Computer Science and Application 计算机科学与应用, 2022, 12(3), 590-601
Published Online March 2022 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2022.123060

Hans汉斯

基于XGBoost特征筛选的工业时序数据的重建异常检测算法研究

周旭荣，郑建立

东华大学信息科学与技术学院，上海

收稿日期：2022年2月14日；录用日期：2022年3月10日；发布日期：2022年3月17日

摘　要

针对工业生产中产生的大量时序数据，如何对无用数据进行有效剔除，并且判断传感器所采集数据是否正确，如何对时序数据进行有效异常检测，成为了研究者们关注的问题。在此期间，很多研究者都提出了自己的异常检测算法，但大多只考虑了时序数据的时间性特征，并未将传感器之间的相关性特征考虑进去。所以本文提出一种基于XGBoost特征筛选的多维自注意卷积门控循环编码解码器(MDACGA)，对原始的数据集进行有效特征筛选，根据得分，剔除无关变量，提取有效变量。之后利用有效信息构建特征矩阵，采用全卷积编码器来对特征矩阵进行编码，提取不同时间序列间的相关性特征，采用基于注意力机制的ConvGRU来提取不同时间序列间的时间性特征。最后利用卷积解码器对前一步得到的特征矩阵进行联合解码，从而得到重建后的特征矩阵，利用Adam优化器和小批量随机梯度下降法来最小化重建误差。最终利用残差特征矩阵进行异常检测。实验结果显示，该算法达到0.989的准确率、0.996的召回率，足以表明该异常检测算法具有有效性，并且异常检测效果也优于一般基准算法。

关键词

时序数据，XGBoost，卷积编码器，解码器

Research on Reconstruction Anomaly Detection Algorithm of Industrial Time Series Data Based on XGBoost Feature Selection

Xurong Zhou, Jianli Zheng

School of Information Science and Technology, Donghua University, Shanghai

Received: Feb. 14th, 2022; accepted: Mar. 10th, 2022; published: Mar. 17th, 2022

文章引用: 周旭荣, 郑建立. 基于 XGBoost 特征筛选的工业时序数据的重建异常检测算法研究[J]. 计算机科学与应用, 2022, 12(3): 590-601. DOI: 10.12677/csa.2022.123060

Abstract

In view of the large amount of time series data generated in industrial production, how to effectively eliminate useless data, judge whether the data collected by sensors is correct, and how to effectively detect anomalies of time series data have become the focus of researchers. In this period, many researchers have proposed their own anomaly detection algorithm, but most of them only consider the temporal characteristics of time series data, and do not take into account the correlation between sensors. So this paper proposes a Multi-Dimensional Self-Attention Convolutional Gated Recurrent Encoder and Decoder (MDACGA) based on XGBoost for feature selection, which can effectively filter the original data set and eliminate irrelevant variables according to the score, extraction of valid variables. Then, the effective information is used to construct the feature matrix, and the full convolution encoder is used to encode the feature matrix and extract the correlation features of different time series. ConvGRU-Attention mechanism is used to extract temporal features of different time series. Finally, a convolution decoder is used to jointly decode the feature matrix obtained in the previous step to get the reconstructed feature matrix, and Adam Optimizer and Mini-Batch Stochastic Gradient Descent are used to minimize the reconstruction error. Finally, anomaly detection is carried out by residual error characteristic matrix. The experimental results show that the accuracy of the algorithm is 0.989 and the recall of the algorithm is 0.996, which shows that the anomaly detection algorithm is effective and the anomaly detection effect is better than the general benchmark algorithm.

Keywords

Time-Series Data, XGBoost, Convolution Encoder, Decoder

1. 引言

随着时代的发展，以及科技的进步，时序数据(按照时间顺序变化的数据)与人们的生活息息相关，在军事、金融、医疗、网络安全方面都得到了很好的体现，尤其是在工业化高度发展的今天，工厂时序数据的异常检测成为了数据分析领域的热门研究方向。所谓异常值是指与其他观测值表现相差较大以至于产生怀疑的观测值[1]。异常检测的任务是根据数据分析的形式，找出哪些数据实例与其他数据实例表现不同，即异常值[2]。

由于异常的不可预知性以及高度变化性，人们通常将异常检测任务视为无监督学习[3]，传统的有监督学习并不适用于时序数据的异常检测。有很多研究人员开发出了一些比较有效的异常检测方法，其中包括一些聚类模型，通过聚类不同的数据样本来预定义异常值得分来判别异常。同样，基于距离的方法，例如 K 最近邻算法[4]，其算法是通过计算每个数据样本和相邻数据的平均距离来获得异常得分，从而达到异常检测的效果。还有一些分类方法，例如一类支持向量机[5]，通过对训练数据的密度分布进行建模，将时序数据分为正常值和异常值。虽然在很多研究中证明了这些方法的有效性，但是对于处理多元时序数据，他们的异常检测效果差强人意。之后研究者提出了自回归综合移动平均(ARIMA)模型[6]，虽然可以对多元时序数据进行有效异常检测，但是只能模拟线性特征，无法有效表示系统的非线性特征，并且对噪声相对敏感，抗噪能力差。

