分类号：TP391　　学校代码：10363
密　级：公开　　学　　号：2220910118

安徽工程大学
Anhui Polytechnic University

硕士学位论文

题目　基于密度方法的数据流聚类算法研究

论文作者　张国一
校内导师　刘三民 教授
校外导师　成彬 高级工程师
专业学位类别　计算机技术
研究方向（领域）　数据分析与知识工程

论文提交日期：2025 年 5 月 9 日

分类号：TP391　　单位代码：10363
密　级：公开　　学　　号：2220910118

题　　目　基于密度方法的数据流聚类算法研究

英文并列

题　　目　Research on Data Stream Clustering Algorithm Based on Density Method

学生姓名：张国一

指导教师：刘三民 教授 成彬 高级工程师

专　　业：计算机技术

研究方向：数据分析与知识工程

论文答辩日期：2025 年 5 月 10 日

二、安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名：张凯一

日期：2025 年 5 月 9 日

三、安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 □，在　　年解密后适用本版权书。

本学位论文属于

不保密 ☑。

学位论文作者签名：　　指导教师签名：

日期：2025年 5月 9日　　日期：2025年5月9日

基于密度方法的数据流聚类算法研究

摘要

在信息技术迅速发展的时代，大量数据以流的形式呈现。如何从连续不断的流数据中挖掘和提取隐藏的知识与结构，已经成为当前研究热点之一。数据流聚类挖掘，是数据流挖掘领域的重要研究方向之一，受到广泛的关注。由于在数据流环境中数据形状不确定性、异常、噪声和概念漂移等情景的出现，给数据流聚类挖掘研究带来诸多挑战。因此，本文将基于密度聚类思想，研究动态聚类模型，解决数据流聚类挖掘中的相关问题。具体工作如下：

（1）现有数据流聚类算法易受数据流中噪声和数据形状不确定性等因素影响，导致数据流聚类性能不足。针对这类问题，论文从样本特征空间分布随时间推移发生显著变化的角度，提出一种基于密度峰值聚类的数据流动态聚类方法，旨在解决数据流中数据形状不确定性和噪声等问题，提高模型聚类性能。该聚类方法由在线和离线两个阶段构成。在线阶段能实时处理连续到达的数据，通过设计微簇的不均匀衰减策略，降低历史数据对模型聚类的影响，根据样本与微簇的距离动态地对样本进行加权。离线阶段在密度峰值聚类的基础上，设计一种基于最近邻域自适应的局部密度计算方式，以降低密度峰值算法在分配阶段的“多米诺效应”，即样本分配阶段的连带错误影响，提高模型在噪声环境下的聚类性能。

（2）现阶段数据流聚类挖掘模型，易受数据流中异常和概念漂移等因素影响，导致挖掘质量低下。为解决这类问题，论文从异常与正常样本分布差异显著的角度出发，提出一种基于密度峰值聚类与微簇结构的数据流异常检测方法。该方法在流聚类模型中设计异常检测机制，通过检测流中异常，提高聚类质量，并能适应概念漂移环境。模型包括微簇学习模块和异常检测模块，学习模块依据密度聚类原则，将数据映射至微簇。异常检测模块将微簇作为分析单元，通过改进的K近邻思想，优化局部密度定义，强化样本与邻近样本间的相关性；采用改进的密度峰值算法进行区域划分，提出一种全局异常评分计算方法，结合局部和整体计算数据的异常评分，按异常评分高低筛选出异常，降低异常对聚类性能的影响提高聚类质量。

本文旨在通过密度聚类方法，解决数据流环境中的相关难题。如数据形状的不确定性、异常、噪声、概念漂移和标签缺失等问题，提出相应的优化方案。并从多个方面构建完整流式数据聚类框架，在含噪声和异常的动态数据流环境下数据流聚类研究，具有重要的理论意义和实践价值。

关键词：动态数据流，数据流聚类，微簇结构，概念漂移，密度峰值，异常检测

Research on Data Stream Clustering Algorithm Based on Density Method

ABSTRACT

In the era of rapid development of information technology, large amounts of data are presented in the form of streams. How to mine and extract hidden knowledge and structures from continuous and endless stream data has become one of the current research hotspots. Data stream clustering mining is one of the important research directions in the field of data stream mining, and it has received widespread attention. Due to the uncertainty of data shape, anomalies, noise, and concept drift in the data stream environment, there are many challenges in data stream clustering mining research. Therefore, this paper will study dynamic clustering models based on density clustering ideas to solve the related problems in data stream clustering mining. The specific work is as follows:

(1) Existing data stream clustering algorithms are prone to being affected by factors such as noise and uncertainty in data shape, resulting in insufficient clustering performance. To address these issues, this paper proposes a dynamic data stream clustering method based on density peak clustering, aiming to solve problems such as data shape uncertainty and noise in data streams and improve clustering performance. This clustering

method consists of online and offline stages. The online stage can process continuously arriving data in real-time by designing an uneven decay strategy for micro-clusters, reducing the impact of historical data on model clustering, and dynamically weighting the samples according to their distance from the micro-clusters. The offline stage, based on density peak clustering, designs a locally adaptive density calculation method based on nearest neighbors to reduce the "domino effect" in the assignment phase of the density peak algorithm, i.e., the cumulative error impact during the sample assignment phase, thereby improving clustering performance in noisy environments.

