信息与控制　2019 年　第 48 卷　第 1 期：1 ~ 8

DOI：10.13976/j.cnki.xk.2019.8062　　文章编号：1002 - 0411(2019) - 01 - 0001 - 08

基于深度学习的时间序列数据异常检测方法

胡姣姣[1]，王晓峰[1]，张　萌[1]，张德鹏[1]，胡绍林[2]

1. 西安理工大学理学院，陕西 西安　710054；　2. 广东石油化工学院自动化学院，广东 茂名　525000

基金项目：国家自然科学基金资助项目(61772416，91646108，61473222)；陕西省教育厅重点实验室项目(17JS098)

通信作者：王晓峰，xfwang66@sina.com.cn　收稿/录用/修回：2018 - 01 - 29/2018 - 06 - 08/2018 - 08 - 20

摘要

针对类间分布不平衡的时间序列数据的异常检测问题，提出了一种基于深度卷积神经网络的检测方法. 首先采用抽样法对不平衡时间序列数据进行预处理；其次，将处理后的时间序列数据转换为尺度一致、时长一致的片段；最后将数据送入具有 4 层隐藏层结构的卷积神经网络模型中进行异常检测. 实验结果表明，所提方法弥补了现存的检测技术由于忽略数据分布的偏斜性而造成的少数类检测精度低的缺点，并通过与现有的时间序列分类方法的比较，验证了所提方法的高效性.

关键词

时间序列异常检测

不平衡数据学习

深度学习

卷积神经网络

中图法分类号：TP1

文献标识码：A

Time-series Data Anomaly Detection Method Based on Deep Learning

HU Jiaojiao[1], WANG Xiaofeng[1], ZHANG Meng[1], ZHANG Depeng[1], HU Shaolin[2]

1. *School of Science*, *Xi'an University of Technology*, *Xi'an* 710054, *China*;

2. *School of Automation*, *Guangdong University of Petrochemical Technology*, *Maoming* 525000, *China*

Abstract

With regard to the anomaly detection problem of time-series data with a skewed between-class distribution, we propose a detection method based on deep convolutional neural network. First, we employ the sampling method to preprocess the unbalanced time-series data. Second, the original time-series data are converted into a series of continuous segments with a uniform scale and consistent duration. Finally, we feed the data into a convolutional neural network model with four hidden layers for anomaly detection. The experimental results show that the proposed method covers the shortage of existing detection technologies that ignore the skewness of data distributions and results in a low-detection precision. Compared with the existing time-series classification methods, the proposed method provided a satisfactory performance.

Keywords

time series anomaly detection;

unbalanced data learning;

deep learning;

convolutional neural network

0　引言

随着信息技术的发展，时间序列数据正以惊人的速度产生于现实生活中的几乎每一个应用领域，如金融分析、气象研究、网络安全、航空航天数据处理等. 但是在实际工程中产生的很多时间序列数据集都是不平衡数据集，即序列中的正常信号数量远大于异常信号数量，这种数据的不平衡分布会降低机器学习方法的性能，无法取得理想的分类结果，因此不平衡时间序列数据分类问题成了数据挖掘领域的又一大挑战.

不平衡时间序列数据的研究是包含在不平衡数据的研究之中的. 目前对不平衡数据问题的研究主要分为 3 类：基于数据层面的研究、基于算法层面的研究和集成方法[1]. 对于基于深度学习的时间序列分类方法而言，基于数据层面的研究是最有效的. 其核心思想就是通过数据处理，尽量减少不平衡数据集中正常类样本和异常类样本的数量差异. 典型的不平衡数据处理方法是采样法. 采样的方法分为 2 种：对多数类的欠采样和对少数类的过采样. 在欠采样技术中最重要的就是随机欠采样方法[2]，该方法通过随机移除多数类样本使数据集达到平衡. 欠采样方法存在的问题是可能会造成数据集中有价值的信息丢失. 在过采样技术中，随机过采样方法[3]通过随机复制少数类样本来达到平衡类分布的目的. 过采样技术不会添加额外的信息，只是对数据的重用，所采取的解决方法是人工生成新的少数类样本. 人工生成少数类过采样技术[4](synthetic minority over-sampling technique，SMOTE)是一种广泛使用的技术，该方法首先计算少数类样本集中每一个样本的 K 近邻，然后随机从这 K 个近邻中选出 $n(n<K)$ 个样本，将这 n 个样

本与相对应的少数类样本进行随机差值，从而生成了 n 个新的少数类样本，进而达到平衡数据集的目的. SMOTE 算法及其之后的衍生算法在不平衡分类问题上得到了广泛地应用[5]. Cao 等[6]提出了一种集成过采样(INtegrated Over Sampling, INOS)方法也取得了一个非常好的效果.

时间序列数据分类方法的研究是近年来数据挖掘领域的研究热点，时间序列数据分类方法主要包括基于距离的分类算法、基于特征的分类算法和基于模型的分类算法. 基于距离的分类算法主要集中在不同距离度量的最近邻(one nearest neighbor, 1-NN)算法上，如基于欧氏距离(Euclidean distance, ED)的最近邻算法[7]和基于动态时间扭曲(dynamic time warping, DTW)的最近邻算法[8]. 动态时间扭曲的方法对噪声数据和相位漂移比较稳定，能较好地处理时间轴上的形变. 因此，基于 DTW 的最近邻算法是时间序列中用途最广泛的分类方法. 基于特征的分类算法主要集中在选取区分不同类别的最佳特征. 时间序列的 Shapelets 作为序列中最具代表性的子序列是这类方法的典型代表，通过搜索时间序列的 Shapelets 解决时间序列的分类问题. 基于此提出了很多搜索最佳 Shapelets 的研究方法，如通过自组织增量神经网络来学习时间序列中的 Shapelets[9]的时间序列分类方法和使用 top-k 查询技术[10]筛选最具代表性的 Shapelets 的时间序列分类方法. 基于模型的分类算法主要集中在从时间序列的整体出发，将特征提取阶段与分类阶段组合在一起进行处理. 近年来随着深度学习的发展，基于模型的方法也随之火热起来，神经网络成为了主流的学习算法，通过对数据的自主学习自动提取具有代表性的特征并分类. 卷积神经网络[11]是这种分类器的典型代表，在此基础上还有很多基于卷积神经网络的变体网络也很好的应用于时间序列的分类，如多尺度神经网络(MCNN)[12]及全卷积神经网络(FCN)[13]时间序列分类模型.