由于传统的机器学习方法表现出了诸多的弊端，研究者们将目光放到了基于深度学习的无监督异常检测的方法上。例如深度自编码高斯混合模型[7]，将高斯混合模型与深度自动编码器有效结合，对多元数据的密度分布进行建模，从而达到时序数据异常检测的目标。LSTM 编码解码器[8]利用长短时记忆网络对时序数据的时间依赖性进行建模，相较于传统方法，泛化能力得到了增强。但此类模型依旧存在弊端，他们只考虑了时间序列的时间依赖性，在真实系统中，往往具备若干传感器，所接收到的多元时序数据具有复杂性，不同的时间序列间也表现出时间依赖性和相关性特征，如何将他们有效结合从而达到异常检测的效果成为了研究者们有待解决的问题，这也使得关于时序数据的异常检测成为了一项具有挑战性的工作。

所以本文构建的基于 XGBoost 的多维自注意卷积门控循环编码器不仅考虑了多维时序数据的时间性特征，并且将时序数据的相关性特征也有效提取，二者有机结合，提高了时序数据异常检测的有效性和准确性。其中采用 XGBoost 对多变量输入进行特征提取和筛选，能够做到有效缩短训练时长，并将多余特征变量剔除，增强了网络的健壮性和准确性。

2. 基于 XGBoost 的多维自注意卷积门控循环编码器

2.1. 问题阐述

异常检测的目标是有效地处理、分析真实数据集，利用特定算法识别出异常点。假设给定 N 个长度为 m 的时间序列，可用公式表示为

$$\mathrm{X}_N = \left(\mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_{N-1}, \mathrm{x}_N\right)^{\mathrm{T}} \in R^{N\times m} \tag{2.1}$$

其中 N 表示变量个数，m 表示时间序列的长度。

2.2. 特征筛选

如图 1(1)所示，本文采用 XGBoost 对实验数据集中数据进行特征筛选，过滤无关变量，保留有效变量。XGBoost 是一种基于梯度提升决策树(GBDT)的极致梯度提升算法。该思想在[9]中正式提出，作者提到 XGBoost 基本思想与 GBDT 相同，但是进行了很多优化，比如为了优化损失函数，提高计算精确度，采用二阶泰勒公式进行展开；使用正则项来简化模型，避免过拟合；采用 Blocks 存储结构，实现并行计算。

其目标函数包含两部分，分别为损失函数与正则化项，可表示为：

$$Obj = \sum_i l\left(y_i, \hat{y}_i\right) + \sum_k \Omega\left(f_k\right) \tag{2.2}$$

其中 $\sum_{i=1}^{n} l\left(y_i, \hat{y}_i\right)$ 是损失函数，$\hat{y}_i$ 为预测值，$\sum_k \Omega\left(f_k\right)$ 是正则化项，表示 k 棵树的复杂度。由于

$$\hat{y}_i = \sum_{k=1}^{K} f_k\left(x_i\right) = \hat{y}_i^{(t-1)} + f_t\left(x_i\right) \tag{2.3}$$

所以目标函数可表示为

$$Obj = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t\left(x_i\right)\right) + \sum_k \Omega\left(f_k\right) \tag{2.4}$$

为了优化损失函数，根据二阶泰勒公式展开，则损失函数可表示为

$$\sum_{i=1}^{n}\left[l\left(y_i,\hat{y}_i^{(t-1)}\right)+g_i f_t\left(x_i\right)+\frac{1}{2}h_i f_t^2\left(x_i\right)\right] \tag{2.5}$$

其中 g_i 为损失函数的一阶导数，h_i 为损失函数的二阶导数，这里的求导是指对 $\hat{y}_i^{(t-1)}$ 求导，去除常数项 $l\left(y_i,\hat{y}_i^{(t-1)}\right)$，损失函数可表示为

$$\sum_{i=1}^{n}\left[g_i f_t\left(x_i\right)+\frac{1}{2}h_i f_t^2\left(x_i\right)\right] \tag{2.6}$$

为了优化正则化项，将正则化项展开，得到

$$\sum_k \Omega\left(f_k\right)=\sum_{k=1}^{t}\Omega\left(f_k\right)=\Omega\left(f_t\right)+\sum_{k=1}^{t-1}\Omega\left(f_k\right) \tag{2.7}$$

因为 $(t-1)$ 棵树的结构已经确定，所以可视为常数，所以去除常数项 $\sum_{k=1}^{t-1}\Omega\left(f_k\right)$，则正则化项可表示为 $\Omega\left(f_t\right)$。

定义一棵树，设置叶子结点的权重向量为 w，叶子结点的映射关系为 q，则 $f_t\left(x\right)=w_{q(x)}$，$w\in R^T$，之后定义树的复杂度 Ω，则

$$\Omega\left(f_t\right)=\gamma T+\frac{1}{2}\lambda\sum_{j=1}^{T}w_j^2 \tag{2.8}$$

其中 T 表示叶子结点的数量，w_j^2 表示叶子结点权重向量的 L_2 范数。本文将属于第 j 个叶子结点的所有样本 x_i 划分到一个叶子结点的样本集合中，可用数学公式表示为：

$$I_j=\left\{i\mid q\left(x_i\right)=j\right\} \tag{2.9}$$

则 XGBoost 的目标函数可以表示为：

$$Obj^{(t)}\simeq\sum_{i=1}^{n}\left[g_i f_t\left(x_i\right)+\frac{1}{2}h_i f_t^2\left(x_i\right)\right]+\Omega\left(f_t\right) \tag{2.10}$$