(2) Current data stream clustering mining models are susceptible to being influenced by anomalies and concept drift in the data stream, leading to low mining quality. To address this issue, this paper proposes an anomaly detection method for data streams based on density peak clustering and micro-cluster structures, focusing on the significant distribution difference between anomalous and normal samples. The method designs an anomaly detection mechanism within the stream clustering model to improve clustering quality by detecting anomalies in the stream and adapting to concept drift environments. The model includes a micro-cluster learning module and an anomaly detection module. The learning module maps the data to micro-clusters based on density clustering principles. The anomaly detection module analyzes the

micro-clusters, optimizes the local density definition through an improved k-nearest neighbor approach, and strengthens the correlation between samples and nearby samples. An improved density peak algorithm is used for region division, and a global anomaly scoring method is proposed. By combining local and global anomaly scores, anomalies are detected by selecting those with higher anomaly scores, thus reducing the impact of anomalies on clustering performance and improving clustering quality.

This paper aims to solve related challenges in the data stream environment, such as uncertainty in data shape, anomalies, noise, concept drift, and label missing, through density clustering methods, and proposes corresponding optimization schemes. A complete stream data clustering framework is constructed from multiple aspects, and the research on data stream clustering in dynamic data stream environments with noise and anomalies holds significant theoretical and practical value.

Zhang Guo yi（Computer Application Technology）

Supervised by Prof. Liu Sanmin, Chen Bin

KEY WORDS: dynamic data streams, data stream clustering, micro-cluster structures, concept drift; density peaks, anomaly detection

目录

第 1 章　绪论

应用程序、网站、社交媒体、电子商务和金融交易等众多生活场景，时时刻刻都在产生海量数据[1]。这些数据按照时间顺序，按照快速连续到达的形式被读取或处理[2]，因此这种快速连续到达的数据被称为数据流[3,4,5]。数据流挖掘分为数据流聚类挖掘[6]和数据流分类挖掘[7]。数据流聚类挖掘旨在分析并提取数据流中有价值的信息，发现有价值的聚类结构。数据流聚类挖掘采用无监督聚类方法处理分析数据流，当前数据流聚类挖掘主要面临的问题可以总结成以下两个方面：

（1）动态数据流环境数据流量大、流速快、数据分布复杂，数据流聚类方法必须快速对样本识别处理。但数据流中数据形状的不确定性和噪声等因素，大幅影响数据流聚类模型在实际应用中的精准性和稳定性。

（2）数据流环境存在异常和概念漂移等诸多挑战，聚类模型的性能会受到这些因素影响。且动态数据流环境与静态数据集环境不同，在静态环境下一次聚类就可以获得全部样本信息，流环境是动态变化的，一次聚类不可能获得所有样本信息，聚类模型需要具备自动更新能力。

在真实场景中数据形状不确定性、异常、噪声、概念漂移和标签缺失等因素影响，导致聚类模型的性能下降。在这样的情景中构建稳定有效的聚类模型，成为本文研究的重点。

1.1 研究背景与意义

随着网络应用的快速发展，数据流的产生速度和规模不断增加，数据流中蕴含着大量潜在的、有价值的信息。例如，亚马逊每天处理用户上百万次购买和搜索、推特每天处理超过 5 亿条推文，这就意味着每秒处理 6000 条推文信息、纽约证券交易所在高峰时期每秒处理超过 25000 笔交易等等。这些数据中包含着宝贵的知识与模式，能够为智慧医疗服务[8]、金融诈骗检测[9]和交通检测[10]等领域提供有价值的支持。然而，数据流的信息提取比静态数据更为困难，它需要满足有限内存、单遍扫描和概念漂移检测的要求。因此，数据流挖掘模型必须能实时分析高速生成的数据。为实现高效率的数据流信息提取，促进数据的价值转换近年来对数据流挖掘的研究逐渐深入。

聚类方法是最适合的实时数据流挖掘的方法之一，它可以在标签稀缺场景下，实现对数据分组和统计。聚类可以将数据分为若干个簇，让同一个簇的数据样本尽可能相似，不同簇的数据对象尽可能相异[11,12,13]。传统的聚类方法，适用一次性到达且分布稳定的静态环境。数据流聚类与传统聚类方法，有显著的不同之处，数据流聚类适用场景是连续到达且分布不稳定的动态环境，一次性处理一部分对象。

数据流传输中，常常会发生概念漂移和概念演化等等，这些因素给数据流聚类带来新的挑战。将传统聚类算法的聚类能力扩展到数据流环境理论上可行，如经典的静态聚类算法密度峰值聚类算法（Density Peak Clustering 简写为 DPC）[14]，被改进用于数据流聚类并取得良好的效果。这证明基于密度的聚类方法，在数据流上有着非凡的聚类能力。

实际应用中基于密度的数据流聚类方法，不仅可以作为单独的数据分析方法，还可以为其他方法，提供数据预处理和决策支持。例如，在推荐系统中为电商平台的推荐商品发布策略，对用户行为数据流进行聚类，实时分析用户喜好和个性特点，形成不同的用户集群。针对不同的用户集群，选择相对应的推荐的内容，提高推荐效率。然而，随着应用数据的传输，推荐系统可能会受到恶意网络攻击或生成数据流的设备故障，导致数据流中出现异常。例如，用户的某些喜好可能与实际偏差很大，这会影响推荐系统的准确性，导致原本良好的推荐性能大幅下降。因此，为确保系统能够尽可能准确地推荐商品，需要优化用户兴趣聚类模型，具备识别异常的能力，进一步提高聚类性能。

