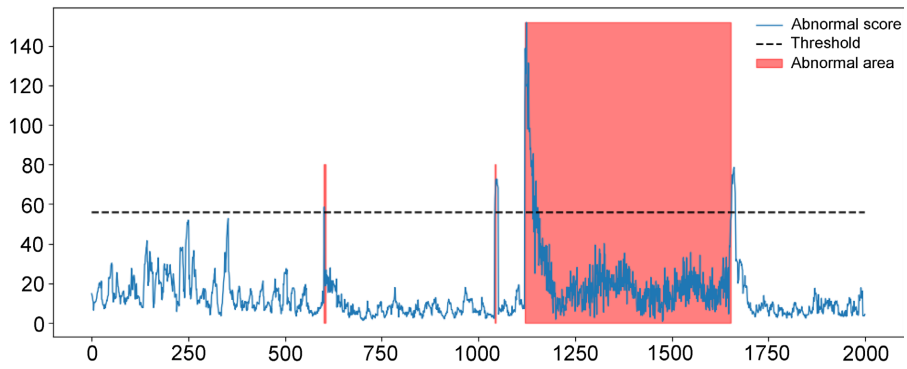


**Figure 7.** Volatility KPI abnormal score 1
**图 7.** 波动型 KPI 异常得分 1

在图 6 上可以看到，第三个异常区间除去在开始和结束时有值突变的情况外，区间内部的指标波动情况直观上来看与正常区间类似。在模型对于该异常区间的异常得分中也体现了这一变化，模型仅在异常开

始和结束部分给出了较高的异常得分，区间内部的异常得分低于阈值。该例中由于 KPI 序列有值突变的现象才使得模型识别到了异常。对于没有值突变的异常区间，模型则很难识成功。如图 8、图 9 所示：

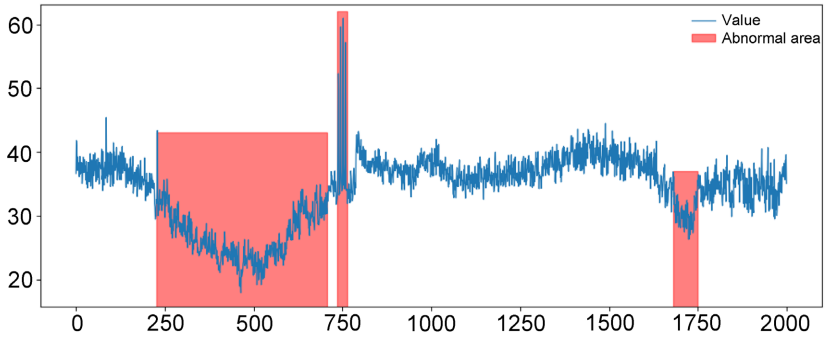


**Figure 8.** Volatility KPI sequence 2
**图 8.** 波动型 KPI 序列 2

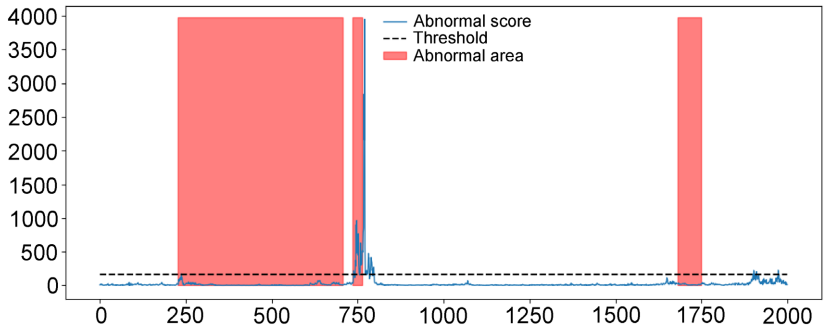


**Figure 9.** Volatility KPI abnormal score 2
**图 9.** 波动型 KPI 异常得分 2

图 8 中可以看到，这段序列有三个异常区间，第一和第三个异常区间直觉上可以理解为指标值波动下降导致的异常。但对于这三个异常区间，模型仅在第二个异常区间上给出了较高的异常得分。对于第一个异常区间，虽然在验证集上最大化 F1 得分时会将异常得分的阈值定的比较低从而成功发现该区间。但就模型效果来讲，模型在异常区间中，并没有根据重构差异给出合适的异常得分。即模型分辨此类异常的能力较差，这在很大程度上影响了模型的精确率。

综上，模型对于周期型 KPI 的异常检测效果较好，对于异常数据的输入能够给出较高的异常得分从而准确及时的分辨指标中的异常数据。对于波动型 KPI 的异常检测具有局限性，重构异常数据能力与重构正常数据差别较小，异常分辨能力较弱。