将 $f_t\left(x_i\right)=w_{q(x_i)}$ 代入目标函数得：

$$Obj^{(t)}=\sum_{i=1}^{n}\left[g_i w_{q(x_i)}+\frac{1}{2}h_i w_{q(x_i)}^2\right]+\gamma T+\frac{1}{2}\lambda\sum_{j=1}^{T}w_j^2 \tag{2.11}$$

最后将所有训练样本，按照叶子结点进行分组得：

$$Obj^{(t)}=\sum_{j=1}^{T}\left[\left(\sum_{i\in I_j}g_i\right)w_j+\frac{1}{2}\left(\sum_{i\in I_j}h_i+\lambda\right)w_j^2\right]+\gamma T \tag{2.12}$$

定义叶子结点 j 所包含样本的一阶偏导数之和为 $G_j=\sum_{i\in I_j}g_i$，叶子结点 j 所包含样本的二阶偏导数之和为 $H_j=\sum_{i\in I_j}h_i$，将其代入目标函数得：

$$Obj^{(t)}=\sum_{j=1}^{T}\left[G_j w_j+\frac{1}{2}\left(H_j+\lambda\right)w_j^2\right]+\gamma T \tag{2.13}$$

最后通过目标函数求解最优解来计算各个特征重要性得分。根据提供的总样本中各个特征的重要性

分数，进行无关变量的剔除，得到处理后的多变量时间序列

$$\mathrm{X}_n = \left(\mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_{n-1}, \mathrm{x}_n\right)^{\mathrm{T}} \in R^{n \times m} \tag{2.14}$$

其中 $n \le N$ ，n 表示变量个数。

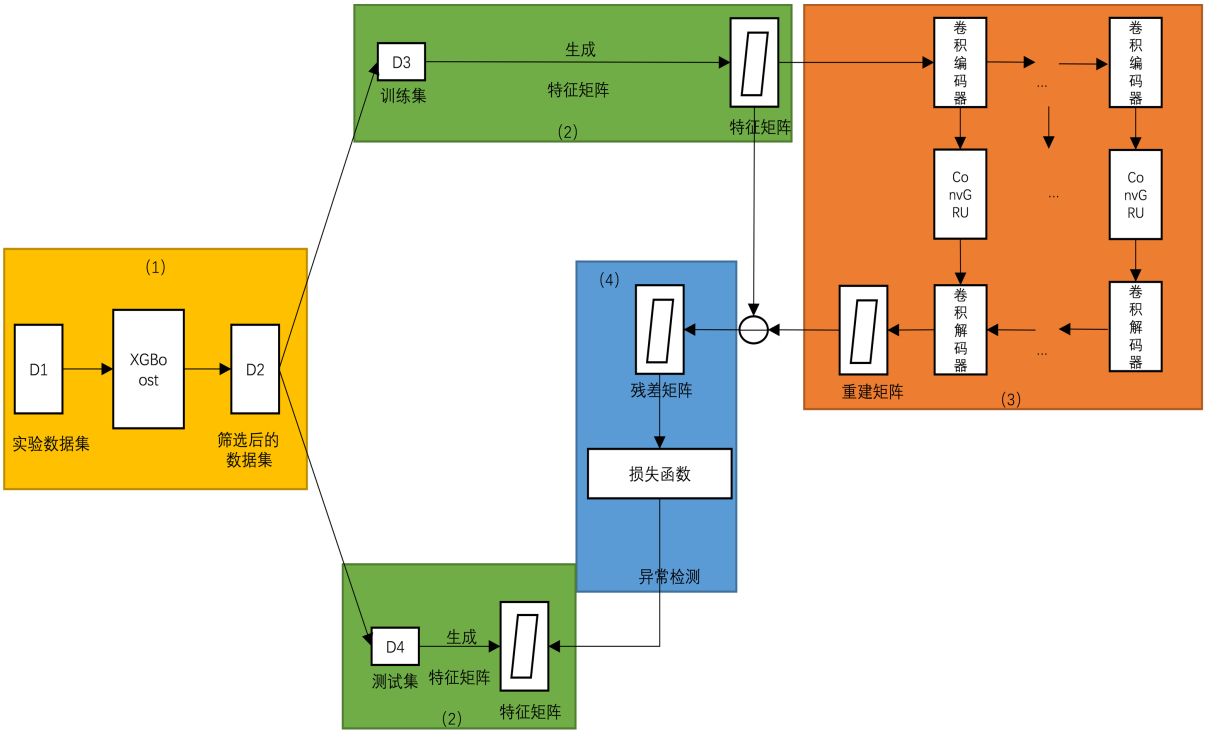

Figure 1. The MDACGA block diagram based on XGBoost: (1) feature selection based on XGBoost; (2) Generate the characteristic matrix; (3) Extracting correlation and temporal features; (4) Loss function
图 1. 基于 XGBoost 的 MDACGA 框图：(1) XGBoost 特征筛选；(2) 生成特征矩阵；(3) 提取相关性、时间性特征；(4) 损失函数