目前不平衡数据问题研究的方法有很多，时间序列分类方法也有很多，但是将这两者结合的方法却微乎其微. 针对这种不平衡时间序列数据，本文提出了一种基于深度卷积神经网络(deep convolutional neural network, DCNN)的偏斜类时间序列数据异常检测算法，通过对不平衡时间序列数据的预处理，缩小类间样本数量的差距，再利用 DCNN[14]进行学习分类. 本文的贡献：

1）针对不平衡时间序列数据提出了一种数据处理算法，增加了少数类样本的代表性数据，提高分类器对少数类样本的特征学习能力.

2）利用尺度变换和时间切片技术对原始时间序列数据进行处理，将原始数据转换为尺度一致、时长一致的片段，提高卷积神经网络的检测性能.

3）研究了不同隐藏层的卷积神经网络对于时间序列数据检测性能的影响，提出了具有 4 层隐藏层的时间序列数据检测模型. 通过仿真实验表明，提出的基于 DCNN 的时间序列异常检测方法在检测精度和时间效率上均有较好的性能.

1 不平衡时间序列数据处理方法

不平衡学习问题主要关注数据表示不充分和类分布扭曲变形时学习算法的性能[15]. 考虑到 DCNN 深度学习系统需要足够多的样本进行训练学习，因此本文采用抽样法对少数类异常值数据集进行处理，缩小异常类样本与正常类样本数量之间的差距，使学习算法得出较好的结果.

本文采用抽样法，对不平衡时间序列数据集中的异常数据集进行理，具体步骤为：

设 $T\{t_i(x_i, y_i)\}(i=1, 2, \cdots, n)$表示数据集，其中 x_i 表示第 i 个样本的时间戳，y_i 表示第 i 个样本的信号值，n 表示数据集中数据的总数量.

步骤 1 采用模糊聚类算法对数据集 $T\{t_i(x_i, y_i)\}$ 进行聚类，将数据集中的样本分为孤立点集 $T_1\{t_{1i}(x_{1i}, y_{1i})\}(i=1, 2, \cdots, n_1)$、零界点集 $T_2\{t_{2i}(x_{2i}, y_{2i})\}(i=1, 2, \cdots, n_2)$、安全点集 $T_3\{t_{3i}(x_{3i}, y_{3i})\}(i=1, 2, \cdots, n_3)$三类，并根据聚类算法得到各类的聚类中心 $O_1(x'_1, y'_1)$、$O_2(x'_2, y'_2)$、$O_3(x'_3, y'_3)$.

步骤 2 分别计算点集 $T_1\{t_{1i}(x_{1i}, y_{1i})\}$、$T_2\{t_{2i}(x_{2i}, y_{2i})\}$、$T_3\{t_{3i}(x_{3i}, y_{3i})\}$中样本点与聚类中心的距离，距离计算公式为

$$\begin{cases} d_{1i} = |y_{1i} - y'_1|, & i=1, 2, \cdots, n_1 \\ d_{2i} = |y_{2i} - y'_2|, & i=1, 2, \cdots, n_2 \\ d_{3i} = |y_{3i} - y'_3|, & i=1, 2, \cdots, n_3 \end{cases} \tag{1}$$

其中 d_{1i}表示点集 $T_1\{t_{1i}(x_{1i}, y_{1i})\}$中第 i 个样本点到聚类中心 $O_1(x'_1, y'_1)$的距离，d_{2i}表示点集 $T_2\{t_{2i}(x_{2i}, y_{2i})\}$中第 i 个样本点到聚类中心 $O_2(x'_2, y'_2)$的距离，d_{3i}表示点集 $T_3\{t_{3i}(x_{3i}, y_{3i})\}$中第 i 个样本点到聚类中心 $O_3(x'_3, y'_3)$的距离.

步骤 3 取点集 $T_1\{t_{1i}(x_{1i}, y_{1i})\}$中某一样本点 $t(x, y)$，此样本点到点集 $T_1\{t_{1i}(x_{1i}, y_{1i})\}$的聚类中心 $O_1(x'_1, y'_1)$的距离记为 d，搜索满足式(2)的所有样本点：

$$d_{1i} = |y_{1i} - y'_1| < d, \quad i=1, 2, \cdots, n_1 \tag{2}$$

并按照时间分量的早晚顺序进行排序，结果记为

$$t_{11}(x_{11}, y_{11}), t_{12}(x_{12}, y_{12}), \cdots, t_{1m}(x_{1m}, y_{1m}) \tag{3}$$

步骤 4 在样本 $t(x, y)$与 $t_{11}(x_{11}, y_{11})$，$t_{12}(x_{12}, y_{12})$，$\cdots$，$t_{1m}(x_{1m}, y_{1m})$的信号分量值之间进行随机线性插值，构造新样本的信号分量值 $\tilde{y}_k(k=1, 2, \cdots, m)$：

$$\tilde{y}_k = y + \text{rand}(0, 1) \times (y - y_{1k}) \tag{4}$$

其中 rand(0, 1)表示区间(0, 1)内的一个随机数.

步骤 5 构造新样本的时间分量值 $\tilde{x}_k(k=1, 2, \cdots, m)$：

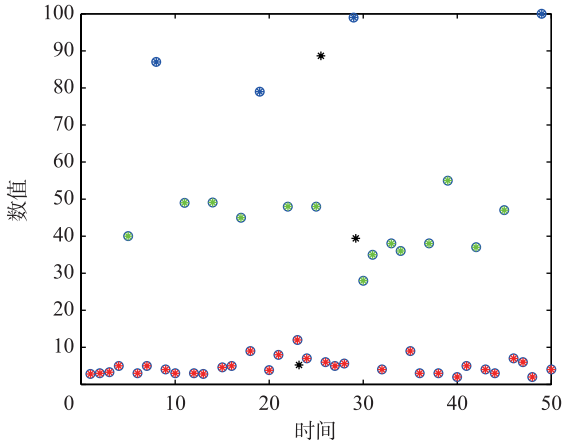
$$\tilde{x}_k = x + \frac{x_{1k} - x}{m} \tag{5}$$

其中 $x_{1k}(k=1, 2, \cdots, m)$为样本 $t_{11}(x_{11}, y_{11})$，$t_{12}(x_{12}, y_{12})$，$\cdots$，$t_{1m}(x_{1m}, y_{1m})$中的时间戳. 最终得到新合成的样本为 $T'\{t_k(\tilde{x}_k, \tilde{y}_k)\}(k=1, 2, \cdots, m)$.