综上所述，数据流聚类仍面临着诸多挑战，如数据形状不确定性、异常、噪声、概念漂移等，解决上述问题研究新的数据流聚类方法已成为必然。

1.2 国内外研究进展

数据流聚类挖掘属于无监督学习[15]，不需要有标签的训练数据[16]可以实现在短时间内，对快速到达的数据流进行归类[17]，与现实生活中数据缺少标签的情况相符[18]。面对不断到达且快速变化的数据流，尽管数据流聚类方法大都能满足单遍扫描、实时响应、概念漂移检测等约束，但这些聚类挖掘方法难以有效处理数据流中的异常，导致聚类性能受到影响。因此，许多学者开始研究如何将异常

检测方法，应用在数据流聚类挖掘领域。本节以数据流聚类、密度峰值聚类、数据流异常检测等三点为切入点，阐述当前的国内外研究进展和发展方向。

1.2.1 数据流聚类挖掘

近年来，国内外学者在数据流聚类方面做了大量研究[19]。数据流聚类是将数据流中的相似对象划分为一个或多个“簇”的过程[20,21]，划分后同一簇中元素彼此相似，不同簇中样本彼此相异。现有数据流聚类方法一般由传统聚类算法扩展而来，可根据所扩展的传统算法分为：基于划分的方法、基于层次的方法、基于密度的方法和基于网格的方法等等。

（1）基于划分的聚类方法

STREAM 方法[22]应用 K-median 算法采用分治思想，使用中位数来计算聚类中心，将聚类问题简化为数据点到最近簇的距离代价最小化问题，即找出代价最低的聚类的数量和位置。AdaptiveKMeans[23]方法基于 K-Means 算法，对数据流聚类，该方法解决需要预设聚类个数和概念漂移等问题。文献[24]提出一种基于 K-Means 的进化方法，可以自动估计聚类数量 K，在线时不需要存储数据概要，只需要维护最终聚类结果，使用 Page-Hinkley 测试跟踪聚类质量，如果质量下降，算法自行调整。基于划分的数据流聚类方法，易于实现且相对简单，虽可发现有意义的聚簇结构，但需预定义聚簇的个数，处理非球形数据时可能会效果不理想。

（2）基于层次的聚类方法

Clustream 方法[25]允许在不同的时间范围对数据聚类，它通过存储所有时间戳的线性和，以及平方和来扩展聚类特征（Clustering feature vector，CF）。为支持不同的时间范围，按照金字塔方案，定期存储当前微簇的快照，每个快照的更新周期不固定，通过存储的快照减去当前微簇，能近似地得到数据流中需要用到的部分。将提取的微簇运行 K-Means 算法来生成宏观簇。基于 Clustream 方法的框架，学者们提出许多改进方法。例如，HPStream[26]引入投影聚类，为每个微簇（子空间微簇）选择最佳属性集，获得了比 Clustream 更好的聚类质量。SWClustream 方法[27]通过滑动窗口创建时间聚簇特征，解决当微簇中心逐渐移动时，SWClustream 方法通过不断增长的半径维持微簇，未将微簇分裂成许多更小的微簇，导致聚类的性能下降，但该方法在运行时间和内存使用方面具有更好的

性能。E-Stream 方法[28]将聚类演化划分为五种类型：出现、消失、自演化、合并和分裂，并采用时间衰减模型和簇直方图来识别聚类演化的不同类型。REPSTREAM 方法[29]是一种基于图的数据流层次聚类方法。该算法通过更新两个稀疏图来识别聚类，这两个图是通过将每个顶点与其 K 近邻顶点连接形成的。第一个图捕捉到数据点之间的连接关系，用于选择一组具有代表性的顶点；第二个图由这些代表性顶点构成，有助于在更高层次上做出聚类决策。REPSTREAM 方法还应用时间衰减窗口来减少旧数据的影响，并跟踪代表性顶点之间的连接性，根据它们的连接性决定是否合并或拆分。基于层次的数据流聚类方法虽可发现有意义的聚簇结构，但其具有较高的计算代价，而且对流数据到达的顺序敏感。

（3）基于密度的聚类方法

Denstream 方法[30]是一种基于密度的数据流聚类方法，使用微簇捕获数据流概要信息，在线阶段每个微簇都有一个聚类特征向量，通过聚类特征向量可以得到微簇中心和微簇半径等信息。Denstream 方法的微簇分为 2 种类型：核心微簇和离群微簇，离线阶段在 2 种类型的微簇上应用 DBSCAN 算法，可以识别任意形状的聚类，但是过于依赖参数选择。NTOUTSI 等人提出一种基于密度的高维数据流投影聚类方法[31]，该方法提出一种特殊结构，即投影微簇用于在相关维度上总结一组对象。在流环境，微簇很可能会随着时间的改变成为离群数据，该方法提出不同类型的微簇，以便逐步形成真实的投影微簇，并安全删除离群数据。与 HPStream 方法不同，当一个新数据到达时，如果它远离现有的微簇，将被视为一个潜在微簇（即使它是一个离群点）。同时，一些旧的微簇将被删除，以保持微簇总数不变。ACSC 蚁群流聚类方法[32]提供一种对非平稳微簇进行总结的流聚类方法，解决微簇个数变化问题，使用采样方法解决数据流聚类的速度要求，点与聚簇的相似性评估，仅需要使用这个微簇中的一个点。该方法使用随机抽样取代传统的穷举法，搜索每个点合适的微簇，然后搜索每个微簇的最近邻，使用蚂蚁启发的排序方法对这些微簇细化。CEDAS 方法[33]可聚类演变概念漂移数据流和任意形状的簇。MuDi-Stream[34] 是一种在线－离线框架的数据流聚类方法。在线阶段，以核心微簇形式保存多密度数据流演化的汇总信息；离线阶段，使用自适应的基于密度的聚类算法生成最终的聚簇。文献[35]提出一种基于密度的两阶段数据流聚类方法适用于分布复杂的数据流环境，该方法主要特点是根据输入