2.3. 构建特征矩阵

论文[10]中提到，不同时间序列间的相关性是表征系统状态的重要因素，为了提取不同时间序列间的相互关系，根据[11]中提到的方法，利用前面处理过的多变量时间序列，取特定时间长度的时间序列片段，然后将不同时间序列片段两两内积，形成 $n \times n$ 的特征矩阵 M_t，如图 1(2)所示。假设两个时间序列，取时间窗口为 w，时间序列片段可表示为 $\mathrm{x}_i^w = \left(x_i^{t-w}, x_i^{t-w+1}, \cdots, x_i^{t-1}, x_i^t\right)$ 和 $\mathrm{x}_j^w = \left(x_j^{t-w}, x_j^{t-w+1}, \cdots, x_j^{t-1}, x_j^t\right)$，则他们之间的相关性 $m_{ij}^t \in M^t$ 按下列公式计算：

$$m_{ij}^t = \frac{\sum_{\delta=0}^{w} x_i^{t-\delta} x_j^{t-\delta}}{\kappa} \tag{2.15}$$

其中 κ 为缩放因子($\kappa = w$)。这样构造的特征矩阵 M^t 不仅具有不同时间序列间的相关性特征，而且对输入的噪声具有鲁棒性。

2.4. 提取相关性特征

如图 1(3)中所示，本文利用全卷积编码器[12]来对特征矩阵进行编码，提取不同时间序列间的相关性

特征。将前面构建的特征矩阵看作是一个张量 $\chi^{t,0} \in R^{n\times n\times 1}$，之后将其输入进若干卷积层，假设 $\chi^{t,l-1} \in R^{n_{l-1}\times n_{l-1}\times 1}$ 表示第 $(l-1)$ 层的特征映射，那么第 l 层的输出可表示为：

$$\chi^{t,l} = f\left(W^l * \chi^{t,l-1} + b^l\right) \tag{2.16}$$

其中 $*$ 表示卷积运算，$f(\cdot)$ 为激活函数，$W^l \in R^{k_l\times k_l\times d_{l-1}\times d_l}$ 表示大小为 $k_l \times k_l \times d_{l-1}$ 的 d_l 卷积核，$b^l \in R^{d_l}$ 为偏置项，$\chi^{t,l} \in R^{n_l\times n_l\times d_l}$ 表示第 l 层的输出特征图。

之后，利用缩放指数线性单元(SELU) [13]作为激活函数，设置四个卷积层，分别具有 32 个步长为1×1，大小为3×3×3 的卷积核；64 个步长为 2×2，大小为 3×3×32 的卷积核；128 个步长为 2×2，大小为 2×2×64 的卷积核；以及 256 个步长为 2×2，大小为 2×2×128 的卷积核。

2.5. 提取时间性特征

前文利用卷积编码器所提取的相关性特征映射在时间上依赖之前的时间步，早在[14]中就已提及使用 ConvLSTM 来捕获视频序列中的时间信息，但是需要计算大量的参数，耗时较长。相比较于 ConvLSTM 而言，ConvGRU 具有相似的结构、相似的建模效果，但是参数较少，训练时间较短。与此同时，传统的 ConvLSTM 和 ConvGRU 在处理时间序列时，随着时间序列长度的增加，效果明显下降。为了有效处理长时间序列，如图 1(3)所示，本文提出基于注意力机制的 ConvGRU，能够有效提取前一个时间步所包含的信息。假设给定第 l 个卷积层 $x^{t,l}$ 和之前的隐藏层 $h^{t-1,l}$，ConvGRU [15]作为 GRU 的拓展可定义为：

$$r^{t,l} = \sigma\left(W_r^l * x^{t,l} + U_r^l * h^{t-1,l} + b_h^l\right) \tag{2.17}$$

$$z^{t,l} = \sigma\left(W_z^l * x^{t,l} + U_z^l * h^{t-1,l} + b_z^l\right) \tag{2.18}$$

$$\tilde{h}^{t,l} = \tanh\left(W_h^l * x^{t,l} + r^{t,l} \odot U_h^l * h^{t-1,l} + b_h^l\right) \tag{2.19}$$

$$h^{t,l} = z^{t,l} \odot \tilde{h}^{t,l} + \left(1 - z^{t,l}\right) \odot h^{t-1,l} \tag{2.20}$$

其中 $*$ 表示卷积操作，$\odot$ 表示哈达玛积，W，U，b 分别表示 ConvGRU 的可学习参数：前向连接权值、循环连接权值和偏置参数。$r^{t,l}$ 和 $z^{t,l}$ 分别表示 ConvGRU 的复位门和更新门，$h^{t-1,l}$ 为$(t-1)$时刻隐藏状态，$\tilde{h}^{t,l}$ 则为$(t\text{-}1)$时刻候选状态，最后得出 t 时刻隐藏状态 $h^{t,l}$。与 GRU 相比，ConvGRU 的卷积运算保留了空间拓扑，在二维特征图上使用二维权值核。考虑到并不是前面所有的时间步都与 $h^{t,l}$ 相关，所以采用注意力机制，来自适应地选择与当前时间步相关的时间步和聚合信息，从而将特征映射到一个精炼的输出特征图 $\hat{H}^{t,l}$，公式如下：

$$\hat{h}^{t,l} = \sum_{i\in(t-h,t)} \alpha^i h^{i,l} \tag{2.21}$$

$$\alpha^i = \frac{\exp\left\{\dfrac{\exp\left\{Vec\left(h^{t,l}\right)^{\mathrm{T}} Vec\left(h^{i,l}\right)\right\}}{\chi}\right\}}{\sum_{i\in(t-h,t)} \exp\left\{\dfrac{Vec\left(h^{t,l}\right)Vec\left(h^{i,l}\right)}{\chi}\right\}} \tag{2.22}$$

