

DOI:10.19651/j.cnki.emt.2210658

基于改进 GMM 算法的综合能源数据清洗研究

杨柳林 胡贺骏

(广西大学电气工程学院 南宁 530000)

摘要: 针对数据中台在采集数据过程中会产生异常值的问题,提出一种改进 GMM 算法的数据清洗方法。首先,将边缘计算引入来解决负载过大的问题;其次,为避免 EM 算法计算参数时陷入局部最优解,通过对 GMM 算法中的参数进行优化,改善了陷入局部最优解的缺点。实验结果表明,一定的数据量下,改进的 GMM 算法在召回率、F 值等指标上均优于 GMM-EM 算法。由此可知,改进算法在一定程度上提高了对异常数据的清洗效果,保证数据的可靠性。

关键词: 综合能源;数据中台;数据清洗;高斯混合模型;粒子群算法

中图分类号: TP301 **文献标识码:** A **国家标准学科分类代码:** 520

Research on comprehensive energy data cleaning based on improved GMM algorithm

Yang Liulin Hu Hejun

(School of Electrical Engineering, Guangxi University, Nanning 530000, China)

Abstract: Aiming at the problem of outliers in the process of data collection, a data cleaning method based on improved GMM algorithm is proposed. Firstly, edge computing is introduced to solve the problem of excessive load. Secondly, in order to avoid falling into the local optimal solution when calculating the parameters of EM algorithm, the disadvantages of falling into the local optimal solution are improved by optimizing the parameters of GMM algorithm. The experimental results show that the improved GMM algorithm outperforms the GMM-EM algorithm in terms of recall and F value under a certain amount of data. It can be seen that the improved algorithm can improve the cleaning effect of abnormal data to a certain extent and ensure the reliability of data.

Keywords: integrated energy; data platform; data cleaning; gaussian mixture model; PSO algorithm

0 引言

综合能源系统是指电、水、气等多能源在生产、分配、消费等过程中进行规划和完善后形成的一体化系统,实现不同能源之间的优化调节。随着数据量的日益增加,各种数据系统逐渐融合,综合能源数据激增,由一种数据类型逐渐转变为非结构化、结构化等类型。不同能源的数据也存在差异,电网数据来源多,分布区域广,类型丰富,包括结构化数据、半结构化数据以及非结构化数据^[1];处理数据的速度快,能源的传输和分配在瞬间完成。水务数据的来源复杂,标准不统一,存在一定的数据质量问题,信息整合过程繁杂困难,无法进行有效的数据分析。综合能源系统中都会产生大量的脏数据,所以进行数据清洗,对提高数据质量具有一定意义。

在集中式清洗方法中,连接数据源的各个数据节点将所有数据传输到数据平台,包括数据类型,数据精度和数据

长度等等,数据平台负责处理不同数据节点收集到的数据。Wang 等^[2]提出一种数据清洗方法,在边缘节点采用离群点检测方法,并将在线学习的方式引入来优化模型。分布式清洗方法是在数据采集完成后在数据采集点进行数据清洗,无需将数据统一到数据平台进行处理。Liu 等^[3]提出一种基于 Hadoop 的分布式数据清洗方法,阐述了异构数据的不统一性和反复性,设计数据清洗的流程、架构和算法。Tian 等^[4]设计了一种分布式流数据清洗系统,将基于规则的清洗问题解决。系统对流数据实时监测和完善,凭借有效、密集和分布式的数据结构来完善数据,采用滑动窗口的方法提升数据清洗准确度。Chiang 等^[5]提出一种数据清洗框架,在对隐私数据清洗的同时辨别属性值之间的信息,保证了数据的一致性。刁赢龙等^[6]提出一种配电网负荷数据的清洗方法,在 Hadoop 平台上测试了分布式解决方法,有效的处理负荷数据。由于现实中可能存在噪声数据以及错误数据,首先要对数据中的异常数据进行筛查

