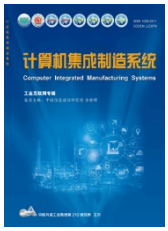

计算机集成制造系统
Computer Integrated Manufacturing Systems
ISSN 1006-5911,CN 11-5946/TP

《计算机集成制造系统》网络首发论文

题目：基于 MOPSO 算法改进的异常点检测方法
作者：高勃，柴学科，朱明皓
收稿日期：2021-12-29
网络首发日期：2022-06-21
引用格式：高勃，柴学科，朱明皓．基于 MOPSO 算法改进的异常点检测方法[J/OL]．计算机集成制造系统.
https://kns.cnki.net/kcms/detail/11.5946.TP.20220620.1824.008.html

网络首发时间：2022-06-21 18:00:07
网络首发地址：https://kns.cnki.net/kcms/detail/11.5946.TP.20220620.1824.008.html

基于 MOPSO 算法改进的异常点检测方法

高 勃[1]，柴学科[1]，朱明皓[2+]
（1.北京交通大学 计算机与信息技术学院，北京 100044; 2.北京交通大学经济管理学院，北京 100044）

摘 要：挖掘工业大数据的隐含价值是智能制造的一个重要研究方向，针对工业大数据特点开展异常点检测是实现数据分析的前提。首先，介绍了工业大数据异常点检测解决的主要问题，提出本文的相关定义。其次，基于多目标粒子群算法(MOPSO)，提出一种工业大数据异常点检测的改进 DBSCAN 模型，介绍了模型的算法设计思想、算法步骤，完成了算法伪代码的编写，并提出了算法时间复杂度的计算方法。最后，通过某电芯工厂制造数据，进行了模型仿真与实验，经试验验证，本文提出的模型提高了工业大数据异常点检测的准确率。本文为数据挖掘在工业异常点检测中的应用提供参考。

关键词：工业大数据；异常点检测；多目标粒子群算法；DBSCAN 模型
文献标识码：A

Outlier detection model modified based on MOPSO algorithm

GAO Bo[1], *CHAI Xueke*[1], *ZHU Minghao*[2+]
(1.School of Computer and Information Technology, Beijing Jiaotong University，Beijing 100044, China;2. School of Economics and Management, Beijing Jiaotong University，Beijing100044, China;）

Abstract: Mining the implied value of industrial big data is an important research direction of intelligent manufacturing, and carrying out outlier detection is a prerequisite for realizing data analysis. Firstly, the main problems addressed by industrial big data anomaly detection are introduced, and the relevant definitions in this paper are proposed. Secondly, based on the multi-objective particle swarm algorithm (MOPSO), an improved DBSCAN model for industrial big data outlier detection is proposed, the algorithm design idea and algorithm steps of the model are introduced, the pseudo-code of the algorithm is completed, and the calculation the time complexity of the algorithm is proposed. Finally, the model simulation and experiments are carried out by using the manufacturing data of an electric core factory, and it is verified that the model proposed in this paper has improved the accuracy of industrial big data outlier detection. This paper provides a reference for the application of data mining in industrial outlier detection.

Keywords: industrial big data; outlier detection; multi-objective particle swarm algorithm; DBSCAN model

1 引言

数据挖掘是指从数据中发现知识，其目的是从大量数据中发现有价值的隐含信息[1]。当前，数据采集技术已经成熟应用，直接推动了工业数据来源、类型、容量的爆炸式增长，从工业大数据中研究数据隐含的信息价值已经成为智能制造的一个重要研究方向。由于工业生产特性，其多源异构数据中存在一些数值明显偏离其余熟知的样本点，即异常值，也称为离群点[2]。这些异常值会导致数据整体质量差，无法反应工业本身特征，因此在开展工业数据特征工程研究前，需要通过剔除异

收稿日期：2021-12-29；修订日期：2022-04-08。Received 29 Dec. 2021; accepted 08 Apr. 2022.

常值、缺失值填补、不处理等异常值处理方法[3]进行处理。异常数据对整体数据几乎没有影响时，直接剔除异常值这种方法比较有效[4]；缺失值填补包括统计学方法和机器学习方法两类[5]，统计学方法代表性的有 EM（Expectation Maximization）填充算法、回归分析法、多重插补等[6]，机器学习方法的代表性方法有：K 最近邻填补、K-means 填补等。

异常值检测是异常值处理的前提。基于模型的传统数学统计方法通过分布假设，创建概率分布模型，将数据对象与该模型进行拟合后判断数据是否为异常值，但该方法不适用于多源异构数据。而基于距离的异常点识别有效的解决了高维数据集和无法预知数据集分布情况的问题[7]，针对不同场景下的数据，选定一个距离进行度量，使用距离阈值作为准则进行异常值检测，如，k 近邻（k-Nearest Neighbor，KNN）[8]方法、反向 k 近邻（Reverse k-Nearest Neighbor，RKNN）[9-10]，实验证明，该方法对局部异常数据的检测效果不高。基于密度的异常点识别方法[11]综合考量了数据的分布稀疏程度，解决了对局部异常数据的检测问题，对数据分布不均匀的数据集的检测效果大幅提高，如，局部离群因子（LOF， Local Outlier Factor）算法[12]为每一个点指定一个局部离群因子，离群因子越大，表明其为异常值的概率越大。通过引入反向 k 近邻的 INFLO（Influenced Outlierness）[13-14]算法，进一步提高了算法的准确性。针对回归模型和 ARMA 模型研究新的异常值检验的统计量，解决了常规异常值检测方法中出现的"遮蔽现象"[15]。