的数据自动调整阈值。I-HASTREAM 方法[36]也是基于密度的两阶段数据流聚类方法，为 HASTREAM 方法的改进版本，在线阶段它通过创建微簇保存数据概要，离线阶段微簇得到最小生成树，最后采用层次聚类方法进行聚类。基于密度的数据流聚类方法，能够识别任意形状的聚类，但其缺点在于需要设定过多的预设参数。

（4）基于网格的聚类方法

D-Stream 方法[37]采用在线－离线框架处理数据流。在线阶段，将输入的数据点映射到网格；离线阶段，计算网格密度基于密度对网格进行聚类。与其他方法不同 D-Stream 自动、动态地调整聚簇，不需要用户额外设置聚类参数。这种方法能清晰表明聚类间的距离关系，提高学习数据流的效率，一定程度上降低噪声的影响。但忽略网格之间区域的数据密度，可能会将靠近但同时被低密度区域分开的网格合并。MR-Stream 方法[38] 通过使用树状数据结构，将数据细分为更多单元网格，促进在多种情景下发现聚簇。在线阶段，它将最新数据分配给适当的单元，并更新概要信息；离线阶段，在固定高度 h 获得树的一部分和 h 确定的分辨率级别上执行聚类。基于网格的数据流聚类方法，运行速度较快能发现任意形状数据，但聚类质量取决于选取的网格参数的定义。

综上所述，当前数据流聚类技术处于持续发展的状态，数据流聚类技术被应用到各个领域，如数据流异常检测[39]、数据流概念漂移检测[40-45]和数据流集成分类等等。然而，当前的数据流聚类技术仍存在一些缺陷，如易受数据形状不确定性和噪声等因素影响，导致降低聚类性能。针对薄弱项拓宽数据流聚类研究方向，研究新的数据流聚类方法，解决传统流式聚类方法在聚类时的“通病”，提高模型在流式环境下聚类的性能。在静态聚类算法中，密度峰值聚类算法通过决策图，能高效地发现不同密度的簇，为提高数据流聚类的效率，满足其他领域对数据流聚类技术的需求。本课题改进微簇结构，缓解数据流概念漂移的影响和提高模型聚类效率，设计基于密度峰值数据流聚类方法，提升模型数据流聚类性能。

1.2.2 密度峰值聚类

密度峰值聚类算法是 2014 年发表在 Science 上的一种静态聚类方法，它可以识别任意形状的簇，具有发现异常和识别噪声点的能力。将密度峰值聚类算法在静态数据集上的聚类能力，拓展到数据流领域，利用密度峰值聚类的特性快速删

除噪声和检测异常，对提高数据流聚类性能具有非凡意义。密度峰值聚类通过计算样本的局部密度和该样本与其他样本的相对距离，画出二者关系的决策图。通过决策图选择聚类中心，对其他点进行类簇分配。该算法简单易行，可识别任意形状的数据，快速识别噪声，聚类效果良好，但阈值参数和聚类中心需要主观判断，数据点分配未考虑样本邻域信息，易发生连锁的分配错误。

针对 DPC 算法的阈值参数和聚类中心难以选择的问题，文献[46]提出一种改进的 K 近邻密度峰值聚类算法，通过 K 近邻的思想来计算数据点的局部密度，并针对聚类中心难以选择的问题，该方法定义了自适应的阈值参数。文献[47]等人提出一种基于残差的密度峰值聚类方法，采用残差计算测量数据邻域的局部密度，可以发现重叠簇或者低密度的簇，强化密度峰值算法对不同数据集的通用性。Chen 等人提出一种基于 KNN 的快速密度峰值聚类方法[48]，使用 KNN-Density 获局部密度，KNN-Density 通过覆盖树方法快速计算局部密度，极大提高局部密度计算的效率。张等人提出基于代表点与 K 近邻的密度峰值聚类方法[49]，使用 K 近邻和代表点刻画样本的全局分布，利用 K 近邻信息提出一种加权 K 近邻分配策略，缓解 DPC 算法在分配阶段的“多米诺效应”影响。

综上所述，近几年学者们对密度峰值算法的改进和应用，包括采取 K 近邻等新方法来计算局部密度等。在目前看来，密度峰值算法仍有改进和应用的空间。比如方法对阈值参数的变动依旧敏感，DPC 的“多米诺效应”仍然存在这些问题亟待解决。因此，有必要寻找一种新的局部密度度量规则和聚类中心确定方法。

1.2.3 数据流异常检测方法

数据流聚类和异常检测密切相关，尤其是在数据生成速度快、样本数量庞大的流环境，数据往往会受到传输时的因素影响，产生一些异常。霍金曾提出过一个广泛的异常定义：“一个观察结果与其他观察结果偏离如此之大，以至于引起怀疑它是由不同的机制产生的[50]”。即异常产生是因为样本的产生机制发生变化，这往往隐藏着重要的关键信息。例如，医疗领域，异常可能代表医疗检测器械发生故障或病人病情突然恶化；工业领域，异常可能是设备过热或者故障[51]；金融交易领域，异常可能是金融欺诈[9]等。通过识别异常，可以提高数据流聚类模型的聚类质量，还能利用异常中的潜在信息，预防事故的发生减少损失。因此，数据流聚类与异常检测的紧密结合，不仅能提升聚类性能，还为各个领域提供风险