收稿日期:2022-07-12

并且去除,以保证数据质量,否则将影响整体的准确性。数据清洗技术分为两类:基于异常点检测和基于机器学习的。异常点检测方法包括基于聚类的异常点检测、基于距离的异常点检测和基于密度的异常点检测^[7]。Bah 等^[8]提出一种基于距离的异常点检测方法,利用提出的微聚类最小探测算法降低查找范围,既保证异常点检验准确度,使计算效率进一步提升。Yang 等^[9]提出一种离群因子检测方法,将改进后的 SOFM 算法进行聚类,采用相对 K 距离领域的熵再次定义离群因子,提升了算法检测精度。Lei 等^[10]提出一种轨道异常点检测方法,利用聚类相似距离与角距离来衡量聚类与轨迹之间的相似度。Wahyuni 等^[11]利用 K-means 方法对异常点数据进行检测,欧氏距离方法获取的结果与标准差极值等相比较。Ismail 等^[12]采用 K-means 聚类与基于密度的空间聚类技术来检测异常值,最后利用曼哈顿距离与欧氏距离进行比较,展示不同距离方法对聚类效果的作用。Su^[13]提出一种非离群对象剔除方法和一种局部离群点检测方法,降低了数据量,在检验准确度和耗时方面都有改进。针对数据清洗还可以采用机器学习的方法,实现对大量数据的清理和修复。Verbraeken 等^[14]指出传统集中式机器学习逐渐无法满足训练数据的需求,阐述其面临的挑战,因此需要将机器学习分别部署在多台机器上。Wang 等^[15]提出一种基于分类能力排序的数据清洗方法,降低了分类错误率和计算复杂度。Shen 等^[16]利用 BP 神经网络建立模型,将辅助点逼近模型与真实值的距离,使用阈值分离正常数据和异常数据。对异常数据值采用另一种阈值判断方法筛选正常数据。Hossen 等^[17]利用随机森林方法提取特征,对数据集进行训练,结果表明线性回归与随机森林准确率一致。Liu 等^[18]改进了随机森林算法,可对缺失数据尤其是块状数据进行插补,提高了数据填补精确度。毕雪婷等^[19]表明对目标聚类之后并完成数据重构,新数据质量高,鲁棒性强。

在传统的数中台中,数据采集点无数据清洗功能,只存在于数据中台,由于数据传输速度的提升以及数据量的暴增,将所有数据集中到数据中台进行处理会影响整体性能,增加中台的计算负荷,中台将无法满足不同增长的需求。集中式清洗方法将数据节点的海量数据发送到数据平台统一处理,传输过程可能由于数据量过多引起数据丢失而导致数据不完整,信息不准确。大数据量传输也会无法满足数据平台实时处理的需求,致使整个清洗过程效率降低。

本文将边缘计算引入,计算任务分散在不同的采集点,减缓了数据中台的计算负载,加快整体的运算速度和效率。采用改进的高斯混合模型进行异常数据的检测,提高了清洗效率,保证数据的可靠性。

1 数据中台架构

数据中台技术是最近几年兴起的数据技术,受到各界

的广泛关注。数据中台包括信息采集、数据清洗、数据治理、数据发布等,其特点是在具有接入、存储、治理、发布的基础上,更加具有实时性的一个数据中台。数据清洗是数据中台中重要的组成部分^[20]。在大数据环境下,尤其是数据的多元化、复杂化,数据清洗技术已存在于交通、医疗等领域。利用数据清洗技术检测并修改脏数据,提高数据质量。数据清洗技术在数据中台的应用,缩短了数据中台的响应时间,提高了数据中台的数据处理速度。

数据采集是整个数据中台的基础,数据采集点需要部署在不同的网络环境中,这就意味着容易受到恶意节点的攻击,导致数据异常、丢失^[21]。数据异常会使传输效率下降,增加能耗,最后将影响数据的应用。因此,数据清洗是数据传输过程中必不可少的部分。

近年来,边缘计算已在物联网等许多技术中发挥重要作用^[22]。边缘计算是在网络终端执行计算的一种计算模式^[23],是存在于数据源与数据中心之间的各种计算,不仅是将数据源中的数据进行操作后上传至数据中心,而且可以计算数据中心的数据,并传输到各个数据源和终端。采用边缘计算增强数据中心的响应能力,降低了传输数据的延迟,缓解了数据中心处理事务的压力,减轻上传数据带来的功耗。

