并行处理网络下半结构化大数据快速聚类方法

王 珂 [1]
WANG Ke

摘 要 半结构化数据量已经超过 PB 级，在这种大规模数据集上快速响应交互式请求，对关系数据库查询和大数据处理技术都提出了挑战。然而当前的聚类算法均是离线批量处理结构化、非结构化数据。面对半结构化数据对象和应用需求的转变，需要对传统聚类算法针对性地优化和改进。设计一种并行处理网络下半结构化大数据快速聚类方法。首先，在常用的 Linux 与 Windows 网络环境中捕获大数据包，并对捕获的多源异构大数据做清洗和集成等预处理操作，完成数据准备工作；然后在并行处理网络下，基于 MapReduce 框架改进常规 CanpoyK-means 聚类算法形成 BCK-means 并行聚类算法，对多源异构大数据进行并行化聚类分析，实现半结构化大数据的快速聚类挖掘。实验结果表明，设计方法在 10 s 内即可完成不同类别的半结构化数据集的聚类且聚类结果稳定，在半结构化数据聚类挖掘效率方面具有优势。

关键词 并行处理网络；半结构化大数据；数据快速聚类；聚类方法

doi：10.3969/j.issn.1672-9528.2024.01.027

0 引言

随着信息技术与计算机软件的快速普及，无处不在的互联网应用每日产生数据量呈指数级增长，这些数据具有规模大、种类多且复杂可变等特点，也称之为大数据，如何从互联网大数据中提取出有用且有价值的信息变得越来越重要，而数据挖掘技术的发展为解决这个难题带来了机会。数据挖掘也就是从规模巨大且信息冗余的随机大数据中，提取到隐藏的人们事先不知道的潜在有用信息的过程，从相对学术的角度来说，数据挖掘横跨了分类、聚类、关联等众多学科，而这些学科知识的交叉，也推动了数据挖掘技术的可持续发展。在如今这个大数据时代，全球互联网每日产生与传输的数据规模不断增大，维度也逐渐复杂，导致传统的数据挖掘技术陷入到“数据繁琐但知识贫乏”的尴尬境地，难以满足当下互联网技术的发展需要。

作为数据挖掘技术中十分重要的一个分支的聚类分析方法，受到了国内外众多学者的广泛关注。王雪蓉等人 [1] 基于开展物联网事件的云模式通用解析模型，设计一种大数据聚类方法，解决了传统聚类方法考虑不足、聚类质量不高等问题。胡晓东等人 [2] 利用基于分组的引力搜索算法进行大数据聚类，与传统方法相比数据聚类效率更高。申锐等人 [3] 将常规数据聚类转换为图划分问题，再引入抽样改进加权核 K-means 算法，实现了大数据的快速聚类。

时至今日，互联网环境中各式各样的数据格式层出不穷，其中介于结构化与非结构化之间的半结构化数据，常在人们的生产、生活等实际问题中出现，所以本文参考上述文献研究一种针对半结构化大数据的快速聚类算法，弥补当前研究的不足。

1 网络环境中大数据包的捕获方法设计

网络环境中大数据包的捕获是半结构化大数据聚类分析的基础，而且大数据包的捕获效率将直接影响后续数据聚类的速度。随着计算机技术的迅猛发展，网络环境中的大数据规模越来越大，且结构越来越复杂。因此为实现半结构化大数据的快速聚类，本章将采用高效的方法来捕获网络环境中的大数据包 [4]。大数据包的捕获就是将网络环境中大数据报文信息完整收集起来，从而进行分析处理，由于实际网络环境中大数据包的源地址与目的地址均为未知，甚至数据的结构类型也是不可知的，所以在进行半结构化大数据快速聚类挖掘时，必须准确捕获到各种来源与各种结构类型的数据报文 [5]。当下，我国常用网络系统包括 Linux 与 Windows，所以本次半结构化大数据快速聚类研究主要从这两个网络环境中捕获大数据包 [6]，其中 Linux 网络系统属于一种四层的概念模型，也就是由应用、

1. 广州华南商贸职业学院云智信息技术学院 广东广州 510550
[基金项目] 广州华南商贸职业学院 2020 年大学生校外实践教学基地项目“粤嵌通信 - 计算机应用技术专业大学生校外实践教学基地”（编号 2020HMZLGC29）的研究成果之一