防范的可能。

为应对数据流中样本缺失标签和概念漂移情况。现在大部分异常检测方法，通过无监督方法检测异常值，常见的数据流异常检测方法可以分为三类：基于预测的方法、基于分布的方法和基于聚类的方法。基于聚类的方法尤其重要，因为它通过将数据流聚类，能够有效地识别与大多数样本偏离较大的异常样本，提高聚类质量并检测数据流中的潜在异常。本节依据三种异常检测类型，介绍当前异常检测的研究进展。

（1）基于预测的方法

该类方法通过比较实际值与预测值的差值，衡量数据异常程度[52-55]。如FuseAD 方法[53]结合 CNN 和 ARIMA 作为预测器，通过计算预测值与实际值之间的欧氏距离，计算异常评分。如果预测值超过阈值，则将该数据报告为异常，否则将真实值添加到预测器的数据序列中。该方法通过修改输入序列实现预测器的动态更新，与固定模型相比，它能在一定程度上适应数据的动态变化。基于预测的方法能够有效捕捉数据流中的非线性特征和时序模式，从而提高预测精度。但是，这类方法计算复杂度较高，且对阈值设置敏感，可能导致误报或漏报。

（2）基于分布的方法

一些基于离群数据分布的检测方法[56-58]，通过增量学习方法能够在线处理数据流，不需要设置额外参数。还有些方法采用统计技术，如 Costa 等人提出利用典型化和偏心率分析，检测工业数据中的异常[59]，它不需要额外的参数设置，这种方法在检测空间异常方面非常有效。然而，这类方法计算量大、延迟高、难以适用在数据流环境。

（3）基于聚类的方法

聚类方法不需要考虑数据标签信息，这些方法将距离其他簇较远且规模较小的簇或点视为异常，并以此作为判断异常的依据[60-62]。Jain 等人将 K-Means 和 SVM 结合起来检测网络流量异常[63]并取得不错的效果，但模型性能容易受到聚类个数 K 的影响。EEAC 提出一种基于证据积累的动态集成聚类方法，在心律失常检测中取得较好的效果[64]。基于聚类方法不依赖于数据标签信息，通过判断聚类的远离度和规模，能够有效识别异常数据。但这类方法的效果容易受到聚类个数 K 和数据特性的影响，可能导致检测性能不稳定。

上述方法主要侧重在数据挖掘过程中异常检测的准确性，忽略模型自身的挖掘性能。在业务应用中监控系统往往需要实时地处理数百万条数据，优秀的异常检测方法需要满足单遍扫描快，快速给出异常检测结果[65-67]。

一些学者对两阶段挖掘框架加以改进，提出一些在线的数据流异常检测方法。例如文献[68]提出了一种轻量级的在线异常检测方法 LODA，专为处理大规模流数据中的异常检测问题而设计，并具有较低的计算开销和较高的准确性，提高挖掘模型的效率。还有一些学者们，针对异常影响模型聚类性能的角度出发，提出基于聚类的异常检测方法[69-70]，这些方法通过检测异常提高一定的聚类性能。但是，这些方法受流环境中概念漂移影响，稳定性较差，检测精准性下降和聚类性能降低。这需要设计一种新的基于聚类的数据流异常检测方法，解决数据流聚类挖掘领域异常和概念漂移等因素影响聚类性能的问题。

使用 KD-Tree 结构的密度峰值聚类算法[71]，可以快速检测出数据集中的异常值、减少误判率提高聚类质量，但 KD-Tree 的存储结构难以适应，这种需要不断更新存储信息的环境。因此如何在数据流聚类挖掘中识别异常，提高模型的聚类性能，成为本研究的重点研究内容之一。

1.3 论文主要研究内容

本研究根据现阶段主流的数据流聚类方法，结合处理概念漂移的思想，构建新的数据流聚类模型，并通过数据流聚类模型解决上述数据流中的实际问题。例如，解决在动态的数据流环境中数据不确定性、异常、噪声和概念漂移等因素影响聚类性能问题。

（1）基于密度峰值的数据流动态聚类方法。

数据流聚类挖掘的目标是识别有价值的聚类结构和数据。但数据流具有不断生成、高速到达、数据分布动态变化等特性，数据流聚类方法需要适应这种特性。传统的数据流聚类方法，易受数据形状不确定性和噪声等因素影响聚类性能。因此，为解决数据形状不确定性和噪声等问题，识别出有价值的聚类结构，本研究提出一种基于密度峰值的数据流动态聚类方法，该方法能够识别任意形状数据且不易受噪声影响。该方法由在线和离线两部分组成，在线阶段通过微簇结构在线处理即时到达的样本，设计基于微簇的不均匀衰减策略，保留重要的历史样本；

离线阶段通过改进局部密度的定义方法，设计基于最近邻域的局部密度定义方法，降低密度峰值算法分配阶段“多米诺效应”的影响。通过继承密度峰值聚类算法的特性，该方法可以适用于含噪声场景，相比于原密度峰值算法具备更好的准确性。整体模型可以适用于有标签，标签稀缺和无标签的数据流环境，相比传统的数据流挖掘方法更具优势。

（2）基于密度峰值聚类和微簇结构的数据流异常检测方法。

在数据流聚类挖掘中，异常会影响模型的聚类质量，为保证模型聚类性能，需要及时检测异常，提前规避风险。针对这些问题，本研究提出一种基于密度峰值聚类与微簇结构的数据流异常检测方法。该方法包含学习模块和异常检测模块：学习模块依据密度聚类原则，将数据映射至微簇；异常检测模块将微簇作为分析单元，通过改进的 K 近邻思想优化局部密度定义，采用改进的密度峰值聚类算法，对微簇进行区域划分，基于区域信息和簇内信息，计算异常评分。为确保方法的准确性与可靠性，本文在人工数据集和真实数据集均进行验证，实验表明该方法能有效地检测异常，提高模型聚类性能。