其中$Vec(\cdot)$表示矢量，χ为缩放因子。即取最后一个隐藏状态$h^{t,l}$作为上下文向量，通过 softmax 函数度量前几步的重要度权重α^i。基于注意力机制的 ConvGRU 很好地在每个卷积层联合建模带有时间信息的特征矩阵的空间结构。

2.6. 卷积解码

卷积解码器的作用是对前一步所得到的特征矩阵进行解码，从而得到重建后的特征矩阵，卷积解码器[12]表达式如下：

$$\hat{x}^{t,l-1}=\begin{cases} f\left(\hat{W}^{t,l}\otimes\hat{h}^{t,l}+\hat{b}^{t,l}\right) & l=4 \\ f\left(\hat{W}^{t,l}\otimes\left[\hat{h}^{t,l}\oplus\hat{x}^{t,l}\right]+\hat{b}^{t,l}\right) & l=3,2,1 \end{cases} \tag{2.23}$$

其中$\otimes$表示反卷积运算，$\oplus$表示串联运算，$f(\cdot)$为激活单元，$\hat{W}^{t,l}$为第l个卷积层的滤波核和偏置参数。具体而言，本文按照逆序将 ConvGRU 第l层的$\hat{h}^{t,l}$输入进反卷积神经网络中。输出特征图$\hat{x}^{t,l-1}$与前一个 ConvGRU 层的输出串联，将他们之间的串联表示进一步输入到下一个反卷积层，最终得到输出$\hat{x}^{t,0}$ (重建后的矩阵)，如图 1(3)所示。

卷积解码器使用与卷积编码器相对应的四个反卷积层，分别为 128 个步长为2×2，大小为$2\times2\times64$的卷积核；64 个步长为2×2，大小为$3\times3\times32$的卷积核；32 个步长为1×1，大小为$3\times3\times3$的卷积核；以及 3 个步长为1×1，大小为$3\times3\times64$的卷积核。该卷积解码器能够有效联合 ConvGRU 和反卷积层的特征映射，提高了异常检测的性能。

2.7. 损失函数

对于 MDACGA 而言，目标定义为特征矩阵的重建误差，即：

$$\Gamma_{\text{MDACGA}}=\sum_{t}\sum_{c=1}^{s}\left\|x_{:,:,c}^{t,0}-\hat{x}_{:,:,c}^{t,0}\right\|_F^2 \tag{2.24}$$

如图 1(4)所示，本文采用[16]中提到的 Adam 优化器和小批量随机梯度下降法来最小化上述损失。经过若干训练阶段后，学习到的神经网络参数可被用来推断验证集和测试集中数据的重建特征矩阵。最后，利用残差特征矩阵进行异常检测。

3. 实验验证与结果分析

3.1. 实验数据及特征筛选

本实验利用田纳西-伊斯曼(TE)仿真平台生成的 TE 标准数据集来进行算法验证，该仿真平台是美国 Eastman 化学公司依靠实际化工反应过程开发的开放性化学仿真平台，具备公认性和权威性。该平台采集到的数据与时间关联性强，并且变量间具备相关性特征，所以在工业控制以及故障检测领域都得到了广泛的应用。整个 TE 数据集由两部分构成，包括训练集与测试集，由于所有的数据都是通过 22 次不同的实验产生的，所以其中包含了 52 个观测变量以及 21 种不同的故障类型。本实验选取 1417 个正常时间点，其中插入 7 个含有故障类型一的时间点，8 个含有故障类型二的时间点，8 个含有故障类型三的时间点，8 个含有故障类型四的时间点。首先利用 XGBoost 来进行特征变量的重要性筛选，如图 2 所示，选取出评分较高的 46 个控制变量，实验数据集具体参数见表 1：