基于聚类的离群点检测方法也是异常点方法中的一个重要分支，一些规模较小的簇中的数据对象或不属于任何簇的数据对象，为异常值[16]。1996 年 Martin Ester 等人提出 DBSCAN 算法[17]，该算法属于密度聚类算法的一种，在数据处理方向得到了广泛应用，算法的核心是设置两个参数 Eps 和 MinPts，这两个参数对结果的影响具有很大的敏感性。国内外很多学者都针对参数设计进行了 DBSCAN 算法改进，VDBSCAN 算法它使用 k-dist 图对不同密度选择合适的参数[18]，AA-DBSCAN 算法提高了数据密度不均的数据集的聚类效果[19]，RNN-DBSCAN 算法观察最近邻图遍历的方式来进行聚类，并以逆近邻计数来表示数据密度的估计，提高密度不均的数据集的聚类效果[20]。借鉴图的强连通分量定义 MinPts 邻域，自适应地根据数据对象的密集程度自动调整邻域大小提高聚类效果[21]，GRDBSCAN 聚类算法依据网格和密度比的计算来提高密度分布不均匀的簇的聚类效果[22]，自适应确定 DBSCAN 参数算法[23]根据数据集的分布特性，生成多个 Eps 参数和 MinPts 参数的集合，然后两两组合进行聚类分析，当聚类结果趋于稳定时，则对应的参数为最终的参数值。

针对工业大数据领域，异常值检测更加重要，降低时间复杂度，提高预测精度能够更好的挖掘数据价值。基于改进密度峰值聚类算法[24]的异常值检测方法通过避免聚类过程，降低时间复杂度。

基于 R-tree 的数据检测算法[25]通过通过优先扫描具有高异常值度的数据点，大大减少了检索空间，一次处理中可以处理多个检测任务。原型分析（ADA）作为一种无监督统计技术，可以识别数据云外围的极端观测[26]。与数据大小相比，维度带来更大的挑战，传统的异常值检测方法在高维数据处理中会面临失败的情况[37]，从高维空间中生成候选子空间，对每个子空间进行潜在异常值识别[28]对于解决高维度问题是一种探索。

但是，传统基于 DBSCAN 算法的异常点检测参数 Eps 和 MinPts 的设置比较复杂，需要不断进行测试，而且由于参数的全局性，密度大的数据点的 Eps 与密度小的数据点 Eps 相同，因此统一的全局 Eps 会淹没部分离群点，无法有效的检测异常点。工业大数据来源与具体工业领域紧密相关，随着工业领域的复杂会产生各种各样的异常点数据，不同异常点数据的密度差异较大，若使用统一的全局 Eps 值，对一些异常数据将无法有效的进行检测。鉴于此，本文针对工业生产数据特点，提出了一种基于 MOPSO 优化的 DBSCAN 异常值检测算法，该算法利用 MOPSO 算法为数据集中的每个数据点定义一个不同的 Eps，这样不仅考虑了数据集密度不均匀的问题，也解决了参数设定困难的问题，当真实验结果表明，该算法取得了较好的效果。

2 问题描述与相关定义

2.1 问题描述

在企业生产制造经营过程中，由于周围环境的变化、人为操作不当等原因，会导致传感器采集到了一些异常数据、不完整数据，如图 1 所示，A、B、C 显著偏离，是典型的异常数据，需要对这些数据进行清理，但是，这些异常数据并非都是无用数据，一些异常数据为工业故障数据，对工业过程的故障分析有巨大的意义，因此，在工业异常数据检测过程中，不能直接采用剔除处理的方法，需要识别出不同原因产生的异常点，从而将真正的错误数据剔除。

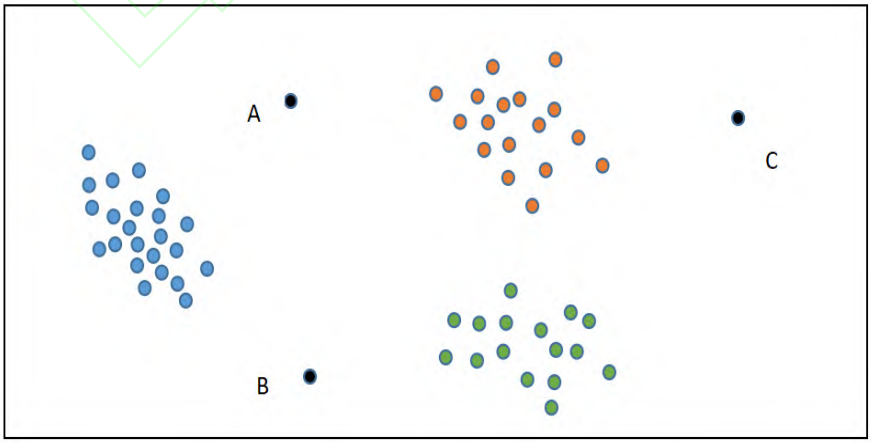

图 1 异常数据点

2.2 相关定义

定义 1 相关系数：相关关系是一种非确定性的关系，相关系数是研究变量之间线性相关程度的量，较为常用的是皮尔逊相关系数。

定义 2 互信息：互信息是信息论里面一种信息度量方式，指的是两个随机变量之间的关联程度，它可以看成是一个随机变量中包含的关于另一个随机变量的信息量，或者说是一个随机变量由于已知另一个随机变量而减少的不肯定性。互信息用于表示信息之间的关系，是两个随机变量统计相关性的测度。