传输、互联及网络接口这四个层次构成，基于 Linux 网络环境的结构层次特点。本文采用开源软件 snort 底层的 Libpcap 数据包捕获模块进行 Linux 网络环境中的大数据包的捕获，其结构如图 1 所示。

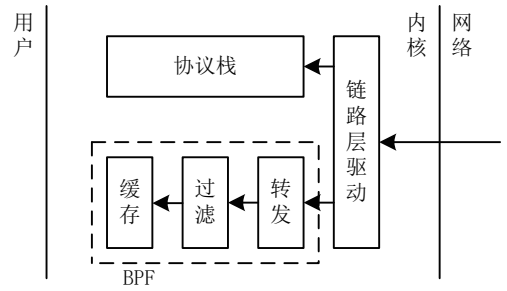

图 1 Libpcap 数据包捕获模块结构图

分析图 1 可知，Linux 网络环境中大数据包的传输路径是从网卡、驱动器、链路层、传输层，直到用户层，利用 Libpcap 捕获大数据包时，应在 Linux 网络系统的数据链路层上增添一个旁路监听器，只要 Linux 网络环境中存在数据传输，Libpcap 就会通过 DMA 操作将数据帧接收到自身程序中，再通过 BFP 过滤器按照预先设定好的规则对接收数据帧进行逐帧过滤。如果满足设置规则将数据包存储到缓冲区，如果不满足设置规则将数据包直接丢弃，从而完成了 Linux 网络环境中大数据包的捕获。

Libpcap 在捕获 Linux 网络环境中的大数据包时，定义了底层的网络监听框架，但无法直接用于 Windows 网络环境中，所以本文在捕获 Windows 网络环境中的大数据包时，引入了 Winpcap 程序，将 Libpcap 模块移植到 Windows 网络环境下，从而进行大数据包的捕获。Winpcap 是一个由核心过滤驱动程序 NPF 和动态链接库 Packet.dall、Wpcap.dll 所组成的体系结构，实际应用中，Winpcap 可以将从 Windows 链路层捕获数据帧的驱动程序写入自身内核中，再由专用驱动组件进行网络大数据包的捕获与缓存 [7]。

2 半结构化大数据的清洗和集成

在实际网络环境中捕获半结构化大数据包时存在很多问题，如数据属性值缺失、信息冗余、形式不匹配等，在半结构化大数据快速聚类挖掘之前不解决这些数据质量问题 [8]，势必会影响最终的聚类结果，所以本文主要从数据清洗和数据集成这两个方面，对原始捕获数据做对应处理 [9]。

首先是数据清洗环节，本文主要从缺失属性值填充与平滑去噪这两个方面入手，其中缺失属性值填充本文主要采用了拉格朗日插值法，简单来说就是根据同一个数据包内数据点的连续性变化规律，对缺失数据点进行插值补全。根据拉格朗日插值法的数学概念可知，已知在某平面上存在 m 个坐标点，那么就可以在该平面上找到 m-1 次多项式，表达式为：

$$\begin{cases} Y_1=\gamma_0+\gamma_1X_1+\gamma_2X_1^2+\cdots++\gamma_{m-1}X_1^{m-1} \\ Y_2=\gamma_0+\gamma_1X_2+\gamma_2X_2^2+\cdots++\gamma_{m-1}X_2^{m-1} \\ \cdots \\ Y_m=\gamma_0+\gamma_1X_m+\gamma_2X_m^2+\cdots++\gamma_{m-1}X_m^{m-1} \end{cases} \tag{1}$$

式中：$(X_1, Y_1), (X_2, Y_2), \cdots, (X_m, Y_m)$ 表示平面上 m 个点的坐标；$\gamma_0, \gamma_1, \gamma_2, \cdots, \gamma_{m-1}$ 表示通过 m 个坐标点的多项式系数。利用式（1）所示多项式，即可完成半结构化数据缺失值的插值填充，表达式为：

$$\begin{aligned} X^{'}=&Y_1\frac{(X-X_2)(X-X_3)\cdots(X-X_m)}{(X_1-X_2)(X_1-X_3)\cdots(X_1-X_m)} \\ &+Y_2\frac{(X-X_1)(X-X_3)\cdots(X-X_m)}{(X_2-X_1)(X_2-X_3)\cdots(X_2-X_m)} \\ &+\cdots+Y_m\frac{(X-X_1)(X-X_2)\cdots(X-X_{m-1})}{(X_m-X_1)(X_m-X_2)\cdots(X_m-X_{m-1})} \end{aligned} \tag{2}$$