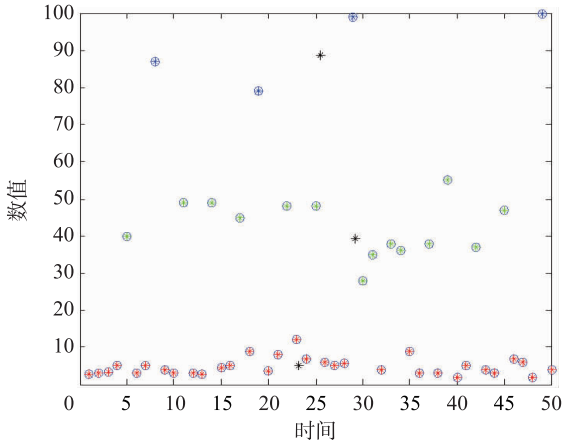
步骤 6 对点集 $T_2\{t_{1i}(x_{1i}, y_{1i})\}$、$T_3\{t_{3i}(x_{3i}, y_{3i})\}$中的每个样本点分别重复步骤 3 ~ 步骤 5 中的操作，得到所有的合成样本，将这些新样本点合并到原来的数据集里即可以产生新的数据集 $\bar{T}\{t_i(x_i, y_i)\}(i=1, 2, \cdots, N)$，$N$

表示经过不平衡数据处理后新生成的数据集中数据的总数量.

图 1(a)展示了步骤 1 中聚类后的结果，红色、绿色、蓝色的实心圆分别表示聚类后样本的分布，黑色星形表示聚类后每类的聚类中心. 图 1(b)展示了步骤 3 中距离搜索的过程，以其中一类(临界点样本)为例，计算出每个样本与聚类中心的距离，搜索到聚类中心小于样本点 $t(x, y)$ 到聚类中心距离的样本点(未加粗的灰色线条连接的样本点). 图 2(a)展示了未经过不平衡数据处理后的原始样本的分布. 图 2(b)展示了经过不平衡数据处理后新生成样本的分布.



(a) 聚类结果



(b) 距离比较

图 1　数据生成过程

Fig.1　Data generation process

2　基于 DCNN 模型的不平衡时间序列数据异常检测方法

本节基于文[16]和文[17]的卷积神经网络模型，确定了一种具有 4 层隐藏层结构的卷积神经网络模型，用于时间序列数据异常检测.

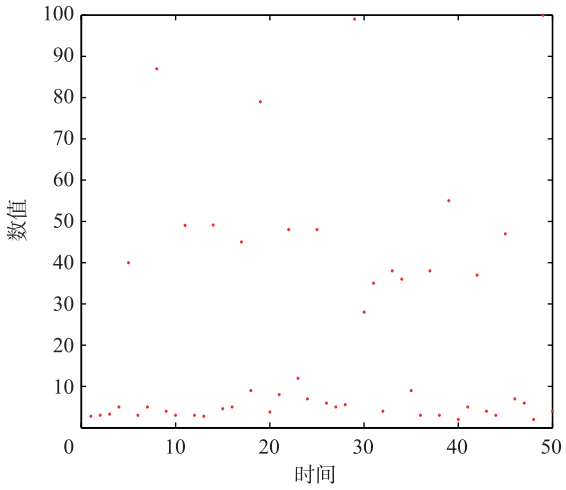
2.1　输入处理

2.1.1　尺度变换

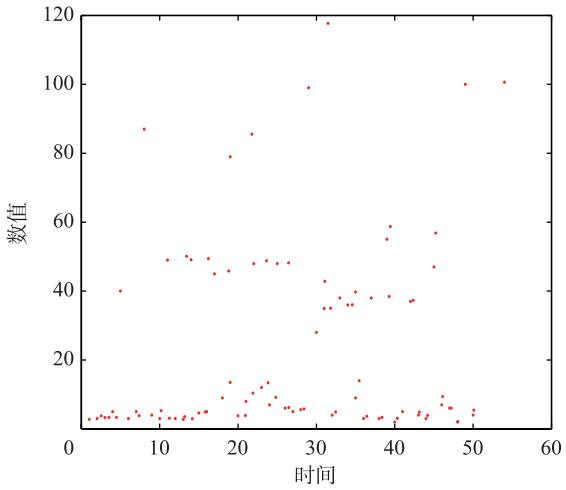
在实际工程中采集到的数据都有其自身的数据特性，数据的量纲都不一致. 当训练学习时，由于某些数据的范围可能特别大，使得其在模式分类中的作用偏大，从而导致神经网络收敛慢、训练时间长；而数据范围小的输入，其作用就可能会偏小. 因此需要将所有的原始数据进行标准化处理，得到尺度一致的数据集：

$$\bar{y}_i = \frac{y_i - \bar{Y}}{S} \tag{6}$$

其中，$\bar{Y}$ 为信号分量的平均值，S 表示信号分量的标准差.