定义 3Eps、MinPts：Eps 表示的是距离阈值，即表示以某一个数据样本为中心点，以 Eps 大小为半径的邻域；MinPts 表示以某一个数据样本为中心，Eps 邻域内数据点的个数阈值。

定义 4　Eps 邻域：给定对象半径为 Eps 内的区域称为该对象的 Eps 邻域。如果给定对象 Eps 邻域内的样本点个数大于等于 MinPts，则称该对象为核心点；非核心点但在某核心点的 Eps 邻域内的对象称为边界点。

定义 5　直接密度可达：对于样本集合 D，如果样本点 q 在 p 的 Eps 邻域内，并且 p 为核心对象，那么对象 q 从对象 p 直接密度可达。

定义 6　密度可达：对于样本集合 D，给定一串样本点$p_1,p_2 \ldots p_n$，p=p_1,q=p_n,若p_i∈D(1<i≤n), 且对象p_i从p_{i-1}直接密度可达，那么对象 q 从对象 p 密度可达。

定义 7 密度相连：若存在对象 r∈D，使得对象 p 和 q 是从 r 密度可达的，那么称对象 p 和 q 密度相连，密度相连是对称的。

3 算法介绍

针对工业大数据异常点检测的特点，本文首先使用 DBSCAN 算法对数据集进行聚类分析，解决数据量局大的问题，筛选出数据集中最可能成为异常点的数据，然后，构建基于 MOPSO 优化的 DBSCAN 算法，为数据集中的每个数据求解一个 Eps 值，解决了全局参数淹没部分异常点的问题，找出真正的异常点。

3.1 算法设计

3.1.1 传统 DBSCAN 算法预处理

使用传统基于全局参数的 DBSCAN 算法对工业数据集进行聚类预处理，筛选可能为异常点的数据，为本文提出的算法处理数据集降低时间消耗量，提高算法的效率。

3.1.2 数据变量相关性分析

对数据集中的任意两两属性进行关联分析，得到关联性最大的两个属性。将关联性最强的两个

属性筛选出，进行数据集的重新构建出两个数据集，使用本文算法对两个数据集分别进行异常值检测，根据检测结果进行分析，分析两关联性最强的属性所在的数据是否同时异常，如果同时异常，则为工业故障数据，可以使用这些数据进行工业故障分析，否则，为记录错误或者传感器采集故障数据，需剔除这类无效数据。

3.1.3 Eps 优化参数设计

使用 MOPSO 算法为数据集中的每个数据点求解一个 Eps，在该模型中，算法为数据集中的各个点使用 k-dist 曲线初始化一组 Eps 解，随后的迭代过程进行更新，不断更新粒子的位置，以此寻找最优解，Eps 优化模型如下所示：

稀疏值： x_i 与周围 MinPts 的点的邻域半径之差的绝对值之和的倒数为稀疏值，如果 x_i 与周围 MinPts 的点的邻域半径之差的绝对值之和越大，则说明 x_i 与周围点的差异越大，越可能成为异常点，则 F_1 的值越小。

$$F_1 = \frac{1}{\sum_{x_j \in D_1} abs(Eps_i - Eps_j) + 1} \tag{1}$$

公式（1）中，D_1 表示 x_i 的最近 MinPts 个点的邻域，$abs(Eps_i - Eps_j)$ 表示 x_i 与 x_j 半径差值的绝对值。

离群值： x_i 与其半径 Eps 内的点的距离之和的倒数为离群值， x_i 与其半径 Eps 内的点的距离之和越大， x_i 越可能成为异常点， F_2 值越小。

$$F_2 = \frac{1}{\sum_{x_j \in D_2} Dis\tan ce(x_i, x_j) + 1} \tag{2}$$

公式（2）中， $Dis\tan ce(x_i, x_j)$ 表示 x_i 与 x_j 的距离， D_2 表示 x_i 的 Eps 邻域。

表 1 Eps 求解伪代码

输入：
数据集 dataSetD
最大迭代次数 gMax
1: Initialize the position and velocity of the swarm particle swarm #初始化粒子群的位置和速度

2: Calculate the target values according to formulas 1 and 2

3: Calculate the Archive set　#根据 pareto 支配原则，计算得到 Archive 集(存放当前的非劣解)

4:　while Number of iterations <gMax　#迭代次数不超过 gMax

5:　　for　particle p in the swarm　　#粒子群的每个粒子

6:　　Select gBest in the Archive set　#在 Archive 集选择 gbest(全局最优值）

7:　　Update　velocity,　position, and　function values of p　#更新粒子速度、位置、目标函数值

8:　　Update pBest

9:　　if　the values of p better than pBest。#判断当前解的优劣

10:　　Set pBest is the values of p

11:　　else:　the values of p cannot be compared with pBest

12:　　the values of p and pBest are randomly selected as the pBest

13:　　Update the Archive collection #更新 Archive 集

14:　endfor

15: End

输出：每个数据点的 Eps 值

3.1.4 基于 MOPSO 的 DBSCAN 改进算法设计

设原始数据集中包含了 n 个数据点，为 $X_1, X_2, \cdots, X_n$，使用 MOPSO 算法求解时产生粒子规模为 n 的粒子群，则第 i 个粒子代表了数据点 X_i 的 Eps 值，即 Eps_i，即 MOPSO 算法中粒子的位置值 x_i 表示第 i 个粒子的 Eps 值，随后不断迭代更新每个粒子的位置值，最终得到各个粒子对应的 Eps 参数值。迭代过程如下：