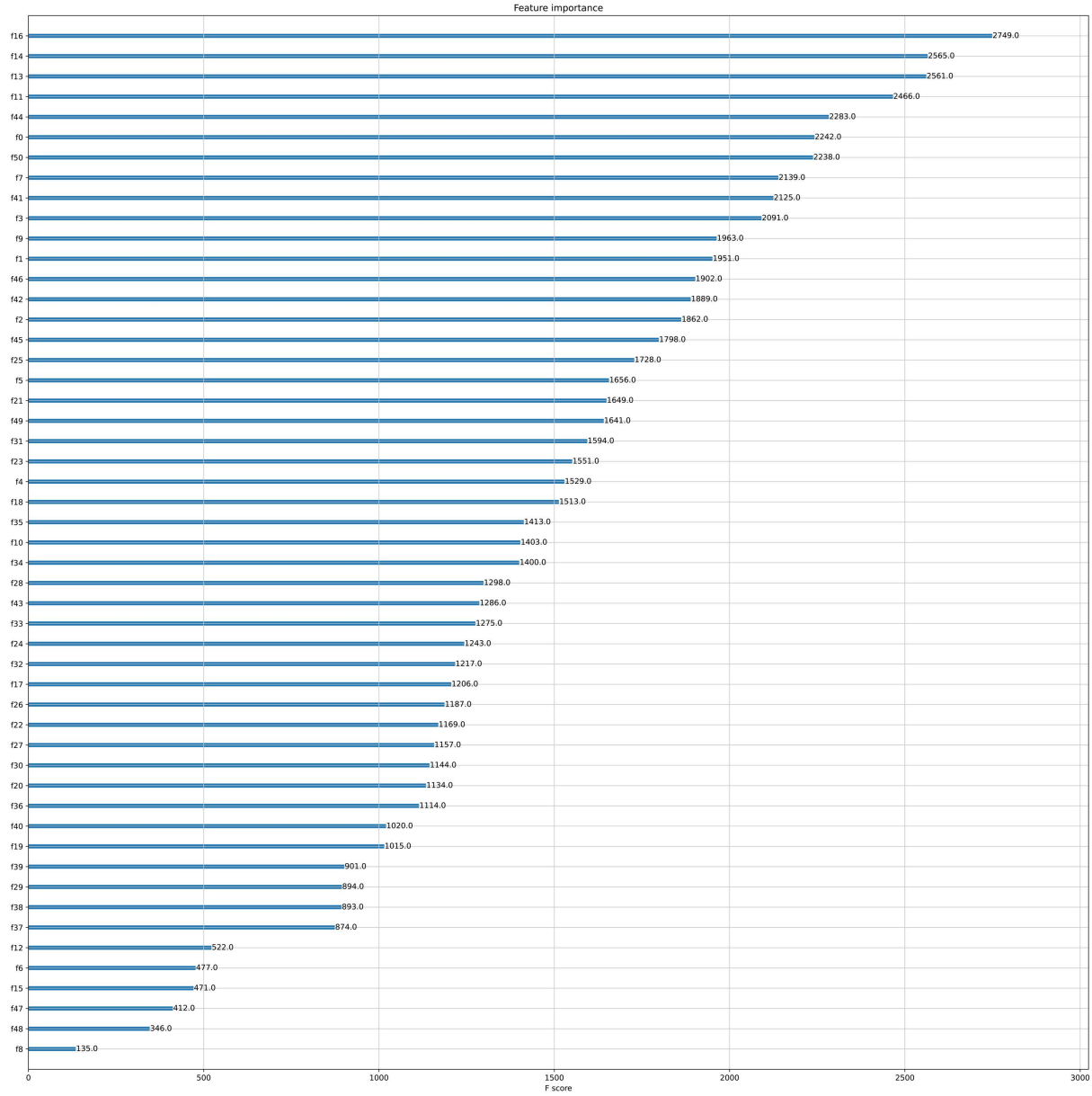

Figure 2. The XGBoost feature filtering
图 2. XGBoost 特征筛选

Table 1. The specific parameters of the experimental data set
表 1. 实验数据集具体参数

参数	值
控制变量个数	46
时间点个数	1449
异常类型个数	4
训练集长度	863

Continued

验证集长度	292
测试集长度	293

3.2. 评价指标

本文使用精度(Precision)、召回率(Recall)和F1分数(F1 Score)三个指标来评估本文的方法和对比模型的性能。

$$F1 = \frac{2 \times Prec \times Rec}{Prec + Rec} \tag{3.1}$$

$$Prec = \frac{TP}{TP + FP} \tag{3.2}$$

$$Rec = \frac{TP}{TP + FN} \tag{3.3}$$

其中 Prec 表示精度，Rec 表示召回率，TP、TN、FP、FN 分别表示真阳性、真阴性、假阳性和假阴性的数量。Prec 为精确率，等于 TP 比上检测结果为正常的样本($TP + FP$)， $Prec < 1$，Prec 越大表示精确度越高；Rec 为召回率，等于 TP 比上实际为正常的样本($TP + FN$)， $Rec < 1$，Rec 越大表示召回率越高。为了达到异常检测的目标，使用验证数据集上的最大异常分数来设置阈值。在测试集中，任意时间点异常得分超过阈值将被视为异常。

3.3. 实验结果与分析

本文采用基于 XGBoost 特征筛选的 MDACGA 与两种深度学习算法进行对比，其中包括 LSTM 算法[17]、MSCRED 算法[11]以及 MDACGA 算法，根据表 2 可以看出不同检测算法在使用 XGBoost 与未使用 XGBoost 进行多元时序数据异常检测时的性能，其中在都未使用 XGBoost 进行特征筛选的情况下，实验后计算得到的 MDACGA 的精度优于其他两种算法，说明甄别错误的能力明显优于其他两种算法；根据三种算法的召回率比较，可以得出 LSTM 与 MSCRED 倾向于将数据标记成正常数据，而 MDACGA 则在数据标记方面比较均衡。在都使用 XGBoost 进行特征筛选的情况下，结果亦是如此。

Table 2. The abnormal detection results of different algorithms
表 2. 不同算法异常检测结果

算法名称	未使用 XGBoost			使用 XGBoost		
	Prec	Rec	F1	Prec	Rec	F1
LSTM	0.970	0.936	0.953	0.970	0.946	0.957
MSCRED	0.974	0.964	0.969	0.978	0.971	0.974
MDACGA	0.982	0.979	0.980	0.989	0.996	0.992