1.4 论文结构安排

全文共分为五章，每章的内容安排如下。

第一章绪论，阐述流式数据聚类挖掘背景以及研究意义，介绍与当前研究工作相关现状与亟须解决的问题，最后介绍本文工作内容。

第二章相关研究工作，介绍动态数据流、概念漂移等相关概念、介绍数据流聚类标准、最后介绍模型评价指标。

第三章设计一种基于密度峰值的数据流动态聚类方法，解决流环境中数据不确定性和噪声等因素影响聚类性能问题。该方法分为在线－离线框架，在线阶段采用微簇结构和不均匀衰减策略提高内存使用率；离线阶段改进局部密度定义方法，通过实验对方法进行验证。

第四章设计一种基于密度峰值聚类和微簇结构的数据流异常检测方法，解决数据流聚类模型易受异常和概念漂移等因素影响聚类性能问题。该方法采用在线框架，包含微簇学习模块和异常分析模块，微簇学习模块收集数据关键信息初步聚类，异常分析模块设计改进的密度峰值聚类算法二次聚类，并提出一种新的全

局异常分析策略，最后通过实验进行验证。

第五章总结和展望。对本文工作进行总结，对工作的优势和不足进行分析，给出未来工作的研究方向。

第 2 章　相关工作

本章阐述数据流聚类研究的相关工作。介绍基础概念与知识，包括数据流、概念漂移定义、概念漂移产生原因、数据流聚类标准与框架最后介绍本文实验模型评价指标。

2.1 数据流基本概念

2.1.1 动态数据流

从数据排列角度来说，数据流是一组带有时间序列标记的数据，流中的数据按照时间顺序依次被处理器接收，数据流可以表示为 $S=\{x_1,x_2,\cdots,x_i,x_{i+1}\}$，样本的脚标表示样本在时间戳中的序数，$i$ 可以无限大，其中单个样本可表示为 $x_i=\{x_i^1,x_i^2,\cdots,x_i^n\}$，$n$ 代表样本的维数，越大表明样本的维度越高，处理起来更复杂。这些样本分布在数据空间中，用特征向量的方式，表示样本在空间中的位置。此外动态数据流具备如下特性[3-5]：

（1）数据连续性：在数据流中，数据是高速到达，每个数据时间间隔很短保持连续性。

（2）数据时序性：在数据流中数据间传输是有序的，同一时间点往往只能处理单个数据。

（3）数据海量性：数据流挖掘领域认为数据流中数据可以是无限的，无法将所有数据都保存。

（4）数据有序性：数据流环境处理数据必须按顺序读取，无法像静态环境一次性读取所有数据。

（5）数据不确定性：数据流蕴含各种数据，这些数据分布复杂，且数据形状可能是随机未知的，甚至存在噪声或异常等。

数据流中的概念（concept）是指不同类型数据在特征空间的分布信息。在流环境中，数据分布信息可能随时间发生变化，根据特征向量或类别的变化形式分为概念漂移、概念演化以及特征演化等。概念漂移是数据流环境常见的一种情景，通常是数据的特征分布发生变化。

本章在后续 2.1.2 节详细介绍概念漂移产生原因，及概念漂移种类。

2.1.2 概念漂移

在数据流传输过程中，如果t时刻数据分布和u时刻数据分布发生变化则认为发生概念漂移，即$P_t(X,Y) \neq P_u(X,Y)$，其中Y表示样本在现实中的真实标签，X则表示样本在特征空间中的分布信息。受到现实环境数据分布影响数据流中数据分布信息发生变化是必然的，概念漂移成为数据流聚类挖掘必须解决的问题之一。根据分类边界随时间变化快慢，可以将概念漂移分为突变型漂移（Sudden Drift）、渐变型漂移（Gradual Drift）、增量型漂移（Incremental Drift）以及重现型漂移（Reoccurring Drift）四种类型[72]。

突变型漂移：指数据分布在极短时间内发生剧烈变化，导致原有的概念突然被新的概念完全取代。这种漂移的产生通常是由于突发性事件或环境剧变，例如政策变更、市场崩盘、网络攻击等。典型的例子包括金融市场的剧烈波动，例如某国突然实施经济制裁导致股市暴跌，或者企业因丑闻曝光引发用户行为的迅速转变。此外，网络安全领域中的新型恶意软件攻击也可能导致传统检测系统失效。应对突变型漂移的方法一般包括快速检测和自适应调整，如采用滑动窗口更新模型、使用数据流学习算法或基于异常检测机制及时识别变化。

渐变型漂移：指数据分布在较长时间内逐渐变化，通常由长期趋势或社会、技术进步引起。这种漂移的转变过程较为缓慢，并且在过渡期内原有的概念可能仍然在一定时间内有效。例如，随着技术的发展，消费者购物行为逐渐从传统的实体店购物转向在线购物。此类漂移往往会影响到传统的零售预测模型，因为模型的预测依据可能随着消费者行为的变化而变得不再准确。应对渐变型漂移的方法通常包括自适应学习算法，它能够根据新数据的到来逐步调整模型。常见的策略有时间加权学习（对新数据给予更高的权重）和主动学习（不断更新训练数据集以反映新的数据分布）。

增量型漂移：指数据分布随着时间的推移持续变化，尽管单次变化可能较小，但长期累积后会导致显著的概念变化。这类漂移的产生原因通常是系统内部缓慢演变或长期趋势变化，例如语言的演化、城市交通流量的增长或个体兴趣偏好的缓慢改变。例如，推荐系统中的用户兴趣会随着年龄、社会环境和生活经历逐渐变化，一个用户可能在年轻时偏好动作片，但随着年龄增长更倾向于剧情片或纪