（1）初始化 DBSCAN 中每个数据点的 Eps 的解，即粒子的位置值 x_i，同时使用随机函数初始化每个粒子对应的速度值 v_i。

每个粒子的 Eps 值的初始化方法为：首先，计算出每个数据点到其余各数据点的距离，将距离升序排列，找到每个数据点的第 MinPts 个点的距离，设第 i 个数据点对应的值为 d_i；然后找到 n 个

数据点对应的第 MinPts 个点的距离中的最小值 $d_{\min}$ 和 $d_{\max}$，进而对粒子群中的解 Eps_i 进行赋值。

$$Eps_i = d_{\min} + rand * (d_{\max} - d_{\min}) \tag{3}$$

公式（3），中 rand 表示一个随机数，是使用随机数函数生成的一个值。

使用公式(1)和公式(2)计算出每个粒子的稀疏值和离群值，即得到粒子的目标值，然后依据 Pareto 支配原则对解进行支配关系比较，即比较粒子间的目标解，若一个粒子 p 的目标解中稀疏值和离群值均比另一个粒子 q 对应的值小，则判定粒子 p 的解优于粒子 q 的解，即粒子 p 的解为非劣解。将粒子群中的所有非劣解拷贝到非劣解集合中，即得到 Archive 集。

Archive 集中包含了粒子中的所有非劣解，粒子的全局最优值从 Archive 集中取得。

MOPSO 算法中选取粒子的全局最优值方法为：首先，计算 Archive 集中粒子的密度信息，将目标空间用网格等分成若干个小区域，以每个区域中包含的粒子数作为粒子的密度信息，粒子的密度值与其所在网格中包含的粒子数成正比；然后，为群体中的粒子选取一个 gbest(全局最优值），通过 Archive 集中粒子的密度信息作为选择依据。粒子的选择概率与该粒子的密度值成反比，即密度值越高，该粒子被选择的概率越低。

在第 j 次迭代过程中选择粒子的全局最优值的具体实现为：

Step1：计算在第 j 次迭代更新过程时目标空间的边界，即求出所有粒子的稀疏值中的最小值和最大值，记做 $\min F_1^j$ 和 $\max F_1^j$，以及所有粒子的离群值中的最小值和最大值，记做 $\min F_2^j$ 和 $\max F_2^j$。则使用网格将目标空间划分为若干个小区域时，$(\min F_1^j, \max F_1^j)$ 和 $(\min F_2^j, \max F_2^j)$ 即为边界。

Step2：计算网格划分后的每个小区域的长、宽值，根据网格空间的边界值和网格空间中包含的小区域的总数，计算得到网格小区域的模为：

$$\Delta F_1^j = (\min F_1^j, \max F_1^j) / M \ ， \ \Delta F_2^j = (\min F_2^j, \max F_2^j) / M \tag{4}$$

公式（4）中，M 为网格中包含的区域总数，由人为设置，本文取 M=50。

Step3：遍历第 j 次迭代过程所对应的非劣解集 Archive ，计算出每个粒子在网格空间的编号。对于第 i 个粒子，其所对应的网格编号为：

$$(\lfloor (F_1^i - \min F_1^j) / \Delta F_1^j \rfloor + 1, \lfloor (F_2^i - \min F_2^j) / \Delta F_2^j \rfloor + 1) \tag{5}$$

公式（5）中，[x]为向下取整函数，表示小于 x 的所有整数中的最大值。

Step4：计算网格的信息和粒子的密度估计值，根据公式（5）中计算得到 Archive 集中所有粒子的网格编号，则可以计算出各个网格中粒子的数目，即各个网格中非劣解的个数。由于粒子的选择概率与该粒子的密度值成反比，因此找到包含非劣解最少的网格编号，进而从该网格中随机选择一个非劣解作为本次迭代过程中的全局最优值。

（2）通过个体最优值与全局最优值更新粒子的速度值与位置值，得到粒子的个体最优值。粒子的速度更新公式如下：

$$v_{i+1} = \omega * v_i + c_1 * rand() * (pbest_i - x_i) + c_2 * rand() * (gbest_i - x_i) \tag{6}$$

公式（6）中，第一项为惯性保持部分，表示上次速度值对本次速度值的影响；第二项为自身认知部分，表示本次速度值受自身历史最好位置的吸引力；第三项为群体认知部分，表示本次速度值受全局最优位置的吸引力，是粒子间协同合作关系的体现。c_1 与 c_2 表示学习因子。ω 为惯性因子，其值影响寻优解的范围，其值较大，则对全局的寻优效果强，而对局部的寻优效果弱，因此可采用动态 ω 获得全局和局部的寻优结果的平衡。