由图 3~5 可以看出，使用相同算法进行异常检测，在有无使用 XGBoost 进行特征变量筛选的前提下，相同算法表现出了不同的异常检测结果，其中可以看到，使用了 XGBoost 的算法，无论是异常检测的精确性，还是异常检测的稳定性，相较于未使用 XGBoost 的算法，表现都要更加优越。尤其是使用 XGBoost 进行特征筛选之后，模型都表现出了更好的健壮性与适应性。

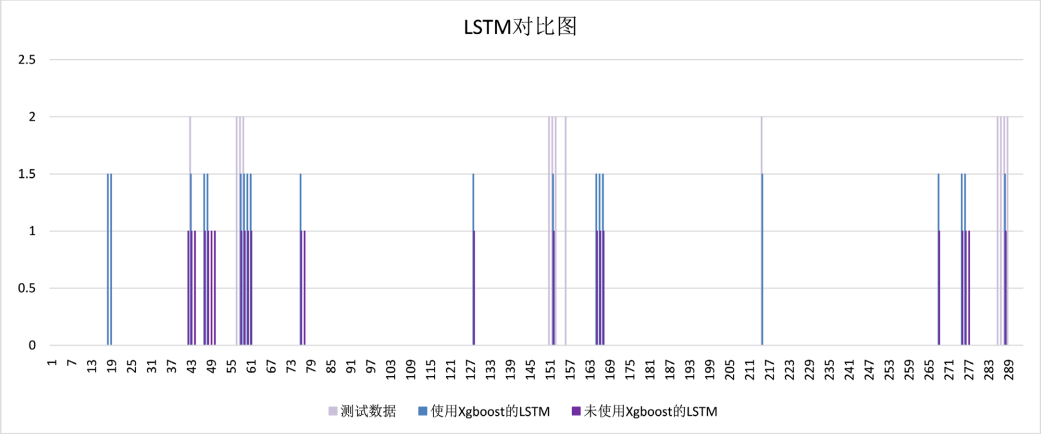

Figure 3. Using XGBoost to filter the abnormal detection results of LSTM before and after feature selection
图 3. 使用 XGBoost 进行特征筛选前后的 LSTM 异常检测结果

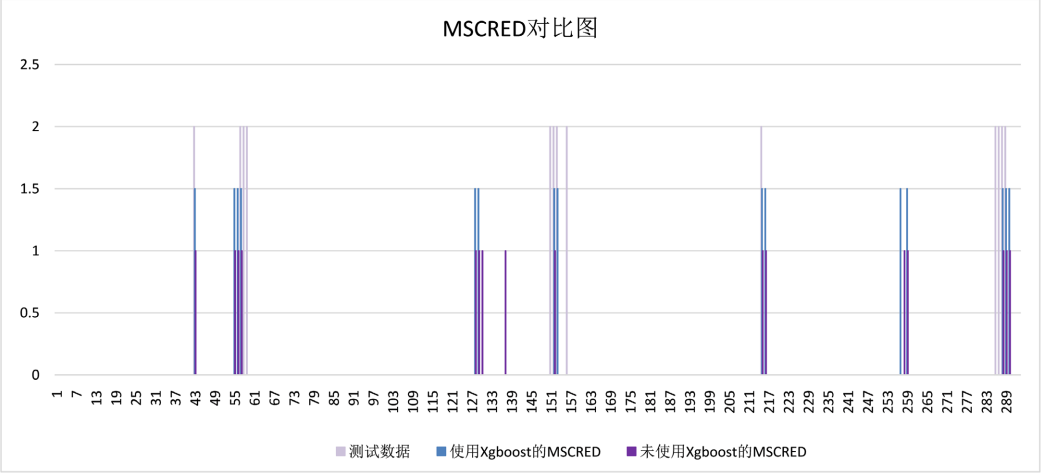

Figure 4. Using XGBoost to filter the abnormal detection results of MSCRED before and after feature selection
图 4. 使用 XGBoost 进行特征筛选前后的 MSCRED 异常检测结果

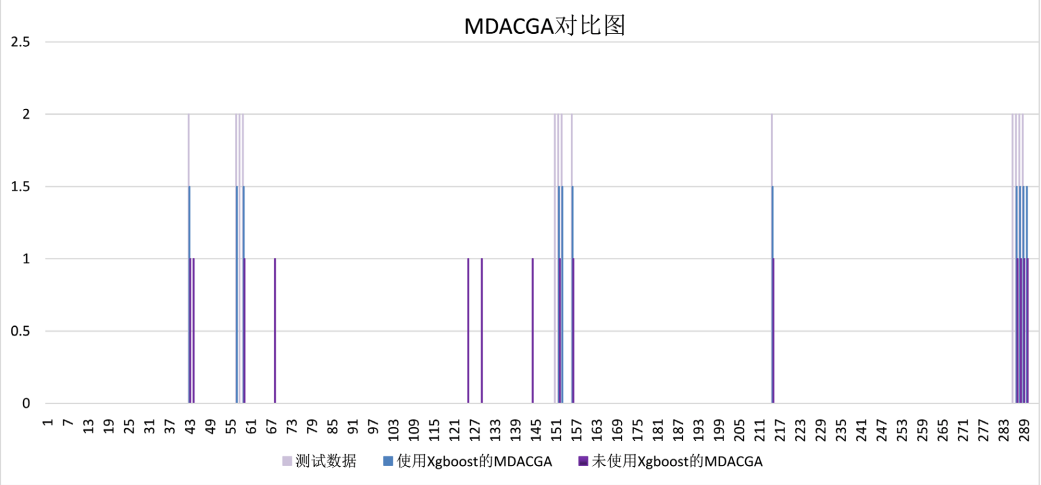

Figure 5. Using XGBoost to filter the abnormal detection results of MDACGA before and after feature selection
图 5. 使用 XGBoost 进行特征筛选前后的 MDACGA 异常检测结果