录片。应对增量型漂移的方法主要包括在线学习和持续模型更新，可以使用增量学习算法，使模型能够逐步适应数据分布的演变，同时采用滑动窗口策略只保留最新的数据，以减少过时信息的影响。

重现型漂移：指的是某个过去的概念在一段时间后再次出现，通常由周期性因素或偶发事件引起。这类漂移可以分为循环重现和非循环重现。例如，在零售行业，每逢节假日消费者的购买行为会发生周期性变化，如“双 11”购物节、圣诞促销等；在疾病传播领域，某些传染病可能在几年后因环境变化或病毒变异再次流行。应对重现型漂移的方法一般包括存储历史概念并进行回溯，例如利用概念存档技术在相同模式出现时迅速恢复旧模型，或者在存在周期性的情况下，采用时间序列预测模型提前预测概念回归的时间点，并动态调整模型适应变化。

2.2 数据流聚类标准与框架

2.2.1 数据流聚类标准

数据流聚类与传统聚类不同，传统聚类是在静态数据集上一次性聚集所有数据，得到的结果是准确值，数据流聚类不能一次性处理所有数据，得到的结果是近似值。数据流聚类通常要满足如下 4 个标准[74]：

（1）单次扫描：在一个时间段数据流到达产生庞大的数据，受限于设备每个样本应当按照时间顺序扫描一次，读取元素；

（2）实时响应：数据流聚类方法必须快速响应并处理到达的数据，以满足数据流到达更新的需求；

（3）有限内存：数据流是动态的无穷的，传统聚类可以一次访问所有数据，流聚类需要依赖设备一次性访问部分的数据；

（4）概念漂移检测：在动态数据流传输中常常会发生概念漂移，数据流聚类方法应当能通过数据分布变化检测概念漂移，并降低概念漂移对聚类结果的影响。

2.2.2 数据流聚类框架

数据流聚类方法通常采用两种处理框架：以增量学习为基础的在线框架和在线－离线框架（两阶段框架）。在线处理框架旨在学习数据后通过增量学习方法，自动更新模型输出预测结果；在线－离线框架旨在在线阶段学习和保存数据概念，

在离线阶段通过用户发出的请求，基于在线阶段学习的概念对数据信息聚类。在线框架通常在实时处理数据时，相对速度更快耗时更短，但是需要更多的计算资源；两阶段框架可以处理高速数据流响应速度稍慢，但需要人工介入发送聚类请求不适合处理某些特殊场景。

本文根据 1.2.1 节中列举的主流的数据流聚类方法，总结出两种一般性的数据流处理框架如图 2-1 和图 2-2 所示。

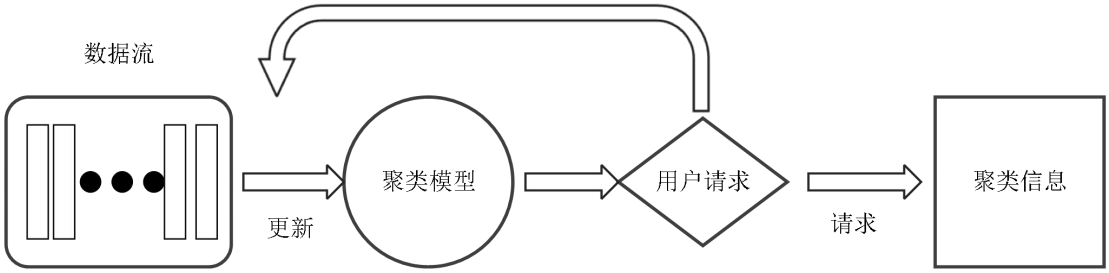


图 2-1 在线框架

Fig.2-1Online Framework.

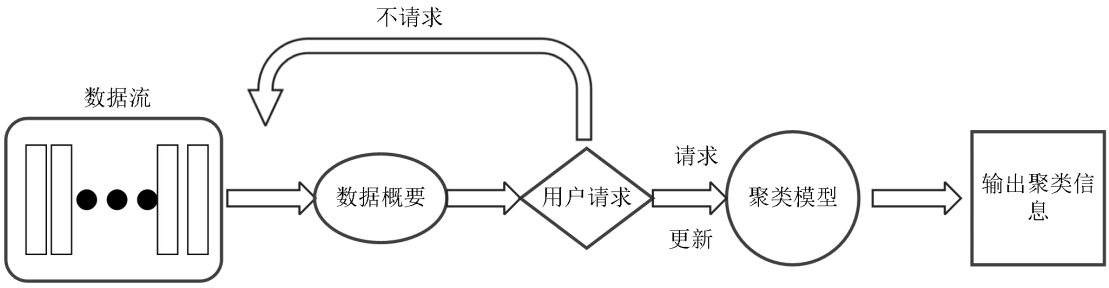


图 2-2 在线－离线框架

Fig.2-2 Online-Offline Framework

2.3 模型评价指标

为评价数据流聚类性能[75,76]和异常检测性能[77]，本文采用外部评价指标：纯净度（*Purity* 简写 *Pur*）、采用精准度（*Precise*）、召回率（*Recall*）和 *F1* 值（*F1 score*）等和内部评价指标轮廓系数（ *Silhouette coefficients* 简写 *SC*）评价模型性能。