粒子的位置更新公式：

$$x_{i+1} = v_{i+1} + x_i \tag{7}$$

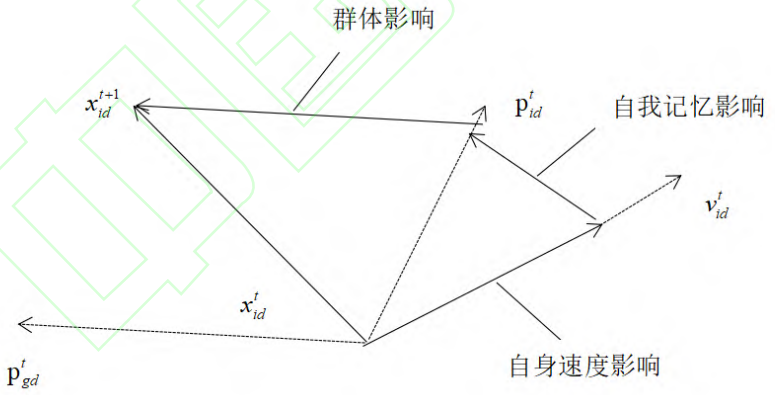

图 2 粒子位置更新示意图

具体实现过程为，首先根据本次迭代过程中保留的 Archive 集选择出粒子群的全局最优值 $gbest_i$，同时根据上次迭代保留的粒子的个体最优值 $pbest_i$，以及上诉公式进行更新粒子的速度值，然后再根据更新后的速度值来更新本次迭代的粒子的位置值x_i，即对应粒子的 Eps 参数值。

（3）根据支配关系，更新本次迭代后的粒子的个体最优值和 Archive 集。更新粒子的个体最优解方式为，依据 Pareto 支配原则对解进行支配关系比较，将粒子的当前解与历史个体最优解进行比

较，选择两者中的较优解作为新的个体最优解。同时依据 Pareto 支配原则更新 Archive 集，将独立于 Archive 集中的解加入到 Archive 集中，以便于下一次迭代过程进行全局最优值的选择，同时注意 Archive 集的容纳范围是有限的，若 Archive 集中的解超过设定的阈值，则对 Archive 集进行截断处理，即将 Archive 集进行目标空间划分后，将非劣解多于 1 个的网格空间中的非劣解进行随机删除，直至满足规定的阈值。

表 2 改进的 DBSCAN 算法伪代码

输入：

数据集 dataSet D

模型参数 MinPts，Eps

```
1: Initialize all points in D are unvisited   #标记所有对象为未访问
2: use MOPSO to obtain Eps values of each point   #使用 MOPSO 算法为各个点求解 Eps 值
3: for each p in D    #遍历数据集中的每个对象 p
4:     if p is unvisited   #p 是否已经被访问
5:         Mark p as visited    #标记为访问
6:         if p is core point    #p 是否为核心点
7:             Set N as Eps-neighbor of p    #将 p 的 Eps 内的数据集设置为 N
8:             Create a new Cluster C and put p in C      #建立新簇
9:             for each unvisited q in N       #遍历所有未访问的对象
10:                Mark q as visited
11:                put qin C        #将 p 加入 C 中
12:                Set M as Eps-neighbor of q
13:                if size of M ≥ MinPts       #判断 M 包含的数据个数是否大于 MinPts
14:                    put those points in N
15:                end if
16:            end for
17:        end if
18:    end if
19: end for
```

20: Mark all unvisited points as Outlier　　#标记所有为访问的点为异常点

21: End

输出：异常点集

本文改进的算法是基于 MOPSO 和传统 DBSCAN 的结合，设本文 MOPSO 算法中，粒子数目为 n，迭代次数为 m，每次迭代中，各粒子求解的时间为 t，由于目标函数 1 和目标函数 2 的求解，涉及到粒子 Eps 邻域的使用，故 t 的上限为 n，即该粒子的 Eps 邻域包含了所有粒子。因此本文 MOPSO 算法的时间复杂度可记为：$O(n*m*t)$。DBSCAN 算法的时间复杂度与数据规模、求解 Eps 邻域的时间相关，设数据规模为 n，求解 Eps 邻域的时间为 T，则时间复杂度为$O(n*T)$。故本文的时间复杂度为：$O(n*m*t)+O(n*T)$，其上限为$O(n*m*n)+O(n*n)$，即为$O(m*n^2)$。

4 仿真与结果分析

4.1 参数与环境

本文所有实验均是在 MatlabR2019a 版本上实验的，实验环境为 64 位的 Windows10 系统。为了进一步验证本文算法的有效性和先进性，在电芯智能制造数据集上进行了实验，并与其他算法进行对比。

本文通过与自适应确定 DBSCAN 参数算法、传统 DBSCAN 算法进行实验结果对比验证模型的有效性。其中，参数设置如表 3 所示：

表 3 参数设置

参数	数值
初始粒子数目	N（数据集的大小）
惯性因子ω	0.5
学习因子c_1	1
学习因子c_2	2
迭代次数	100
MinPts	8（输入数据的维度+1）

4.2 某电芯工厂制造数据

4.2.1 数据集说明

本文采用电芯智能制造数据来验证本文算法的实际效果。电芯智能制造数据集来自江西某工厂的电芯制造数据，时间范围为 2020 年 3 月至 2020 年 8 月。该工厂具有电芯极片制造，芯片卷饶、电芯组装等电芯制造的完整生产线。