(a) 原始数据



(b) 新生成样本

图 2　原始样本与新生成样本对比

Fig.2　The original sample is compared with the newly generated sample

2.1.2　时间切片

实际的时间序列数据是带有时间戳的长序列，为了更好地进行数据特征的学习，必须将原始信号创建为固定大小的段. 为了保持时间序列数据在时序上的依赖性，受文[16]切割方法的启发，本文提出了一种滑动窗口的重叠切片方法，选择固定长度 L 的窗口函数 f 以固定步长 l 移动对原始信号进行分割，将原始信号分为等间隔的时间序列片段：

$$\boldsymbol{S}=\boldsymbol{f}(\{\overline{T}(t_i(x_i,\ y_i)\})=\begin{bmatrix}\boldsymbol{s}_1\\ \boldsymbol{s}_2\\ \vdots\\ \boldsymbol{s}_j\\ \vdots\\ \boldsymbol{s}_{\overline{N}}\end{bmatrix}=\begin{bmatrix}[t_1(x_1,\ y_1):\ t_L(x_L,\ y_L)]\\ [t_{l+1}(x_{l+1},\ y_{l+1}):\ t_{L+l}(x_{L+l},\ y_{L+l})]\\ \vdots\\ [t_{lj-l+1}(x_{lj-l+1},\ y_{lj-l+1}):\ t_{lj+L-l}(x_{lj+L-l},\ y_{lj+L-l})]\\ \vdots\\ [t_{N-L+1}(x_{N-L+1},\ y_{N-L+1}):\ t_N(x_N,\ y_N)]\end{bmatrix} \tag{7}$$

其中，$\boldsymbol{S}$ 表示切片后时间序列片段的矩阵，$\boldsymbol{s}_j$ 表示切片后第 j 个时间序列片段，N 为实验数据集中的数据总数量，$\overline{N}$ 为切片后时间序列的片段总数. 本文所采用的实验数据集中时间序列数据的采样点为 450 次/分钟，为了控制卷积神经网络输入层神经元数量在适当的范围及充分利用样本数据，设置时间切片窗口大小为 150，步长为 75.

2.2 网络结构

本文采用 4 层隐藏层的卷积神经网络模型(隐藏层层数的确定见 3.3 节)，如图 3 所示，整体结构由 2 个部分组成：特征提取和分类. 其中，特征提取部分由 4 层的隐藏层组成，分类部分选用 SoftMax 作为分类器. 下面详细描述了 DCNN 对于时间序列数据的处理过程.

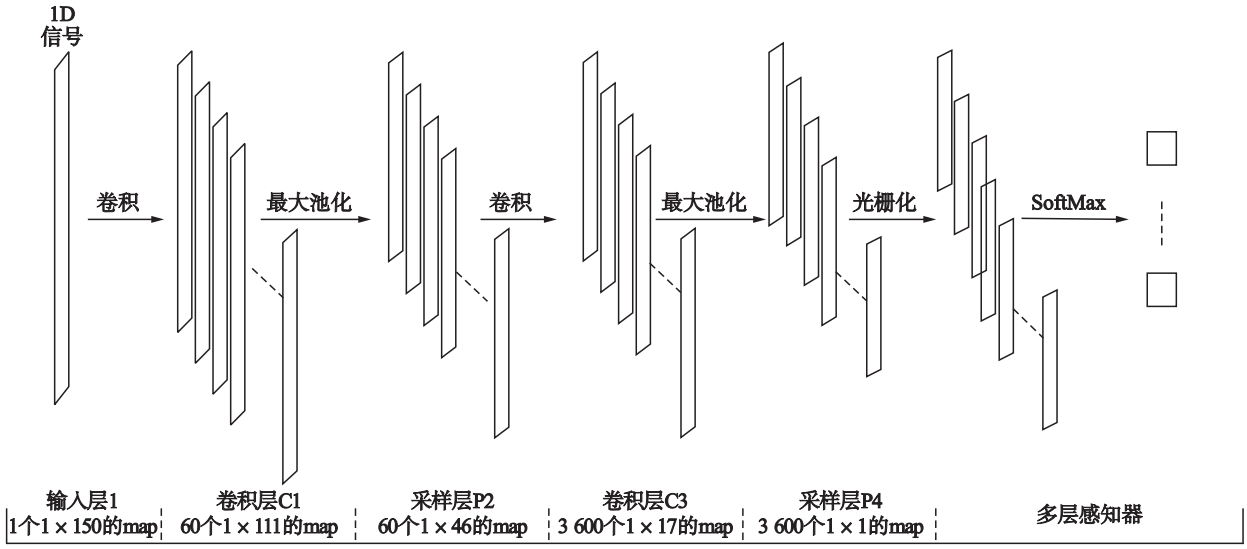


图 3 卷积神经网络结构图[16-17]

Fig.3 Structure diagram of convolutional neural network

输入层：用于数据输入，本文是将经过尺度变换、时间切片等预处理后所得到的长度为 150 个时间戳的时间序列片段 $\{\boldsymbol{s}_j\}(j=1,\ 2,\ \cdots,\ \overline{N})$ 输入到 DCNN 网络中.

隐藏层：隐藏层是 DCNN 网络中进行特征提取和特征映射的重要部分，其中包括卷积和池化两个重要操作. 卷积就是两个函数和生成第 3 个函数的一种数学算子，即 $f(\cdot)\otimes g(\cdot)$，表示函数 $f(\cdot)$ 与 $g(\cdot)$ 经过翻转和平移的重叠部分的面积. 本文中 $f(\cdot)$ 表示输入信号 s_j；$g(\cdot)$ 表示高斯卷积核，如式(8)所示：

$$g(x)=\mathrm{e}^{\frac{-x^2}{2\sigma^2}} \tag{8}$$

其中，σ 表示函数的宽度参数，用于控制函数的径向作用范围，本文中取 $\sigma=0.1$.

经过卷积层后，输出的特征向量尺寸较大，对于后面的计算复杂度较高，因此需要减小它的尺寸，保留主要的信息特征. 最常见的池化操作为最大值池化、平均值池化. 本文中采用最大值池化，通过在小窗口长度上获取最大值来对特征图进行下采样，保留最具价值的特性.

C1 层：假定 C1 层有 n_1 个大小为 c_1 的卷积核 $\boldsymbol{K}_j^1(j=1,\ 2,\ \cdots,\ n_1)$，本文取 $n_1=60$，$c_1=40$，则将输入层得到的数据，经过 60 个大小为 40 的卷积核 $\boldsymbol{K}_j^1(j=1,\ 2,\ \cdots,\ 60)$，卷积生成 60 个大小为 111 的特征向量 $\boldsymbol{X}_j^1(j=1,\ 2,\ \cdots,\ 60)$，

$$\begin{cases}\boldsymbol{u}_j^1=\boldsymbol{C}(\boldsymbol{s}_j,\ K_j^1)+\boldsymbol{b}_j^1\\ \boldsymbol{X}_j^1=\mathrm{ReLU}(\boldsymbol{u}_j^1)\end{cases} \tag{9}$$

其中，$\boldsymbol{b}_j^1$ 表示 C1 层的偏置，$\boldsymbol{C}(\cdot)$ 表示卷积函数，ReLU$(\cdot)$ 表示激活函数.