### 4.2.2. 对比实验

模型在数据集 1、2 上的实验结果如表 2、表 3 所示。

对比本文模型与其他用深度生成模型进行异常检测的方法，本文模型效果更佳。AutoEncoder 模型在重构时序数据时缺乏对时间信息的建模，这使得它应用在时序数据中时效果很差。VAE 模型学习参数的概率分布而非确定的参数值，使用 KL 散度来衡量分布间的差异。这种不确定性使得 VAE 模型对于持续波动型的 KPI 有着较好的效果；EncDec-AD 与 LSTM-NDT 均在编码器部分使用了循环神经网络，这使得它们能够充分挖掘时序数据中的时间特征，实验结果也显示了它们在识别精度上的优势。但这些方法均忽视了 KPI 序列中的空间维度特征。本文模型使用 VGG 网络对空间信息建模，使得编码器能够挖掘数据的局部空间特征，增加了模型在短序列上的异常辨别能力。这令本文模型能够在 T 值较小时保持

较高的召回率，也就是说在保证精度的同时，本文的模型能够更及时的发现异常。从实验结果中可以发现，本文模型的异常发现召回率要优于其他所有方法，且精确率不低。以综合考虑精确率与召回率的F1-score 作为衡量标准来看，本文模型相较于其他模型效果更好，在公开数据集上 F1-score 提高 0.04，在自建数据集上 F1-score 提高 0.06。

**Table 2.** Comparison of the results of various algorithms on data set 1
**表 2.** 数据集 1 上各种算法的结果对比

| 模型 | 精确率(P) | 召回率(R) | F1-score |
|---|---|---|---|
| 本文模型 | 0.81 | 0.96 | 0.88 |
| Autoencoder [11] | 0.68 | 0.71 | 0.69 |
| EncDec-AD [13] | 0.97 | 0.69 | 0.81 |
| VAE [15] | 0.75 | 0.90 | 0.82 |
| TAnoGan [16] | 0.73 | 0.71 | 0.72 |
| LSTM-NDT [17] | 0.92 | 0.73 | 0.81 |

**Table 3.** Comparison of the results of various algorithms on data set 2
**表 3.** 数据集 2 上各种算法的结果对比

| 模型 | 精确率(P) | 召回率(R) | F1-score |
|---|---|---|---|
| 本文模型 | 0.74 | 0.91 | 0.82 |
| Autoencoder [11] | 0.63 | 0.71 | 0.67 |
| EncDec-AD [13] | 0.80 | 0.71 | 0.75 |
| VAE [15] | 0.72 | 0.85 | 0.78 |
| TAnoGan [16] | 0.73 | 0.64 | 0.68 |
| LSTM-NDT [17] | 0.81 | 0.69 | 0.74 |

#### 4.2.3. 参数影响

T 值的设定对模型效果有很大影响，T 值越小代表着模型对异常的发现越及时。表 4 为聚合粒度为 1 分钟的数据在不同 T 值下的实验结果。如表 4 所示，当 T 值为 5 时，模型的查全率较低，这表示数据中有一部分异常，模型是不能在五分钟内识别出来的。当 T 值放宽到 10 的时候，模型效果有着显著提升，查全率从 0.73 提升到了 0.96。这表明当时间放宽一些之后，原本不能发现的异常数据被模型成功识别出来了。当进一步放宽 T 值到 20 时，查准率与查全率均再有提升，分别达到了 0.84 与 0.98，此时的模型效果已经很好，异常的遗漏已经降到 2%。综上，模型可在十分钟内及时的识别绝大多数异常。

**Table 4.** Experimental results under different T
**表 4.** 不同 T 下的实验结果

| T | 精确率(P) | 召回率(R) | F1-score |
|---|---|---|---|
| 5 | 0.79 | 0.73 | 0.76 |
| 10 | 0.81 | 0.96 | 0.88 |
| 15 | 0.81 | 0.97 | 0.88 |
| 20 | 0.84 | 0.98 | 0.90 |

## 5. 结论

本文调研了传统的异常检测方法与近年来 Aiops 领域中新的时间序列异常检测算法，并建立了一个基于 VGG+Bi-LSTM 的编码解码器，深度挖掘时间序列数据中的时间信息与空间信息以重构输入序列，基于重构数据与原始输入数据的偏离程度发现异常。这是一种无监督的、不需要分布假设的方法。这使得该方法能够被很好地应用到流数据中，且能够发现不可预见的错误。模型通过充分挖掘 KPI 序列的局部空间特征，提高了模型在短序列上的异常辨别能力，使得模型相较于其他方法对于异常的发现更及时。实验表明，这种方法可以准确、及时、全面地发现周期型指标中的异常，异常的误报和漏报都较少。

## 基金项目

暗网通信高可靠节点关键技术研究，项目编号：62162060。

## 参考文献

[1] Beyer, B., Jones, C., Petoff, J. and Jones, C. (2016) Site Reliability Engineering: How Google Runs Production Systems. O'Reilly Media, Inc., Sebastopol.