该生产线的工艺流程如下：

首先，对正负极片材料进行处理，通过混料、涂布、辊压、极耳焊接等工序，完成正负极片的制作。正负极片制作结束后，卷绕机通过卷绕的方式将正、负极极片和隔膜组装制造形成基本的电芯，然后热压机对电芯进行热压整形，改善电芯的平整度，消除隔膜褶皱，使隔膜和正负极极片紧密贴在一起，降低电池内阻。电芯配对机将满足要求的电芯进行配对，再通过超声波焊接机将其进行焊接，实现电芯的串并联，组装成一定电压等级的电池组。接下来的软连接、包 maylar 和入壳工序将极芯装入已经冲好坑的铝塑膜，并完成顶封、侧封等工序，形成未注液的软包电池，然后通过注液和预焊工序，使电芯内部与外部环境隔离，最终焊接顶盖、进行氦检，测试电芯壳体泄露情况，测验通过后，即完成电芯的制造。该工厂电芯制造工艺流程图如图 3 所示：

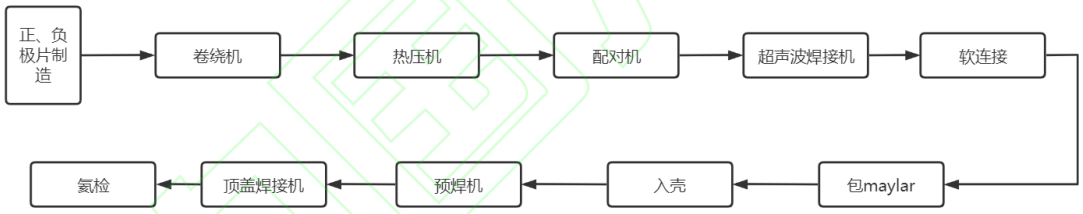

图 3 电芯制造工艺流程图

本次实验数据来源于该工厂卷绕工序中的卷绕数据。对于该工厂而言，影响芯片卷绕的因素主要可以分为 7 项，分别是负极片长度、绝缘阻抗、下隔膜长度、对齐度 2、正极片长度、对齐度 1、对齐度 3。本文使用该工厂的两台卷绕机设备的数据，对本文算法进行验证。部分数据如表 4：

表 4 卷绕工序数据

负极片长度	绝缘阻抗测试	下隔膜长度	对齐度 2	正极片长度	对齐度 1	对齐度 3
8344.550	43.800	8850.640	3.1390	8074.315	0.424	1.755
8347.714	1220.00	8858.268	2.9220	8075.840	0.822	1.729

8344.965	39.100	8852.46	3.1400	8074.615	0.514	1.755
8345.936	1650.000	8857.128	2.8680	8075.031	0.947	1.756
8346.235	39.100	8853.056	3.1660	8074.708	0.448	1.728
8345.890	1700.000	8856.766	2.8670	8077.133	0.770	1.729
8344.203	1740.000	8856.214	3.2210	8075.192	0.396	1.756
8345.196	1170.000	8855.955	2.8100	8076.001	0.874	1.812
8344.342	45.400	8856.887	3.2200	8074.153	0.455	1.783
8346.027	1030.000	8855.247	2.9770	8076.348	0.728	1.812
8343.856	43.600	8856.541	3.1390	8075.239	0.396	1.783
8345.590	1550.000	8861.460	2.8370	8075.909	0.867	1.840
8345.012	1700.000	8857.629	2.9740	8074.869	0.718	1.811
8345.844	1750.000	8861.270	2.9220	8076.278	0.881	1.785
8345.035	1740.000	8858.802	3.0290	8075.447	0.773	1.840
8345.658	1740.000	8862.529	2.7280	8075.262	1.002	1.840
8344.873	1720.000	8858.198	3.0550	8075.770	0.794	1.840
8345.704	1080.000	8864.566	2.7820	8075.701	0.943	1.784
8343.926	1770.000	8859.182	2.9710	8075.078	0.641	1.840

开始实验前，首先对数据进行 min-max 标准化处理，以负极片长度为例，设 n 个数据的数据片长度为 $x_1, x_2, \cdots, x_n$，其中最小值为 minx，最大值为 maxx，则标准化的结果为：

$$y_i = (x_i - \min x) / (\max x - \min x) \tag{8}$$

在每月的数据中，各随机选择若干条条数据，共组成 10000 条数据，利用传统 DBSCAN 算法进行聚类分析，使用多个 Eps 参数进行实验。具体选择策略如下：

首先，计算任意两数据间的欧式距离，得到距离矩阵，则矩阵的第 k 列数值表示各点邻域包含 k 个值的 Eps 值。然后将 k 列的数值取出，进行升序排序，设最小值为 disMin，最大值为 disMax，则 $Eps = disMin + rand() * (disMax - disMin)$，k 取值为 MinPts 的值。最后，通过 5 次取随机值，得到 5 个 Eps 值，然后与 MinPts 取值为 8 进行结合，使用 DBSCAN 算法进行聚类，结果如下表 5 所示。

表 5 聚类结果

序号	聚类个数	最大类与最小类的差值
1	7	568
2	6	753
3	8	459
4	7	681
5	8	735

可以看出，电芯数据集内部数据分布不均匀，随着 Eps 参数的取值不同，聚类个数不同，且最大类与最小类簇的数据个数差异较大，因此使用全局统一的 Eps 参数值的传统 DBSCAN 算法，不能有效的对其进行异常点检测。