P2 层：假定 P2 层有步长为 s_2 大小为 p_2 的池化窗口，本文取 $s_2=2$，$p_2=20$，则将 C1 层得到的特征向量 $\boldsymbol{X}_j^1$，经过大小为 20 移动步长为 2 的小窗口采样后，生成 60 个大小为 46 的特征向量 $\boldsymbol{X}_j^2(j=1,\ 2,\ \cdots,\ 60)$：

$$\begin{cases}\boldsymbol{u}_j^2=\boldsymbol{\beta}_j^2\boldsymbol{d}(\boldsymbol{X}_j^1)+\boldsymbol{b}_j^2\\ \boldsymbol{X}_j^2=\mathrm{ReLU}(\boldsymbol{u}_j^2)\end{cases} \tag{10}$$

其中，$\boldsymbol{\beta}_j^2$ 表示 P2 层的共享权值，$\boldsymbol{b}_j^2$ 表示 P2 层的偏置，$\boldsymbol{d}(\cdot)$ 表示下采样函数.

C3 层：假定 C3 层有 n_3 个大小为 c_3 的卷积核 $\boldsymbol{K}_{ji}^3(i=1,\ 2,\ \cdots,\ n_3,\ j=1,\ 2,\ \cdots,\ n_1)$，本文取 $n_3=60$，$c_3=30$，

则将 P2 层得到的特征向量 $\boldsymbol{X}_j^2$，经过 60 个大小为 30 的卷积核 $\boldsymbol{K}_{ji}^3(i, j=1, 2, \cdots, 60)$，卷积生成 3 600 个大小为 17 的特征向量 $\boldsymbol{X}_j^3(j=1, 2, \cdots, 3\ 600)$：

$$\begin{cases} \boldsymbol{u}_j^3 = \sum_{i=1}^{60} \boldsymbol{C}(\boldsymbol{X}_j^2, \boldsymbol{K}_{ji}^3) + \boldsymbol{b}_j^3 \\ \boldsymbol{X}_j^3 = \mathrm{ReLU}(\boldsymbol{u}_j^3) \end{cases} \tag{11}$$

其中，$\boldsymbol{b}_j^3$ 表示 C3 层的偏置.

P4 层：假定 P4 层有步长为 s_4 大小为 p_4 的池化窗口，本文取 $s_4=2$，$p_4=16$，则将 C3 层得到的特征向量 $\boldsymbol{X}_j^3$，经过大为 16 移动步长为 2 的小窗口采样后，生成 3 600 个大小为 1 的特征向量 $\boldsymbol{X}_j^4(j=1, 2, \cdots, 3\ 600)$：

$$\begin{cases} \boldsymbol{u}_j^4 = \boldsymbol{\beta}_j^4 \boldsymbol{d}(\boldsymbol{X}_j^3) + \boldsymbol{b}_j^4 \\ \boldsymbol{X}_j^4 = \mathrm{ReLU}(\boldsymbol{u}_j^4) \end{cases} \tag{12}$$

其中，$\boldsymbol{\beta}_j^4$ 表示 P4 层的共享权值，$\boldsymbol{b}_j^4$ 表示 P4 层的偏置.

光栅化：最后将 $\boldsymbol{X}_j^4(j=1, 2, \cdots, 3\ 600)$ 顺序展开成一个长向量，作为全连接层网络的输入.

F5 层：多层感知器(multilayer perceptron，MLP)：是一种前向结构的人工神经网络，映射一组输入向量到一组输出向量. 本文使用了 3 层感知器，即 1 个输入层、1 个隐藏层及 1 个输出层. 将光栅化后的特征 $\boldsymbol{X}_j^4(j=1, 2, \cdots, 3\ 600)$ 作为 MLP 的输入，进入隐藏层进行特征映射：

$$\begin{cases} \boldsymbol{u}_j^5 = \sum_{j=1}^{3\ 600} \boldsymbol{X}_j^4 \boldsymbol{w}_j^5 + \boldsymbol{b}_j^5 \\ \boldsymbol{X}_j^5 = \tanh(\boldsymbol{u}_j^5) \end{cases} \tag{13}$$

其中，$\boldsymbol{w}_j^5$ 表示 F5 层的共享权重，$\boldsymbol{b}_j^5$ 表示 F5 层的共享偏置，$\tanh(\cdot)$ 表示 tanh 激活函数.

输出层：使用 SoftMax 类器进行逻辑回归，输出信号属于类别 1(正常值)或者 2(异常值)的概率 $P_k(k=1, 2)$，将概率大的类别作为 DCNN 分类的结果：

$$P_k = \mathrm{softmax}(\boldsymbol{X}_j^5), \quad k=1, 2 \tag{14}$$

输出层输出了每个时间片段属于每一类别的概率，本文以交叉熵(cross entropy)作为代价函数(见式(15))，以自适应学习率优化算法—AdamOptimizer—作为反向传播训练算法进行误差的最小化训练，得到最优的权重参数，通过分类精度(见式(16))评估模型的分类性能，建立最优的时间序列数据分类模型.

$$H = -\sum y_k \log p_k \tag{15}$$

其中，y_k 表示期望的标签类型，p_k 为实际的输出.

$$\mathrm{Acc} = \frac{n'}{N'} \tag{16}$$

其中，n'表示测试数据集中被正确识别的时间片段，N'表示测试数据集中总的时间片段的数量.

2.3　隐藏层层数设置

隐藏层是 DCNN 结构中进行特征学习与提取的重要部分，隐藏层层数的不同将导致学习到不同的特征表示，因此本文将预处理后的实验数据集以 8:2 的比例分为训练数据集和测试数据集进行实验，确定分类性能最好的隐藏层层数. 本文所使用的网络结构中隐藏层参数以及不同隐藏层结构的神经网络模型的分类结果见表 1. 表 1 中，layer(n)($n=3, 4, 5, 6$)表示隐藏层结构为 n 层，conv1、conv2 和 conv3 分别表示第 1 个、第 2 个和第 3 个卷积层. conv1 下面的一行表示卷积层的参数，如(50, 60)表示卷积核大小为 50，数量为 60. maxpool 表示池化层，下面的一行表示池化层参数，例如(20, 2)表示池化窗口的大小为 20，步长为 2. 表 1 的最后一行为实验分类结果.