[2] Dang, Y., Lin, Q. and Huang, P. (2019) Aiops: Real-World Challenges and Research Innovations. 2019 *IEEE/ACM* 41*st International Conference on Software Engineering*: *Companion Proceedings* (*ICSE-Companion*), Montreal, 25-31 May 2019, 4-5. https://doi.org/10.1109/ICSE-Companion.2019.00023

[3] Sadik, M.S. and Gruenwald, L. (2010) DBOD-DS: Distance Based Outlier Detection for Data Streams. *International Conference on Database and Expert Systems Applications*, Bilbao, 30 August-3 September, 122-136. https://doi.org/10.1007/978-3-642-15364-8_9

[4] Sadik, S. and Gruenwald, L. (2014) Research Issues in Outlier Detection for Data Streams. *ACM SIGKDD Explorations Newsletter*, **15**, 33-40. https://doi.org/10.1145/2594473.2594479

[5] Laptev, N., Amizadeh, S. and Flint, I. (2015) Generic and Scalable Framework for Automated Time-Series Anomaly Detection. *Proceedings of the* 21*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney, 10-13 August 2015, 1939-1947. https://doi.org/10.1145/2783258.2788611

[6] Liu, D., Zhao, Y., Xu, H., Sun, Y., Pei, D., Luo, J., *et al*. (2015) Opprentice: Towards Practical and Automatic Anomaly Detection through Machine Learning. *Proceedings of the* 2015 *Internet Measurement Conference*, Tokyo, 28-30 October 2015, 211-224. https://doi.org/10.1145/2815675.2815679

[7] Wang, J., Jing, Y., Qi, Q., Feng, T. and Liao, J. (2019) ALSR: An Adaptive Label Screening and Relearning Approach for Interval-Oriented Anomaly Detection. *Expert Systems with Applications*, **136**, 94-104. https://doi.org/10.1016/j.eswa.2019.06.028

[8] Agarwal, D. (2005) An Empirical Bayes Approach to Detect Anomalies in Dynamic Multidimensional Arrays. 5*th IEEE International Conference on Data Mining* (ICDM'05), Houston, 27-30 November 2005, 8 p. https://doi.org/10.1109/ICDM.2005.22

[9] Eskin, E. (2000) Anomaly Detection over Noisy Data Using Learned Probability Distributions. *Proceedings of the* 17*th International Conference on Machine Learning*, Stanford, 29 June-2 July, 2000, 255-262.

[10] Chandola, V., Banerjee, A. and Kumar, V. (2009) Anomaly Detection: A Survey. *ACM Computing Surveys*, **41**, Article No. 15. https://doi.org/10.1145/1541880.1541882

[11] Hawkins, S., He, H. and Williams, G. (2002) Outlier Detection Using Replicator Neural Networks. *International Conference on Data Ware-Housing and Knowledge Discovery*, Aix-en-Provence, 4-6 September 2002, 170-180. https://doi.org/10.1007/3-540-46145-0_17

[12] Sakurada, M. and Yairi, T. (2014) Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. *Proceedings of the MLSDA* 2014 2*nd Workshop on Machine Learning for Sensory Data Analysis*, Gold Coast Australia, 2 December 2014, 4-11. https://doi.org/10.1145/2689746.2689747

[13] Chauhan, S. and Vig, L. (2015) Anomaly Detection in ECG Time Signals via Deep Long Short-Term Memory Networks. 2015 *IEEE International Conference on Data Science and Advanced Analytics* (*DSAA*), Paris, 19-21 October 2015, 1-7. https://doi.org/10.1109/DSAA.2015.7344872

[14] Malhotra, P., Ramakrishnan, A. and Anand, G. (2016) LSTM-Based Encoder-Decoder for Multi-Sensor Anomaly Detection. 2016 *International Conference on Machine Learning*, New York, 19-24 June 2016. arXiv pre-print arXiv: 1607.00148.

[15] Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., *et al*. (2018) Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal Kpis in Web Applications. *Proceedings of the* 2018 *World Wide Web Conference*, Lyon, April 2018, 187-196. https://doi.org/10.1145/3178876.3185996

[16] Bashar, M.A. and Nayak, R. (2020) TAnoGAN: Time Series Anomaly Detection with Generative Adversarial Networks. 2020 *IEEE Symposium Series on Computational Intelligence* (*SSCI*), Canberra, 1-4 December 2020, 1778-1785. https://doi.org/10.1109/SSCI47803.2020.9308512

[17] Hundman, K., Constantinou, V., Laporte, C., Colwell, I. and Soderstrom, T. (2018) Detecting Spacecraft Anomalies Using Lstms and nonparametric Dynamic Thresholding. *Proceedings of the* 24*th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, 19-23 August 2018, 387-395. https://doi.org/10.1145/3219819.3219845

[18] Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W. and Pei, D. (2019) Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. *Proceedings of the* 25*th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, 4-8 August 2019, 2828-2837. https://doi.org/10.1145/3292500.3330672

[19] 戚琦，申润业，王敬宇. GAD: 基于拓扑感知的时间序列异常检测[J]. 通信学报, 2020, 41(6): 152-160.