4.2.2 实验评价指标

本文采用异常点准确率作为实验结果评价指标，即将检测出的异常点与总的异常点的比值作为评价指标，准确率为：

$$accuracy = N_1 / N \tag{9}$$

公式（9）中，N_1为算法检测出的异常点数目，N 为总的异常点数目。

4.2.3 实验结果分析

对数据之间的相关性分析，得出正极片长度和负极片长度的相关性最大，因此构成两个数据集，数据集 1 由正极片长度和其余 5 个数据维度构成，数据集 2 有负极片长度的另外 5 个数据维度构成，通过对数据集 1 和数据集 2 进行异常点检测，并将两数据集的非公共异常点定义为需要剔除的异常点数据。

同时，分别使用本文算法和其他算法对该工厂的卷绕数据集进行异常检测，每次实验取 10 次实验结果的平均值作为实验结果，为更直观的观察实验结果，如下图 4 所示：

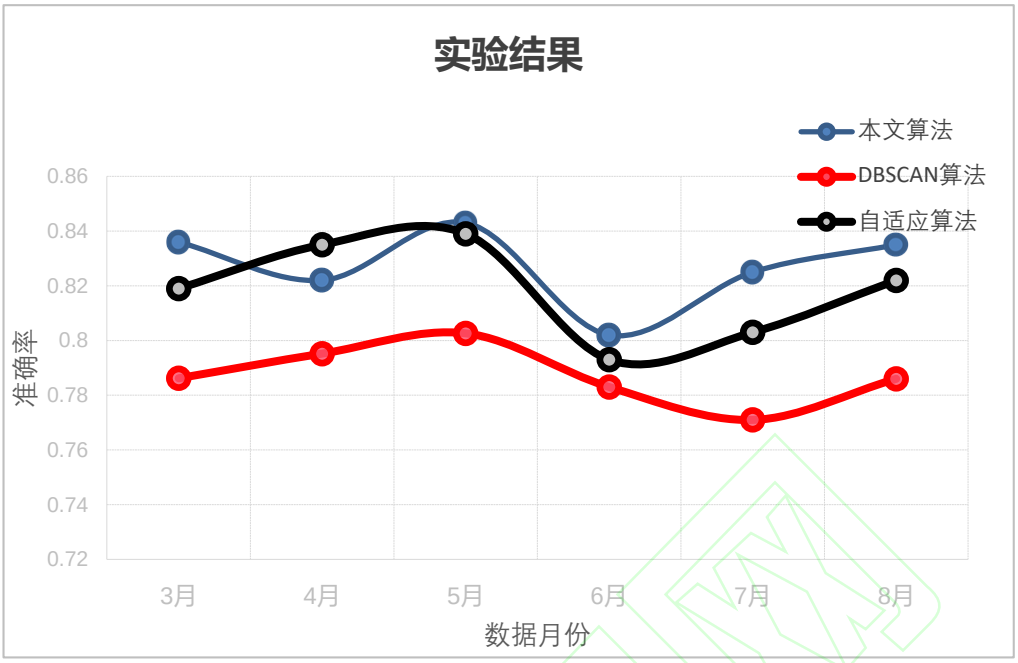

图 4 实验结果

根据异常数据检测实验的结果可以得出：本文及自适应算法的准确率都高于传统 DBSCAN 算法的准确率，除了 4 月份的数据，本文算法的准确率高于自适应算法。同时，自适应算法需要使用多个 Eps 和 MinPts 参数进行组合，多次进行 DBSCAN 聚类，以得到最终的参数值，而本文通 MOPSO 算法对数据集中的数据点都分别计算一个 Eps 值，然后进行 DBSCAN 聚类，只进行了一次聚类处理，对数据规模较大的数据集来说，本文的算法时间复杂度要比自适应算法的低。因此，本文算法对不同数据密度的数据点独立的设置参数值，解决了全局 Eps 值覆盖掉一些异常点的信息，提高了异常数据检测的准确率。

5 结束语

DBSCAN 算法可以有效的检测异常点，但是该算法对参数的选取敏感，取值不当会导致无法有效检测异常点，而且由于参数的全局性，该算法在密度分布不均匀的数据集上的检测效果较差。本文首先使用 DBSCAN 异常点检测算法筛选出异常点子集，然后针对 DBSCAN 异常点检测算法中全局参数 Eps 的不确定问题，使用多目标粒子群优化算法 MOPSO 对数据集中每个数据点求解一个 Eps 值，然后再调用 DBSCAN 算法筛选出真正的异常数据，最后通过某电芯工厂的实际工业数据进行仿真实验，证明了本文提出的算法的有效性。

本文需要对数据集中的每个数据进行 Eps 的求解，算法的时间性能不高，后续将进一步提高算法的时间性能。同时，在未来的工作中，我们也要考虑 DBSCAN 的 MinPts 参数的选择问题，通过

构建模型确定 DBSCAN 算法中的参数值，以提高检测异常点的准确率。

参考文献：

[1]Arthur Zimek,Peter Filzmoser. There and back again: Outlier detection between statistical reasoning and data mining algorithms[J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,2018,8(6).

[2]SHAO Jingfeng, HE Xingshi, WANG Jinfu, et al. Identification method for abnormal factors of spinning quality based on massive data[J]. Computer Integrated Manufacturing Systems, 2015, 21(10): 2644-2652.

[3] PATIDAR P，TIWARI A.Handling missing value in decision tree algorithm[J]. International Journal of Computer Applications, 2013,70(13):31-36.

[4]郭超,陆新建.工业过程数据中缺失值处理方法的研究[J].计算机工程与设计,2010,31(6):1351-1354.