将边缘计算与数据中心、数据采集点相互配合,组成数据清洗架构。架构如图 1 所示。每个数据采集点将不同数据库中的数据、Json 文件、csv 文件和 Excel 文件等收集,传输到边缘节点进行数据清洗,完成对异常数据的处理,之后将数据上传到数据中心。数据清洗采用改进的高斯混合模型(gaussian mixture model, GMM)进行检测,通过不断迭代进行求解。将数据中心的计算压力分散到各个边缘节点,提升了整体的性能。由于边缘节点的存在,提防恶意攻击,提高了整体的可靠性和安全性。

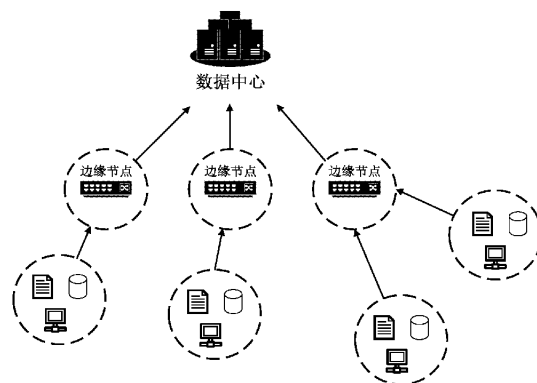


图 1 数据中台架构

2 算 法

2.1 GMM-EM 算法

GMM 是指多个高斯分布函数的组合^[24]。假设随机变量 x , 则高斯模型可以表示为:

$$P(x) = \sum_{k=1}^K \pi_k N(x | \mu_k, \sigma_k) \quad (1)$$

$$\sum_{k=1}^K \pi_k = 1, 0 \leq \pi_k \leq 1 \quad (2)$$

其中, $P(x)$ 表示 x 的联合概率密度函数, $N(x | \mu_k, \sigma_k)$ 为多维正态分布, π_k 表示第 k 个高斯分布的权重系数, μ_k 表示第 k 个高斯分布的均值, σ_k 表示第 k 个高斯分布的协方差矩阵。采用最大对数似然函数来求解 GMM 中的参数。

为求解 GMM 中的参数, 采用最大期望算法(expectation maximization algorithm, EM) 不断更新获取结果。引入一个隐含变量 z , 含义为随机的数据点 x_i 属于其中一个高斯分布的概率。数据点 x_i 为第 k 个高斯分布的概率为:

$$\gamma = \frac{\pi_k N(x | \mu_k, \sigma_k)}{\sum_{j=1}^K \pi_j N(x | \mu_j, \sigma_j)} \quad (3)$$

求解 GMM 中的参数:

$$\mu_k = \frac{\sum_{n=1}^N \gamma x_n}{\sum_{n=1}^N \gamma} \quad (4)$$

$$\sigma_k = \frac{\sum_{n=1}^N \gamma (x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^N \gamma} \quad (5)$$

$$\pi_k = \frac{\sum_{n=1}^N \gamma}{N} \quad (6)$$

最后计算对数似然函数:

$$L(x | \pi, \mu, \sigma) = \sum_{n=1}^N \log(\sum_{k=1}^K \pi_k N(x_n | \mu_k, \sigma_k)) \quad (7)$$

检查对数似然函数是否收敛, 若不收敛, 再次重复上述步骤。

$$L(x | \pi^{(t)}, \mu^{(t)}, \sigma^{(t)}) - L(x | \pi^{(t-1)}, \mu^{(t-1)}, \sigma^{(t-1)}) < \epsilon \quad (8)$$

其中, $L(x | \pi^{(t)}, \mu^{(t)}, \sigma^{(t)})$ 表示本次迭代的似然函数值, $L(x | \pi^{(t-1)}, \mu^{(t-1)}, \sigma^{(t-1)})$ 表示上次迭代的函数值。

2.2 σ_k 优化

使用 EM 算法求解对数似然函数的最大值, 容易陷入局部最优解。因此, 本文采用改进的 GMM 算法来求解。粒子群算法^[25](particle swarm optimization, PSO) 是一种全局优化算法, 利用该算法对 σ_k 进行优化。PSO 算法是效仿鸟类捕食行为而设计的。当鸟类寻找食物时, 其策略就是搜寻当前距离食物最近的鸟的周围。将鸟类抽象为粒子, 则每个粒子的速度 $v_{k+1,i}$ 和位置 $x_{k+1,i}$ 更新公式为:

$$v_{k+1,i} = \omega v_{k,i} + c_1 R_1 (x_{pbest,i} - x_{k,i}) + c_2 R_2 (x_{gbest,i} - x_{k,i}) \quad (9)$$

$$x_{k+1,i} = x_{k,i} + v_{k+1,i} \quad (10)$$

其中, ω 为惯性因子, 视作常数。 c_1 和 c_2 为学习因子, 视作常数, R_1 和 R_2 为 $[0, 1]$ 之间的随机值, $x_{pbest,i}$ 表示第 i 个粒子最优位置的值, $x_{gbest,i}$ 表示整个种群中最优位置的值, $x_{k,i}$ 表示第 i 个粒子在第 k 次迭代的位置, $v_{k,i}$ 表示第 i 个粒子在第 k 次迭代的速度。为了求解 σ_k , 采用对数似然函数的负值作为目标函数, 利用目标函数对 $x_{pbest,i}$, $x_{gbest,i}$ 进行更新。

PSO 算法中的惯性因子 ω 使粒子保持运动惯性, 惯性因子的值越大, 全局寻优能力越强, 惯性因子的值越小, 局部寻优能力越强, 所以惯性因子的公式如下:

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \frac{t}{T} \quad (11)$$

其中, ω_{\max} 表示惯性因子的最大值, ω_{\min} 表示惯性因子的最小值。 t 表示当前迭代次数, T 表示最大迭代次数。

采用文献[26]中的方法, 保证在迭代过程中协方差矩阵是正定矩阵。采用循环雅克比算法和 QR 分解算法计算协方差矩阵, 并利用特征值和角度来表示。 σ_k 可以表示为 $V \lambda^T V$, 其中 $\lambda = (\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n)$, $\lambda_i > 0$ 表示对角矩阵, $V = (v_1, v_2, v_3, \dots, v_n)$ 表示特征向量。利用 QR 分解算法来得到 V , 通过旋转矩阵来完成 QR 分解。旋转矩阵可表示为:

$$G = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cos(\varphi^{ij}) & \cdots & \sin(\varphi^{ij}) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & -\sin(\varphi^{ij}) & \cdots & \cos(\varphi^{ij}) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \quad (12)$$

旋转矩阵的非零元素为:

$$\begin{cases} G_{ii} = \cos(\varphi^{ij}) \\ G_{ij} = \sin(\varphi^{ij}) \\ G_{ji} = -\sin(\varphi^{ij}) \\ G_{jj} = \cos(\varphi^{ij}) \\ G_{mm} = 1, m \neq i, j \end{cases} \quad (13)$$

其中, φ^{ij} 位于区间 $[-\frac{\pi}{2}, \frac{\pi}{2}]$ 内。所以协方差矩阵可以由参数 $(\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n; \varphi^{1,2}, \dots, \varphi^{ij}, \dots, \varphi^{(n-1),n})$ 表示。改进 GMM 算法流程如图 2 所示。

3 仿真实验

实验环境为 Intel(R) Core(TM) i5-7200U CPU @ 2.50 GHz 2.70 GHz, 12 GB 内存, Window10 操作系统。采用 Python 语言实现优化算法, 软件环境: Python3.7。部分样本数据如表 1 所示。