式中：X' 表示插值多项式所求数据缺失值的近似值；X 表示半结构化数据缺失值对应的数据点。

由于一般在 Linux 与 Windows 环境中捕获的大数据包中存在很多噪声数据，严重影响了数据的质量，所以本文在进行数据清洗时还需对数据做去噪处理，这里本文主要引入了小波阈值去噪算法，假设现有式（3）所示的异构数据。

$$X(t)=f(t)+\delta(t) \tag{3}$$

式中：$X(t)$ 表示初始网络半结构化数据；$f(t)$ 表示数据中真实有用的信息；$\delta(t)$ 表示数据中高斯白噪声，服从高斯分布。在实际小波阈值去噪中，根据式（4）确定一个阈值为：

$$\mu=\begin{cases} 0.03936+0.1829\left(\ln L/\ln 2\right), L>32 \\ 0, L\le 0 \end{cases} \tag{4}$$

式中：μ 表示小波去噪阈值；L 表示初始网络半结构化数据的长度。

对初始网络半结构化数据做离散小波变换，表达式为：

$$\lambda=\alpha+\beta \tag{5}$$

式中：λ 表示离散变换后的初始网络半结构化数据；α 表示数据中真实有用的信息 $f(t)$ 对应的小波系数；β 表示数据中高斯白噪声 $\delta(t)$ 对应的小波系数。

在半结构化数据的小波域内，α 数量较少但幅值较大，β 数量较多但幅值较小，根据这个特点就可以完成小波去噪。

如式（3）所示，本文根据极大极小阈值法确定了去噪效果最佳的小波阈值，再根据式（5）所求小波系数进行小波去噪，也就是将绝对值小于阈值 μ 的小波系数进行剔除，将绝对值大于阈值 μ 的小波系数保留下来，从而完成网络半结构化数据的小波阈值去噪 [10]。

完成网络半结构化数据的小波阈值去噪后进行数据集成

处理，将网络环境中捕获的半结构化数据进行合并，从而形成一个数据集，便于后续的半结构化数据聚类分析[11]，具体流程如下。

首先定义好数据转换的规则，由于每一个独立的数据源都是一个完整的体系，所以数据类型各不相同，可能是因为结构化数据，也有可能是非结构化数据，所以这就需要进行数据集成的时候，根据捕获大数据包的特点，定义一个数据源格式解析与转换的规则，对原始数据做一些转换工作。然后再将转换后的数据整理到一起，形成一个完整的数据集。最后本次半结构化数据预处理全部完成，经过上述预处理可以提升原始数据质量，避免后续半结构化大数据的聚类挖掘因数据质量出现问题。

3 并行处理网络下半结构化大数据快速聚类

根据文中上述内容可知，本文在 Linux 与 Windows 网络环境中捕获的大数据包，不仅来源不同，而且数据结构存在较大差异，本文所研究的就是从这些多元异构大数据中聚类挖掘到半结构化大数据[12]。半结构化数据就是介于结构化与非结构化之间的一种数据，主要包括 HTML 文档、XML 文档等，这种类型的数据具有自描述、层次结构等特点，所以本文根据不同结构类型数据的特点利用不同的准则，对原始多源异构数据集进行聚类，使得彼此之间相似的数据对象聚到同一个类中，从而实现半结构化数据的挖掘[13]。

当下，常用的聚类算法有划分聚类、层次聚类等，但是本文捕获的互联网大数据集规模极大，导致传统聚类算法无法高效运算，不仅影响了半结构化大数据的聚类效率，而且难以保障聚类结果的准确性，因此本文为解决大规模数据集在聚类挖掘过程中存在的问题，引入了并行处理网络，意在通过并行化运算提升传统半结构化大数据聚类的速度。MapReduce 因其可靠、容错等优势，在并行处理网络中的数据处理分析领域受到广泛关注，MapReduce 可以将并行处理网络的逻辑层和维护层相分离，从而快速解决种种复杂问题。所以本文引入 MapReduce 框架对传统 CanpoyK-means 聚类算法进行改进，生成一个全新的 BCK-means 并行聚类算法，用于半结构化大数据的聚类挖掘中[14]。