表 1　不同隐藏层的参数设置

Tab.1　Parameter settings of different hidden layer

层数	layer(3)	layer(4)	layer(5)	layer(6)
输入	150	150	150	150
隐藏层及参数	conv1 (50, 60) MaxPool (20, 2) conv2 (40, 60)	conv1 (40, 60) MaxPool (20, 2) conv2 (30, 60) MaxPool (16, 2)	conv1 (30, 30) MaxPool (15, 2) conv2 (20, 20) MaxPool (12, 2) conv3 (12, 6)	conv1 (30, 30) MaxPool (20, 2) conv2 (12, 12) MaxPool (10, 2) conv3 (10, 10) MaxPool (6, 2)
特征图维度	3 600	3 600	3 600	3 600
最高分类精度	0.987 30	0.991 7	0.989 75	0.988 28

由表 1 的最高分类精度数据可得，对于实验数据集而言，当设置隐藏层层数为 4 层时分类识别精度最高，其中输入长度均为 150，全连接层最终学习得到的特征维数为 3 600，迭代次数为 1 000.

根据表 1 中的参数进行实验，得到不同隐藏层的分类精度和训练损失，见图 4 和表 2、表 3. 图 4 中上部的 4 条曲线分别表示不同隐藏层结构的神经网络的分类精度，下部的 5 条曲线分别表示不同隐藏层结构的神经网络模型分类时的训练损失.

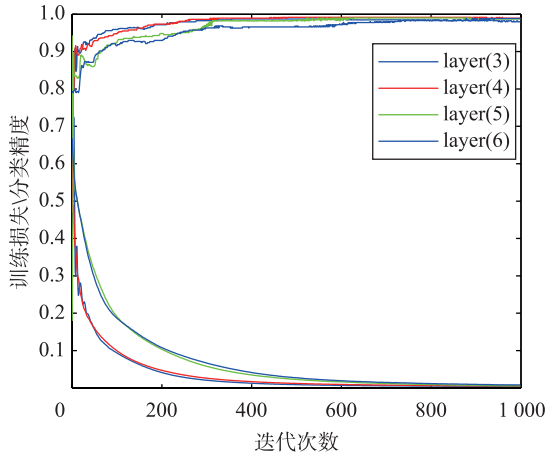


图 4　不同隐藏层结构下 DCNN 的分类性能比较

Fig.4　Comparison of DCNN classification performance under different hidden layers

表 2 不同隐藏层结构下 DCNN 的分类精度比较

Tab.2 Comparison of DCNN classification accuracy under different hidden layers

迭代次数	精度			
	layer(3)	layer(4)	layer(5)	layer(6)
100	0.958 98	0.947 75	0.924 80	0.915 53
200	0.971 68	0.973 63	0.948 24	0.929 69
300	0.982 42	0.986 33	0.966 80	0.963 38
400	0.987 30	0.989 26	0.981 93	0.965 82
500	0.987 30	0.991 70	0.981 93	0.965 82
600	0.987 30	0.991 70	0.989 75	0.971 68
700	0.984 38	0.991 70	0.989 75	0.977 54
800	0.984 86	0.991 70	0.989 75	0.982 91
900	0.981 93	0.991 70	0.987 30	0.985 35
1 000	0.979 49	0.989 26	0.987 30	0.988 28

表 3 不同隐藏层结构下 DCNN 的训练损失比较

Tab.3 Comparison of DCNN training loss under different hidden layers

迭代次数	训练损失			
	layer(3)	layer(4)	layer(5)	layer(6)
100	0.094 33	0.100 30	0.192 07	0.187 65
200	0.041 56	0.047 42	0.104 10	0.108 50
300	0.020 68	0.026 15	0.058 18	0.066 96
400	0.012 67	0.016 94	0.034 90	0.041 50
500	0.008 84	0.012 09	0.022 90	0.027 74
600	0.006 93	0.009 34	0.016 02	0.019 26
700	0.005 74	0.007 70	0.011 96	0.014 87
800	0.005 16	0.006 60	0.009 10	0.011 62
900	0.004 55	0.005 76	0.006 88	0.009 70
1 000	0.004 26	0.005 26	0.005 38	0.008 22

由图 4 和表 2、表 3 可得，4 种隐藏层结构所建立的 DCNN 模型均具有较好的分类性能，并且 4 种隐藏层结构的训练损失曲线均以不同的速度趋于 0，由此也说明了构建的这 4 种隐藏层结构的卷积网络在学习过程中没有出现过拟合，对于时间序列数据集的学习具有较好的泛化能力. 迭代 100 次时，4 种隐藏层结构的分类精度均达到了 90% 以上，其中 3 层隐藏层结构的分类精度最高达到 95.898%，4 层隐藏层结构尾随其后达到 94.727%；迭代 200 次后，4 层隐藏层结构的分类精度居于第 1，达到 97.314%，并在之后[300，1 000]迭代次数区间中稳定且优于其它 3 种隐藏层结构；迭代次数为 500 次时，达到分类最高精度 99.17%. 据此，本文确定使用 4 层的隐藏层结构用于故障检测的 DCNN 模型中.

3 实验结果与分析

3.1 实验设置

数据集：以某姿态控制系统的飞轮转速数据为例，训练数据集的大小为 140 281 个，其中异常数据值有 35 707 个，测试数据集的大小为 17 077 个，其中异常数据占 8 550 个，实验中进行监督训练时将正常值标签记为 1，异常值标签记为 2.

实验平台：实验采用深度学习平台为 tensorflow1.3.0[18-19]，接口为 python3.5，电脑硬件配置为 i7 处理器，显卡为 GTX1050Ti，8 GB 安装内存，64 位操作系统.