[5] BERTSIMAS D，PAWLOWSKI C，ZHUO Y D.From predictive methods to missing data imputation：an optimization approach[J]. The Journal of Machine Learning Research,2017,18(1):7133-7171.

[6]熊中敏,郭怀宇,吴月欣.缺失数据处理方法研究综述[J].计算机工程与应用,2021, 57(14):27-38.

[7]Kishore Vasanth,Ramachandran Varatharajan. An Adaptive Non-Linear filter based on Median of Minimum Distance for Salt and Pepper Noise Removal in Mammogram Images[J]. Current Signal Transduction Therapy,2017,12(2).

[8]Sridhar Ramaswamy,Rajeev Rastogi,Kyuseok Shim. Efficient algorithms for mining outliers from large data sets[J]. International Conference on Management of Data, 2000,29(2):427-438.

[9]Flip Korn,S.Muthukrishnan. Influence sets based on reverse nearest neighbor queries[J]. International Conference on Management of Data,2000,29(2):201-212.

[10]樊瑞宣,姜高霞,王文剑.一种个性化 k 近邻的离群点检测算法[J].小型微型计算机系统,2020, 41(4):752-757.

[11] Yogita, Durga Toshniwal. Variance and density-based anomaly identification and ranking for evolving data streams[J]. Int. J. of Computational Intelligence Studies,2014,3(2-3):251-274.

[12]Breunig M M, Kriegel H P, Ng R T, et al. LOF: Identifying density-based local outliers[C].ACM Sigmod Record. ACM, 2000, 29(2): 93-104.

[13]Jin W, Tung A K H, Han J, et al. Ranking outliers using symmetric neighborhood relationship[C]. Pacific-AsiaConference on Knowledge Discovery and Data Mining.Springer, Berlin, Heidelberg, 2006: 577-593.

[14]邹云峰,张昕,宋世渊,倪巍伟.基于局部密度的快速离群点检测算法[J].计算机应用,2017,37(10):2932-2937.

[15]胡冠锵,张宇山.基于局部常数拟合的异常值检测[J].统计与决策,2021,37(12):15-18.

[16] DheerajKumar,James C. Bezdek,SutharshanRajasegarar,MarimuthuPalaniswami,Christopher Leckie, Jeffrey Chan, Jayavardhana Gubbi. Adaptive Cluster Tendency Visualization and Anomaly Detection for Streaming Data[J]. ACM Transactions on Knowledge Discovery from Data,2017,11(2):1-40.

[17] Ester M, Kriegel H P, Xu X. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise[C]. International Conference on Knowledge Discovery & Data Mining. 1996, 96(34): 226-231.

[18]周董,刘鹏.VDBSCAN:变密度聚类算法[J].计算机工程与应用,2009,45(11):137-141.

[19] Jeong-Hun Kim,Jong-Hyeok Choi,Kwan-HeeYoo,Aziz Nasridinov. AA-DBSCAN: an approximate adaptive DBSCAN for finding clusters with varying densities[J]. The Journal of Supercomputing, 2019, 75(1):142-169.

[20]Bryant A C, Cios K J . RNN-DBSCAN: A Density-Based Clustering Algorithm Using Reverse Nearest NeighborDensity Estimates[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(6): 1109-1121.

[21]冯宪凯,黄树成.基于 DBSCAN 的缺失值填充算法研究[J].计算机与数字工程,2020, 48(7): 1572-1575+1686.

[22]徐红艳,普蓉,黄法欣,王嵘冰.基于网格和密度比的 DBSCAN 聚类算法研究[J].计算机与数字工程,2020, 48(6): 1269-1274+1285.

[23]李文杰,闫世强,蒋莹,张松芝,王成良.自适应确定 DBSCAN 算法参数的算法研究[J].计算机工程与应用, 2019, 55(5):1-7+148.

[24]Shao, Mengliang, Qi, Deyu, and Xue, Huili. Big Data Outlier Detection Model Based on Improved Density Peak Algorithm[J]. Journal of Intelligent & Fuzzy Systems, 2021,40(4) : 6185 – 6194.

[25]X. Wang, J. Li, M. Bai and Q. Ma. RODA: A Fast Outlier Detection Algorithm Supporting Multi-Queries[J]. IEEE Access, 2021, 9:43271-43284.

[26]Vinue G., Epifanio, I. Robust archetypoids for anomaly detection in big functional data[J]. Advances in Data Analysis and Classification, 2021, 15: 437–462.

[27]Mujeeb Ur Rehman, Dost Muhammad Khan. Local Neighborhood-based Outlier Detection of High Dimensional Data using different Proximity Functions[J]. INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS, 2020,11(4):133-137.

[28] Thudumu S., Branch P., Jin, J., Singh, J. Adaptive Clustering for Outlier Identification in High-Dimensional Data[C]. ICA3PP 2019. Lecture Notes in Computer Science, Springer, 2022,1:215-228.

作者简介：

高 勃（1980-），男，山东泰安人，高级工程师，硕士，研究方向:信息物理系统与工业软件，E-mail：gaobo@bjtu.edu.cn；

柴学科（1996-），男，山西长治人，硕士，研究方向：信息物理系统与工业软件，E-mail:20125150@bjtu.edu.cn；

+朱明皓（1984-），男，安徽阜阳人，副教授，博士,硕士生导师，通讯作者，E-mail：mhzhu@bjtu.edu.cn。