以不同用户 1 个月的用水量作为样本数据。其中序号表示用户编号, 水量表示 1 个月实际的用水量。采用 3 500 条

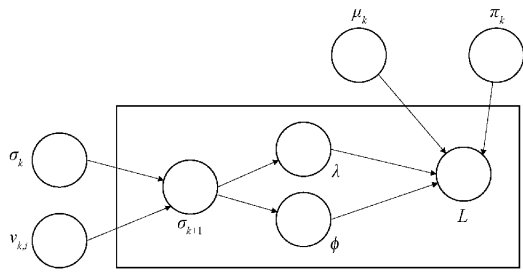


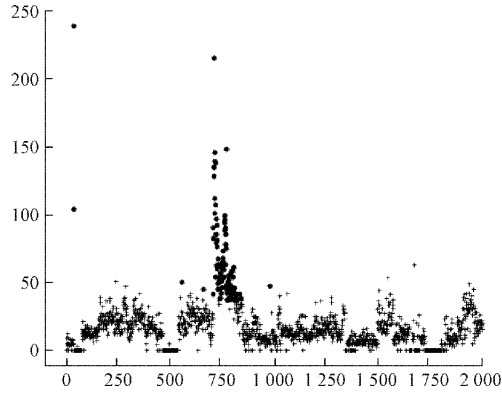
图 2 算法流程

表 1 部分样本数据

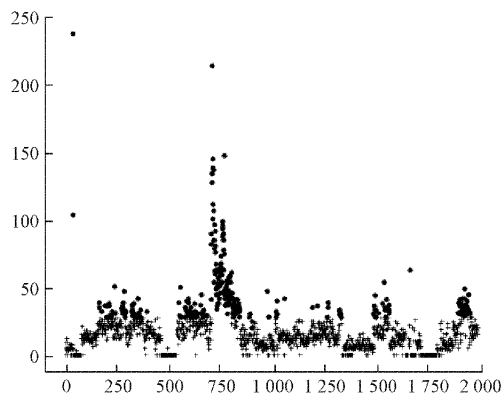
序号	1	2	3	4	5	6	7	8	9	10
水量	9	0	0	5	3	5	4	12	0	9

数据作为测试样本。

在实验中,设置学习因子 c_1 为 1.49, c_2 为 1.49, 惯性因子的最大值 ω_{max} 为 0.9, 最小值 ω_{min} 为 0.4, PSO 算法迭代次数为 100。在实际的数据采集中, 数据采集点受到外界因素的干扰, 会对数据产生较小的影响, 可将其看为正常数据处理。当数据受到大扰动的影响, 则会产生较大偏差, 所以将聚类数设置为 2。图 3 中加号表示正常的的数据, 实心圆表示异常数据。如图 3(a)所示是采用 GMM-EM 算法进行异常值清洗的结果。如图 3(b)所示是采用改进的



(a) GMM-EM 算法清洗效果



(b) 改进算法清洗效果

图 3 改进算法与 GMM-EM 算法比较

GMM 算法进行异常值清洗的结果。通过对比可以看出, 改进算法可以更好的将正常数据与异常数据分离。

实验评价指标采用召回率 (Recall)、F 值^[27] 和运行时间来衡量。将异常数据设置为正类, 并且根据此定义来计算召回率和 F 值。

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (15)$$

其中, TP 表示清洗之后异常数据的数据量, FN 表示未检测出异常数据的数据量, $Precision$ 表示准确率。

如图 4~6 所示为对不同数据量运行 20 次后, 取平均召回率、F 值和运行时间进行比较。由图 4 和 5 可以看出, 改进算法的召回率和 F 值要高于 GMM-EM 算法, 特别是当数据量为 2 000 时, 本文算法的召回率要明显优于 GMM-EM 算法, 而数据量为 1 500 时, 改进算法的 F 值达到最大值。由图 6 可以看出改进算法的运行时间要少于 GMM-EM 算法的。随着数据量的增大, 故改进算法与 GMM-EM 算法的运行时间都会变大, 其中 GMM-EM 算法的运行时间变化较大。