3.2 结果分析

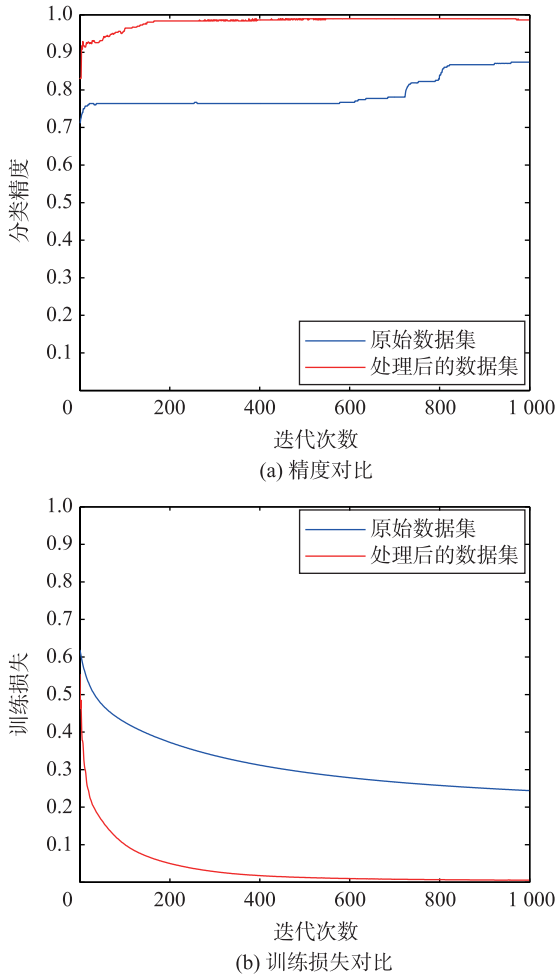


图 5 DCNN 在原始数据集和经过处理后的数据集上的分类性能对比

Fig.5 Comparison of classification performance of DCNN on raw data sets and processed data sets

注 图 5 中的处理后的数据集表示将原始数据集经过第 2 节中的不平衡时间序列数据算法处理后得到的数据集.

图 5 和表 4 展示了 DCNN 模型在原始数据集及处理后的数据集上的检测结果. 图 5(a)中蓝色折线是在原始数据集上经过卷积神经网络学习分类的结果. 当迭代次数达到 600 次时识别精度接近 80%，随后识别精度呈不平稳状态逐渐提高；迭代 1 000 次时，识别精度达到 87.402%. 红色折线是在处理后的数据集上进行卷积学习的结果. 迭代 200 次时，识别精度已经达到 90% 以上；迭代 1 000 次时，识别精度达到 98.633%. 图 5(b)中的蓝色折线是在原始数据集上卷积网络训练时的损失值. 在[0，300]迭代次数区间内，损失值以较快的速率减小；在(300，900]的迭代

次数区间内，损失值减小的速度缓慢；在(900，1 000]迭代次数区间，损失值的大小在0.244 上下浮动并趋于稳定. 据此也说明了在原始数据集上卷积网络学习过于依赖训练数据，导致分类精度较低. 红色折线表示在处理后的数据集上卷积网络训练时的损失值，在[0，300]迭代区间内损失值以较快的速率减小，在(300，1 000]的迭代次数区间内损失值减小为0.03 以内并且仍以较小的速率不断减小，最终在迭代1 000 次时损失值减小为0.005 48，据此也说明了在处理后的数据集上卷积网络对于数据特征学习的高效性. 对于不平衡数据集，学习器学习两种数据后的认知能力不同，会导致少样本类数据的识别能力受限，不平衡时间序列数据处理算法弥补了这样的不足，对于预处理后的数据集，如红色折线所示学习器不论在运行时间还是识别精度都展现出较好的性能.

表4　DCNN 在原始数据集和经过处理后的数据集上的分类性能对比

Tab.4　Comparison of DCNN's classification performance on raw data sets and processed data sets

迭代次数	原始数据集		处理后的数据集	
	Acc	Loss	Acc	Loss
100	0.763 67	0.426 72	0.956 05	0.101 25
200	0.763 67	0.373 03	0.983 40	0.050 31
300	0.763 67	0.337 53	0.986 33	0.028 36
400	0.763 67	0.312 02	0.986 33	0.017 76
500	0.763 67	0.293 06	0.989 26	0.012 67
600	0.767 09	0.278 53	0.989 26	0.010 08
700	0.780 76	0.267 11	0.989 26	0.008 06
800	0.828 61	0.257 93	0.989 26	0.006 91
900	0.867 19	0.250 39	0.989 26	0.006 03
1 000	0.874 02	0.244 08	0.986 33	0.005 48

实验数据集是一个典型不平衡时间序列数据集，本文提出的基于DCNN 模型的不平衡时间序列数据异常检测方法，通过与现有的时间序列数据分类的算法对比，验证了本文所提算法的优越性.

由表5 可得，本文分别在原始数据集和经过本文提出的不平衡时间序列数据处理算法后的数据集上进行，对于原始数据集分别使用PCA和SVD进行特征提取[20]，分别使用SVM[21]和人工神经网络(NN)进行分类，分类精度较低. 其次，在经过不平衡时间序列数据处理后的数据集上使用PCA 和SVD 进行特征提取，分别使用SVM 和NN 进行分类，分类精度明显提高，最后与本文的方法进行对比，体现了本文所提方法的优越性.

表5　不同算法的分类精度

Tab.5　Classification accuracy of different algorithms

数据集	特征提取算法	分类器	分类精度
原始数据	PCA	SVM	0.553 3
		NN	0.721 9
原始数据	SVD	SVM	0.822 5
		NN	0.890 5
经过本文提出的不平衡时间序列数据处理算法后得到的数据集	PCA	SVM	0.724 9
		NN	0.736 7
	SVD	SVM	0.908 3
		NN	0.914 2
	本文的 DCNN 方法		0.986 3

4　总结与展望

随着信息化时代的到来，时间序列数据分析问题引起了广泛的关注，在实际工程中，绝大多数的时间序列数据集都存在数据偏斜的现象. 本文从不平衡时间序列异常检测问题出发，提出了一种基于深度卷积神经网络的偏斜类时间序列数据异常检测方法. 在提出的方法中，对这种典型的不平衡时间序列数据集，首先进行不平衡数据处理，增加少数类样本的代表性数据，缩小数据集类间代表样本数量的差距，提高分类器对于少数类样本集的学习性能. 同时为了提高分类精度选用DCNN 模型作为时间序列数据的分类器，充分发挥了 DCNN 的特征映射能力以及大数据处理的优势.