利用 BCK-means 算法进行半结构化大数据的聚类分析时，可以将整个聚类挖掘分为以下几个主要阶段：粗略估计、统计分布估计以及逼近估计。首先是半结构化大数据的粗略估计，简单来说就是在给定的多源异构大数据集中随机选取一个数据对象 x_i，分别求出数据集中其他数据对象和该对象之间的距离 $d(x_i, x_j)$。其中 x_j 为多源异构大数据集的数据对象，在求出所有数据对象之间的距离后，进行排序，即可得到数据对象 x_i 与其他对象之间的最大距离 $d_{\max}$ 和最小距离 $d_{\min}$。然后在此基础上，构建一个大小为 n 的整型数组 1，从 $d_{\min}$ 到 $d_{\max}$ 范围内进行均匀采样，并将采样数据作为数组元素，那么每一个元素对应的范围长度 D 的计算公式为：

$$D=\frac{d_{\max}-d_{\min}}{n} \tag{6}$$

根据式（6）所求范围长度，即可确定各元素对应的范围：

$$F_i=\left(d_{\max}+(i-1)*D, d_{\min}+i*D\right) \tag{7}$$

式中：F_i 表示第 i 个元素对应的范围。以此完成了并行处理网络下的半结构化数据粗略估计阶段。然后进入统计分布估计阶段，在数组 1 的基础上构建一个数组 2，并在数组 2 中找到一个满足下式条件的最小元素 u_s：

$$u_s \geq \frac{k(n-1)}{n} \tag{8}$$

式中：k 表示 K-means 聚类算法的分类个数。确定了最小元素的序号 s，即可得到最佳聚类阈值，表达式为：

$$\varsigma=d_{\min}+(n-s)\times D \tag{9}$$

式中：ς 表示 CanpoyK-means 聚类算法的最佳聚类阈值。最后根据式（9）所求阈值进行逼近估计[15]，在该阶段中，首先确定 K-means 聚类的聚类中心，再利用阈值 ς 进行半结构化数据的分类，也就是在原始多源异构数据集中选取一个数据对象，将所有与该数据对象之间距离在阈值 ς 之下的数据对象划入其 canopy 子集内，不断重复这个聚类划分过程，直至半结构化数据对象的 canopy 子集内元素数量超过 K-means 聚类中心个数 k。然后对目前所得 canopy 子集进行降序排序，并将子集内前 k 个元素当成最终的聚类中心，此时所得半结构化大数据的聚类结果就是最终结果，进行输出即可。

综上，本文在 MapReduce 框架下，按照粗略估计、统计分布估计以及逼近估计这三个阶段，完成多源异构大数据集的 CanpoyK-means 聚类分析，进而实现了并行处理网络下半结构化大数据的快速聚类挖掘。

4 实验对比与结果分析

4.1 实验环境

为充分验证本文所设计的并行处理网络下结构化大数据快速聚类方法的聚类效果，本章将通过 Hadoop 实验平台展开半结构化数据的聚类实验。首先进行实验环境的搭建，实验中采用 7 台配置一致的计算机组成 Spark 集群，其中 1 台计算机作为主控制节点，剩下 6 台计算机作为计算节点，

各节点部署在实验室内的局域网中，相关软硬件配置如表 1 所示。

表 1 实验环境软硬件配置参数

环境	参数	配置
硬件	CPU	Intel Core i7 12700H
	硬盘	1T
	内存	16 GB
	网络环境	200 M 局域网
软件	操作系统	Ubuntul16.04
	Hadoop 版本	2.7.7
	Spark 版本	2.4.5

已知本次仿真实验采用了 1 个主控节点与 6 个计算节点部署了 Spark 集群，但实验资源有限，本次实验将主控节点同样当作计算节点进行使用，具体的节点规划信息如表 2 所示。

表 2 Spark 集群节点配置

节点名称	IP 地址	节点角色
主控制节点	192.168.1.104	Master、Worker、NameNode、DataNode
计算节点	192.168.1.105	Worker、DataNode
计算节点	192.168.1.106	Worker、DataNode
计算节点	192.168.1.107	Worker、DataNode
计算节点	192.168.1.108	Worker、DataNode
计算节点	192.168.1.109	Worker、DataNode
计算节点	192.168.1.110	Worker、DataNode

本次仿真实验中，将上述 Spark 集群节点都集成到 Hadoop 平台上，再通过 Java 语言实现进行软件安装与测试，在实验环境检测无误后，即可展开本次半结构化数据的聚类实验。