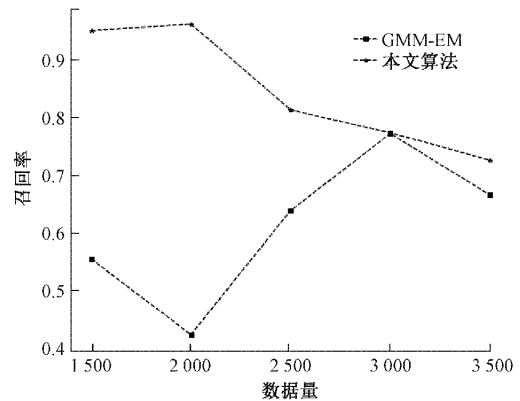


图 4 改进算法与 GMM-EM 算法召回率对比

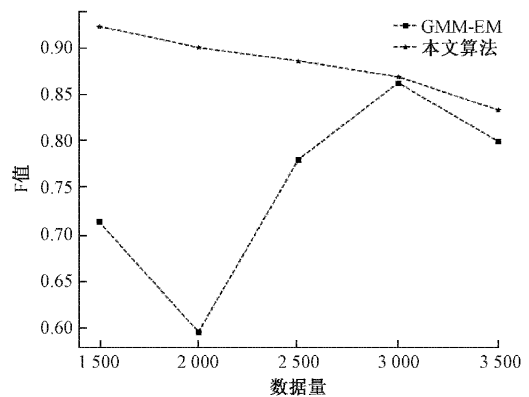


图 5 改进算法与 GMM-EM 算法 F 值对比

表 2 所示为不同算法的指标对比。从表中可知, 与 GMM 算法相比, 在不同的数据量中, 改进 GMM 算法的召

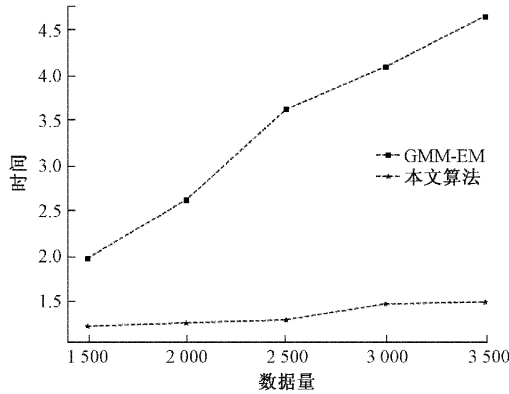


图 6 改进算法与 GMM-EM 算法运行时间对比

回率和 F 值都有所增加,并且计算所消耗的时间均有所减少。

表 2 不同算法的指标对比

指标	算法	数据量				
		1 500	2 000	2 500	3 000	3 500
召回率	改进 GMM	0.951	0.963	0.813	0.773	0.726
	GMM	0.553	0.422	0.638	0.772	0.665
F 值	改进 GMM	0.922	0.899	0.885	0.868	0.832
	GMM	0.712	0.594	0.779	0.861	0.799
运行时间	改进 GMM	1.219	1.256	1.289	1.467	1.490
	GMM	1.970	2.619	3.619	4.090	4.650

4 结 论

通过各个数据采集点收集数据,并采用边缘计算进行数据处理,来减轻数据中心的运行压力。因此提出一种数据清洗方法,采用改进 GMM 算法来检测异常数据,有利于提高整个系统的运行效率。实验表明,将改进后的 GMM 算法与 GMM-EM 算法对比,证明了改进后算法的有效性,在一定程度上提高了对异常数据的清洗效果,保证了数据的可靠性,并且改进算法平均运行时间更短。在改进算法中,若 PSO 算法的迭代次数较少,会导致协方差矩阵不是正定矩阵,使计算无法进行。下一步将针对这些问题进行研究。

参考文献

[1] 刘科研,盛万兴,张东霞,等. 智能配电网大数据应用需求和场景分析研究[J]. 中国电机工程学报, 2015, 35(2):287-293.

[2] WANG T, KE H, ZHENG X, et al. Big data cleaning based on mobile edge computing in industrial sensor-cloud [J]. IEEE Transactions on Industrial Informatics, 2020, 16(2): 1321-1329.

[3] LIU X, GUO A, SUN T. Application of hadoop based distributed data cleaning technology in periodical meta data integration [C]. 2017 10th International

Symposium on Computational Intelligence and Design (ISCID), 2017, DOI:10.1109/ISCID.2017.176.

[4] TIAN Y, MICHIARDI P, VUKOLIĆ M. Bleach: A distributed stream data cleaning system[C]. 2017 IEEE International Congress on Big Data(BigData Congress), 2017, DOI:10.1109/BigDataCongress.2017.24.

[5] CHIANG F, GAIROLA D. InfoClean: Protecting sensitive information in data cleaning[J]. J. Data and Information Quality, 2018, 9(4):26.

[6] 刁赢龙,盛万兴,刘科研,等. 大规模配电网负荷数据在线清洗与修复方法研究[J]. 电网技术, 2015, 39(11): 3134-3140.

[7] MANDHARE H, IDATE S. A comparative study of cluster based outlier detection, distance based outlier detection and density based outlier detection techniques[C]. 2017 International Conference on Intelligent Computing and Control Systems(ICICCS), 2017, DOI: 10.1109/ICCONS.2017.8250601.

[8] BAH M, WANG H, HAMMAD H, et al. An effective minimal probing approach with micro-cluster for distance-based outlier detection in data streams[J]. IEEE Access, 2019, DOI: 10.1109/ACCESS.2019.2946966.