在本文建立的基于 DCNN 的时间序列异常检测模型中，隐藏层结构采用的是卷积层和池化层相交替的结构，这种结构的优点是在提取特征向量后总是保持特征向量的维度在合适的范围内避免出现过拟合. 在后继研究中，将探索更多不同隐藏层结构的卷积神经网络模型对于时间序列数据检测性能的影响，进一步优化网络结构并提高检测性能.

参考文献

[1] 林智勇，郝志峰，杨晓伟. 不平衡数据分类的研究现状[J]. 计算机应用研究，2008，25(2)：332－336.
Lin Z Y，Hao Z F，Yang X W. Research status of unbalanced data classification[J]. Application Research of Computers，2008，25(2)：332－336.

[2] 王晓娟，郭躬德. 不平衡数据采样方法的对比学习[J]. 微计算机信息，2011(12)：155－157.
Wang X J，Guo G D. Comparative study of unbalanced data sampling method[J]. Microcomputer Information，2011(12)：155－157.

[3] Li J，Li H，Yu J L. Application of random-SMOTE on imbalanced data mining[C]//Fourth International Conference on Business Intelligence and Financial Engineering. Piscataway，NJ，USA：IEEE，2011：130－133.

[4] Chawla N V，Bowyer K W，Hall L O，et al. SMOTE：Synthetic minority over-sampling technique[J]. Journal of artificial Intelligence Research，2002，16：321－357.

[5] 许丹丹，王勇，蔡立军. 面向不均衡数据集的 ISMOTE 算法[J]. 计算机应用，2011，31(9)：2399－2401.
Xu D D，Wang Y，Cai L J. ISMOTE algorithm for unbalanced data sets[J]. Journal of Computer Applications，2011，31(9)：2399－2401.

[6] Cao H, Li X L, Woon Y K, et al. Integrated oversampling for imbalanced time series classification[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(12): 2809 - 2822.
[7] Ding H, Trajcevski G, Scheuermann P, et al. Querying and mining of time series data[J]. Proceedings of the Vldb Endowment, 2008, 1(2): 1542 - 1552.
[8] Keogh E, Kasetty S. On the need for time series data mining benchmarks: A survey and empirical demonstration[J]. Data Mining & Knowledge Discovery, 2003, 7(4): 349 - 371.
[9] Yang Y, Deng Q, Shen F, et al. A shapelet learning method for time series classification[C]//International Conference on TOOLS with Artificial Intelligence. Piscataway, NJ, USA: IEEE, 2017: 423 - 430.
[10] 孙其法, 闫秋艳, 闫欣鸣. 基于多样化 top-k shapelets 转换的时间序列分类方法[J]. 计算机应用, 2017, 37(2): 335 - 340.
Sun Q F, Yan Q Y, Yan X M. Classification of time series based on diversified top-k shapelets conversion[J]. Journal of Computer Applications, 2017, 37(2): 335 - 340.
[11] 周飞燕, 金林鹏, 董军. 卷积神经网络研究综述[J]. 计算机学报, 2017, 40(6): 1229 - 1251.
Zhou F Y, Jin L P, Dong J. An overview of convolutional neural network research[J]. Chinese Journal of Computers, 2017, 40(6): 1299 - 1251.
[12] Cui Z, Chen W, Chen Y. Multi-Scale convolutional neural networks for time series classification[J/OL]. arXiv: 1603.06995. [2018 - 01 - 20]. http://xueshu.baidu.com/s?wd=paperuri%3A%289f44ff3471953143144883e017f17356%29&filter=sc_long_sign&tn=SE_xueshusource_2kduw22v&sc_vurl=http%3A%2F%2Farxiv.org%2Fabs%2F1603.06995&ie=utf-8&sc_us=7768111178099589474.
[13] Wang Z, Yan W, Oates T. Time series classification from scratch with deep neural networks: A strong baseline[C]//International Joint conference on Neural Networks. Piscataway, NJ, USA: IEEE, 2017: 1578 - 1585.
[14] Zhao B, Lu H, Chen S, et al. Convolutional neural networks for time series classification[J]. Journal of Systems Engineering and Electronics, 2017, 28(1): 162 - 169.
[15] He H, Garcia E A. Learning from imbalanced data[J]. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(9): 1263 - 1284.
[16] Lekha L, Suchetha M. Real-time non-invasive detection and classification of diabetes using modified convolution neural network[J]. IEEE Journal of Biomedical & Health Informatics, 2018, 22(5): 1630 - 1636.
[17] Ince T, Kiranyaz S, Eren L, et al. Real-time motor fault detection by 1-D convolutional neural networks[J]. IEEE Transactions on Industrial Electronics, 2016, 63(11): 7067 - 7075.
[18] 黄文坚, 唐源. TensorFlow 实战[M]. 1 版. 北京: 电子工业出版社, 2017.
Hu W J, Tang Y. TensorFlow practical experience[M]. 1st ed. Beijing: Electronics Industry Press, 2017.
[19] Abadi M, Barham P, Chen J, et al. TensorFlow: A system for large-scale machine learning[C]//Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation. Berkeley, CA, USA: USENIX Association, 2016: 265 - 283.
[20] 李慧. 奇异值分解在时间序列分析中的应用[D]. 北京: 北京交通大学, 2009.
Li H. The application of singular value decomposition in time series analysis[D]. Beijing: Beijing Jiaotong University, 2009.
[21] 郑恩辉, 李平, 宋执环. 不平衡数据知识挖掘: 类分布对支持向量机分类的影响[J]. 信息与控制, 2005, 34(6): 703 - 708.
Zheng E H, Li P, Song Z H. Unbalanced data knowledge mining: Influence of class distribution on support vector machine classification[J]. Information and Control, 2005, 34(6): 703 - 708.

作者简介

胡姣姣(1993 -), 女, 硕士生. 研究领域为统计学习.
王晓峰(1966 -), 女, 博士, 教授, 硕士生导师. 研究领域为统计学习, 图像取证与认证.
张　萌(1994 -), 女, 硕士生. 研究领域为统计学习.