4.2 实验指标

在上述实验环境的基础上，以本文设计方法为实验组，并以文献 [1]、文献 [2] 中方法为对照组，然后分两个阶段进行本次仿真实验。第一阶段，基于某已公开发表的半结构化数据集为实验数据，分别采用实验组与对照组方法进行实验数据的聚类挖掘，再对比分析不同方法下的数据聚类结果。第二阶段，基于人工生成的半结构化数据集为实验数据，同样采用实验组和对照组方法对数据做聚类挖掘，对比分析不同方法的聚类结果。本次实验采用了不同数据集进行聚类分析，主要目的在于验证本文设计方法在半结构化大数据聚类中的适用程度，避免实验数据的偶然性影响实验结果的准确性，表 3 为本次实验数据集的具体分布情况。

表 3 实验半结构化数据集分布

数据集	样本数	类别数	类型
某公开数据集	300	4	日志文件、XML、JSON、email
人工数据集	180	3	XML、JSON、HTML

与此同时，本次仿真实验以聚类挖掘效率为实验指标，也就是在不同方法下半结构化数据的整个挖掘过程中，每隔一段时间检查一次数据的聚类效果，从而根据数据的聚类质量来衡量实验组方法和对照组方法的聚类挖掘效率。

4.3 实验结果分析

综合考虑半结构化大数据聚类挖掘的实际情况，本次仿真实验中两种不同类型的半结构化数据集聚类挖掘时间均控制为 30 s，并在实验过程中每隔 10 s 检查一次数据聚类效果。实验结束后，统计并整理实验组方法和对照组方法下半结构化数据聚类挖掘结果，如图 2、图 3 所示。