[9] YANG P, WANG D, WEI Z, et al. An outlier detection approach based on improved self-organizing feature map clustering algorithm[J]. IEEE Access, 2019, DOI: 10.1109/ACCESS.2019.2922004.

[10] LEI B, MINGCHAO D. A distance-based trajectory outlier detection method on maritime traffic data[C]. 2018 4th International Conference on Control, Automation and Robotics (ICCAR), 2018, DOI: 10.1109/ICCAR.2018.8384697.

[11] WAHYUNI E, SURONO S, ELIYANTO J. Outlier detection using K-means clustering with minkowski-chebyshev distances for inquiry-based learning results in students dataset[C]. 2021 International Conference on Artificial Intelligence and Big Data Analytics, 2021, DOI:10.1109/ICAIBDA53487.2021.9689736.

[12] ISMAIL Z, CHUN A, SHAPIAI RAZAK M. Efficient herd-outlier detection in livestock monitoring system based on density-based spatial clustering[J]. IEEE Access, 2019, DOI: 10.1109/ACCESS.2019.2952912.

[13] SU S. An efficient density-based local outlier detection approach for scattered data[J]. IEEE Access, 2019, DOI: 10.1109/ACCESS.2018.2886197.

[14] VERBRAEKEN J, WOLTING M, KATZY J, et al. A survey on distributed machine learning[J]. ACM

- Comput, 2020, DOI: org/10.1145/3377454.
- [15] WANG Y, PAN Z, PAN Y. A training data set cleaning method by classification ability ranking for the k-nearest neighbor classifier[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 31(5): 1544-1556.
- [16] SHEN Y, DONG S, WU Q, et al. Data cleaning for wind turbine systems based on iterative learning and neural network[C]. 2021 33rd Chinese Control and Decision Conference (CCDC), 2021, DOI: 10.1109/CCDC52312.2021.9601788.
- [17] HOSSEN J, JESMEEN M, SAYEED S. Modifying cleaning method in big data analytics process using random forest classifier [C]. 2018 7th International Conference on Computer and Communication Engineering (ICCCE), 2018, DOI: 10.1109/ICCCE.2018.8539254.
- [18] LIU J, CAO Y, LI Y, et al. A big data cleaning method based on improved CLOF and Random Forest for distribution network[J]. CSEE Journal of Power and Energy Systems, 2020, DOI: 10.17775/CSEEJPES.2020.04080.
- [19] 毕雪婷,刘小军,邵文远. 基于聚类方法的自动驾驶场景下的三维目标检测[J]. 电子测量技术, 2021, 44(6): 103-107.
- [20] 郝爽,李国良,冯建华,等. 结构化数据清洗技术综述[J]. 清华大学学报(自然科学版), 2018, 58(12): 1037-1050.
- [21] 吴信东,董丙冰,堵新政,等. 数据治理技术[J]. 软件学报, 2019, 30(9): 2830-2856.
- [22] KHAN W, AHMED E, HAKAK S, et al. Edge computing: A survey[J]. Future Generation Computer Systems, 2019, DOI: org/10.1016/j.future.2019.02.050.
- [23] SHI W, CAO J, ZHANG Q, et al. Edge computing: Vision and challenges[J]. IEEE Internet of Things Journal, 2016, 3(5): 637-646.
- [24] LI R, WANG Z, GU C, et al. A novel time-of-use tariff design based on Gaussian mixture model [J]. Applied Energy, 2016, DOI: org/10.1016/j.apenergy.2015.02.063.
- [25] 张颖,杨广媛. 基于FW-PSO算法优化无线传感网络拓扑结构的方法[J]. 电子与信息学报, 2021, 43(2): 396-403.
- [26] ARI C, AKSOY S. Maximum likelihood estimation of gaussian mixture models using particle swarm optimization[C]. 2010 20th International Conference on Pattern Recognition, 2010, DOI: 10.1109/ICPR.2010.188.
- [27] 杜岳峰,申德荣,聂铁铮,等. 基于关联数据的一致性和时效性清洗方法[J]. 计算机学报, 2017, 40(1): 92-106.

作者简介

杨柳林, 博士, 讲师, 主要研究方向为电力系统分析、大数据分析及应用。

E-mail: yangll@gxu.edu.cn

胡贺骏, 硕士研究生, 主要研究方向为数据清洗。

E-mail: 1360502391@qq.com