4. 结语

在真实数据集上的实验结果表明，基于 XGBoost 特征筛选的 MDACGA 在多变量时间序列的异常检测方面，具有有效性；与其他两种深度学习算法相比，基于 XGBoost 特征筛选的 MDACGA 在异常检测方面具有优越性；并且使用 XGBoost 进行特征筛选有效提高了算法进行异常检测的准确性。但是工业数据集种类繁多，在数据集过大、传感器变量过多时，由于噪声过大，导致异常检测结果不尽如人意；数据集过小，传感器变量过少时，由于相关性特征不够明显，结果亦是不够理想，所以在后期研究中，将提高该检测模型的抗噪性与泛化性。

参考文献

[1] Atkinson, A.C. and Hawkins, D.M. (1981) Identification of Outliers. *Biometrics*, **37**, 860-861. https://doi.org/10.2307/2530182

[2] Chandola, V., Banerjee, A. and Kumar, V. (2007) Outlier Detection: A Survey. *ACM Computing Surveys*, **41**, Article 15.

[3] Deng, A. and Hooi, B. (2021) Graph Neural Network-Based Anomaly Detection in Multivariate Time Series.

[4] Hautamäki, V., Kärkkäinen, I. and Fränti, P. (2004) Outlier Detection Using k-Nearest Neighbour Graph. *Proceedings of the* 17*th International Conference on Pattern Recognition*, Cambridge, 26-26 August 2004, 430-433. https://doi.org/10.1109/ICPR.2004.1334558

[5] Manevitz, L.M. and Yousef, M. (2002) One-Class SVMs for Document Classification. *Journal of Machine Learning Research*, **2**, 139-154.

[6] Zhou, Y., Qin, R., Xu, H., Sadiq, S. and Yu, Y. (2018) A Data Quality Control Method for Seafloor Observatories: The Application of Observed Time Series Data in the East China Sea. *Sensors* (*Switzerland*), **18**, 2628. https://doi.org/10.3390/s18082628

[7] Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D. and Chen, H. (2018) Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection. 6*th International Conference on Learning Representations*, Vancouver, 30 April-3 May 2018, 1-19.

[8] Qin, Y., Song, D., Cheng, H., Cheng, W., Jiang, G. and Cottrell, G.W. (2017) A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. *IJCAI International Joint Conference on Artificial Intelligence*, Melbourne, 19-25 August 2017. https://doi.org/10.24963/ijcai.2017/366

[9] Chen, T. and Guestrin, C. (2016) XGBoost: A Scalable Tree Boosting System. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, 13-17 August 2016, 785-794. https://doi.org/10.1145/2939672.2939785

[10] Song, D., Xia, N., Cheng, W., Chen, H. and Tao, D. (2018) Deep r-th Root of Rank Supervised Joint Binary Embedding for Multivariate Time Series Retrieval. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, London, 19-23 August 2018, 2229-2238. https://doi.org/10.1145/3219819.3220108

[11] Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H. and Chawla, N.V. (2019) A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. 33*rd AAAI Conference on Artificial Intelligence*, *AAAI* 2019, 31*st Innovative Applications of Artificial Intelligence Conference*, *IAAI* 2019 *and the* 9*th AAAI Symposium on Educational Advances in Artificial Intelligence*, *EAAI* 2019, July 2019, 1409-1416. https://doi.org/10.1609/aaai.v33i01.33011409

[12] Shelhamer, E., Long, J. and Darrell, T. (2017) Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**, 640-651. https://doi.org/10.1109/TPAMI.2016.2572683

[13] Klambauer, G., Unterthiner, T., Mayr, A. and Hochreiter, S. (2017) Self-Normalizing Neural Networks. *Advances in Neural Information Processing Systems* 30: *Annual Conference on Neural Information Processing Systems* 2017, Long Beach, 4-9 December 2017.

[14] Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K. and Woo, W.C. (2015) Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *Annual Conference on Neural Information Processing Systems* 2015, Montreal, 7-12 December 2015.

[15] Jung, M., Lee, H. and Tani, J. (2018) Adaptive Detrending to Accelerate Convolutional Gated Recurrent Unit Training for Contextual Video Recognition. *Neural Networks*, **105**, 356-370. https://doi.org/10.1016/j.neunet.2018.05.009

[16] Kingma, D.P. and Ba, J.L. (2015) Adam: A Method for Stochastic Optimization. 3*rd International Conference on Learning Representations, ICLR* 2015—*Conference Track Proceedings*, San Diego, 7-9 May 2015.

[17] Malhotra, P., Vig, L., Shroff, G. and Agarwal, P. (2015) Long Short Term Memory Networks for Anomaly Detection in Time Series. 23*rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN* 2015—*Proceedings*, Bruges, 22-23 April 2015, 89-94.