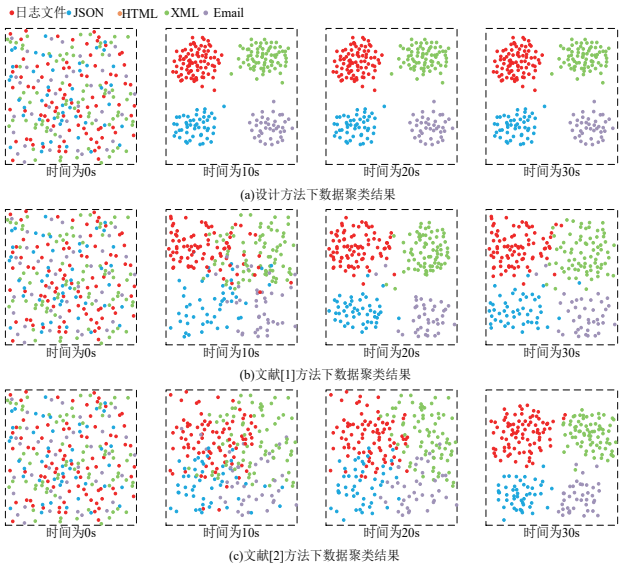

图 2 某公开数据集下的聚类结果对比

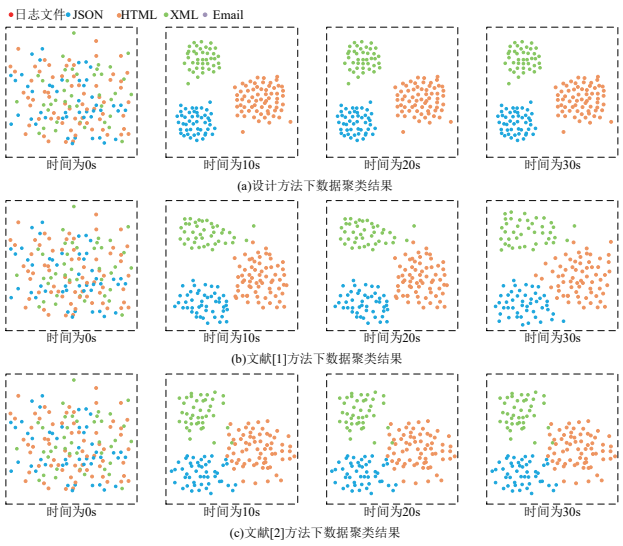

图 3 人工数据集下的聚类结果对比

从图2、图3中可以看出，在人工的半结构化数据集下，无论是本文设计方法还是文献[1]、文献[2]中方法，均可以在10 s内实现数据的聚类。但与对照组方法相比，本文设计方法下的半结构化数据聚类结果更好，且不会随着时间的推移出现任何变化，也就是本文设计方法下半结构化数据聚类结果的稳定性良好。在某公开的半结构化数据集下，随着待聚类局部簇的增加，对照组中的文献[1]与文献[2]方法，不仅无法在10 s内完成数据聚类，而且文献[1]中方法难以保证聚类结果的稳定性，但是本文设计方法仍可以在10 s内完成数据的聚类，且聚类质量一直较为优越，聚类结果仍比较稳定。这主要是因为本文引入了并行处理网络下的MapReduce框架，对半结构化数据进行了并行聚类，所以设计方法的聚类挖掘速度不会受待聚类数据集的规模所影响。由此可以说明，本文设计的并行处理网络下半结构化大数据快速聚类方法是合理且正确的，可以在提升半结构化大数据聚类挖掘效率的基础上，保障聚类结果的稳定性。

5 结语

伴随着大数据时代的来临，互联网环境中半结构化大数据越来越重要，但是计算机等技术的迅速发展使得大数据规模不断扩大、结构逐渐复杂，传统数据聚类分析方法已经难以满足大数据时代对结构化大数据的挖掘需求。所以本文提出一种并行处理网络下半结构化大数据快速聚类方法。首先捕获了人们常用的Linux与Windows网络环境中的大数据包，由于捕获数据属于多源异构数据且质量较差，所以还需对原始大数据做一系列预处理，以此提升大数据质量；然后在并行处理网络下对准备好的多源异构数据进行聚类分析。本文利用MapReduce框架改进了常规K-means聚类算法，形成一种并行化聚类算法，从而完成半结构化大数据的快速聚类挖掘；最后，通过仿真对比实验结果验证了本文设计方法下半结构化大数据具有良好的聚类挖掘效率，且数据挖掘聚类结果十分稳定，可以为实际的大数据分析提供理论参考。

参考文献：

[1] 王雪蓉，万年红．云模式事件混沌关联特征提取的物联网大数据聚类算法[J]. 计算机应用研究，2021,38(2): 391-397.

[2] 胡晓东，高嘉伟．基于分组模型的引力搜索智能大数据聚类方法[J]. 计算机工程与设计，2021,42(6):1660-1667.

[3] 申锐，吴睿．抽样改进加权核大数据谱聚类算法[J]. 机械设计与制造，2021(1):171-174.

[4] 李巍，廖雪花，杨军．基于频繁子树模式的半结构化数据集聚类[J]. 计算机工程与设计，2022,43(10):2783-2789.

[5] 李明倩，王苗，刘芳．基于相似度计算的大数据访存踪迹聚类仿真[J]. 计算机仿真，2023,40(3):485-489.

[6] 李旻，何婷婷．基于随机数三角阵映射的高维大数据二分聚类初始中心高效鲁棒生成算法[J]. 电子与信息学报，2021, 43(4):948-955.

[7] 王延，周凯，沈守枫．基于熵权法的教务大数据的挖掘和聚类分析[J]. 浙江工业大学学报，2023,51(1):84-87.

[8] 齐文，朱曦源，宋杰．基于特征转移概率的网络日志聚类分析算法[J]. 小型微型计算机系统，2023,44(3):514-520.

[9] 张强，白征东，辛浩浩，等．基于共享单车时空大数据的细粒度聚类[J]. 测绘通报，2021(5):15-19+29.

[10] 赵莎莎，朱雅魁，王悦．基于大数据分析的综合能源系统负荷特性聚类分析[J]. 电测与仪表，2023,60(2):10-15+52.

[11] 白雨佳，李靖，高升．基于最优K均值聚类算法的负荷大数据任务均衡调度研究[J]. 电力电容器与无功补偿，2022, 43(6): 85-91.

[12] 郑冬花，叶丽珠，隋栋，等．云计算环境中面向大数据的改进密度峰值聚类算法[J]. 济南大学学报(自然科学版), 2022, 36(5):592-596+602.

[13] 李清．基于改进PSO-PFCM聚类算法的电力大数据异常检测方法[J]. 电力系统保护与控制，2021,49(18):161-166.

[14] 胡健，徐锴滨，毛伊敏．基于MapReduce和IFOA的并行密度聚类算法[J]. 计算机应用研究，2021,38(5):1336-1343.

[15] 陶涛，毛伊敏．基于MapReduce和改进人工蜂群算法的并行划分聚类算法[J]. 科学技术与工程，2021,21(21): 8989-8998.

【作者简介】

王珂（1988—），男，广东潮州人，硕士研究生，讲师，研究方向：软件工程。

（收稿日期：2023-09-26）