（1）纯净度

纯净度计算公式如下：

$$Pur=\left(\sum_{i=1}^{n}\frac{|C_i^d|}{|C_i|}/n\right)\times 100\% \tag{2-1}$$

其中 n 为最终类簇个数 ，$|C_i^d|$ 表示类簇 i 中主导簇号的样本个数，$|C_i|$ 表示类簇 i 中所有样本的个数。纯净度即类簇中同号样本占比的平均值，它的取值范围是(0,1]，纯净度越高说明聚类效果越好。

（2）轮廓系数

轮廓系数 *SC* 不同于纯净度它是一种内部评价指标它不依赖标签信息，考虑样本到距离该样本最近簇中心和到其他簇中心的距离，通过这两个距离的差异来衡量聚类效果。计算 *SC* 前需要先计算类簇的内聚值（*Cohesion* 简写 *Co*）和类簇的离散值（*Separation* 简写 *Se*），*Co* 和 *Se* 计算方法如式（2-2）和（2-3）所示。其中 n 表示目标类簇中样本的数量，$dist(\bullet)$ 表示距离函数，m 表示目标类簇 C_i 的最近类簇 C_j 的样本数量，*SC* 的计算方法如式（2-4）所示。其中 N 表示所有类簇的数量，轮廓系数取值范围为(0,1]，*SC* 值越高表示聚类结果越理想。

$$Co=\sum_{i\in C_i}^{n}(\sum_{j\in C_j}^{m}dist(i,j)/m) \tag{2-2}$$

$$Se=\sum_{i,j\in C_i}^{n}dist(i,j)/n \tag{2-3}$$

$$SC=\frac{1}{N}\sum_{K=0}^{N}\frac{Se-Co}{\max(Co,Se)} \tag{2-4}$$

（3）精准度

精准度计算公式如下：

$$\mathrm{Pr}ecise=\frac{TP}{TP+FP} \tag{2-5}$$

其中 *TP* 预测正确的异常样本数量，FP 是预测错误的异常样本数量。

（4）召回率

召回率计算公式如下：

$$\mathrm{Re}call=\frac{TP}{TP+FN} \tag{2-6}$$

（5）*F1* 值

F1 值计算公式如下：

$$F1 = 2 \times \frac{\Pr esise \times \mathrm{Re} call}{\Pr esise + \mathrm{Re} call} \tag{2-7}$$

F1 值在精确率和召回率之间提供一个平衡，特别是在数据集不平衡时单独依赖精确率或召回率可能会产生误导。

2.4 密度峰值聚类算法

密度峰值聚类算法通过样本间密度处理样本，算法依赖两个前提：聚类中心样本的局部密度高于其他样本；每个局部密度最大值样本到其他局部密度最大值样本的距离很远。

DPC 算法的步骤如下：首先计算每个样本的局部信息 ρ；在此基础上筛选局部较大值样本，计算样本间的相对距离 δ；将局部密度较低的样本，分配给距离样本最近且局部密度较高的样本，分析样本间的局部密度和相对距离，计算每个样本的决策值 γ 并绘制决策图。依据决策值选择聚类中心，以聚类中心为起点，将被分配样本拉入自己的类簇，层层迭代直到所有样本都被分配。

局部密度 ρ 反映目标样本周围样本数量的稀疏程度，局部密度越大说明越密集，局部密度有两种定义方法如式（2-8）和（2-9）所示。

$$\rho_i = \sum_{j \neq i}^{N} X(d_{ij} - d_c) \tag{2-8}$$

$$\rho_i = \sum_{j \neq i}^{N} e^{(\frac{d_{ij}}{d_c})} \tag{2-9}$$

这两种方法分别是：基于截断核函数和高斯核函数计算局部密度。高斯核函数通常应用在小规模数据集，截断核函数通常用在大规模数据集。式（2-8）是截断核函数的局部密度计算方法，其中，*N* 是所有样本数量，设定截断距离是 d_c 样本和目标样本的距离记作 d_{ij}，计算以目标样本为中心 d_c 为半径的圆内样本数量为样本的局部密度。式（2-9）是高斯核函数计算方法，将目标样本到周围样本的高斯距离和作为局部密度。

相对距离 δ 是局部密度较低样本到局部密度高样本的最短距离，若是局部密度最高的样本则相对距离是到最远样本的距离，计算方法如式（2-10）所示。

$$\delta_i = \begin{cases} \min(d_{ij}), if\ \rho_i = \rho_{\max} \\ \max(d_{ij}),\ otherwise \end{cases} \quad (2\text{-}10)$$

决策值是聚类中心选择标准，通过局部密度和相对距离计算每个样本的决策值，如式（2-11）所示。

$$\gamma_i = \rho_i * \delta_i \quad (2\text{-}11)$$

密度峰值聚类算法也存在一个缺陷，所有样本像多米诺骨牌一样，低密度样本分配给高密度样本，若其中一个样本分配错误，会导致一连串的分配错误降低聚类精准性，本章在后续研究中通过进一步研究工作降低“多米诺效应”影响。

2.5 数据流聚类与异常检测基本概念

2.5.1 数据流聚类窗口机制

由于数据流的特性，同一时间只能对一部分数据聚类。因此，哪些样本信息值得保留和哪些样本信息应当丢弃，引发学者们的讨论，数据流窗口机制也由此应运而生。现阶段常见的数据流窗口机制有四种：滑动窗口机制、衰减窗口机制、界标窗口机制和倾斜时间窗口[70]。

滑动窗口：在这个机制中窗口的大小是固定的，但窗口的位置会随着时间的推移滑动，滑动窗口可以根据接收方的处理能力动态调整。

衰减窗口：衰减窗口依赖衰减函数，控制窗口的边界，常见的衰减函数有 $f(t) = e^{-\lambda(t-t_p)}$，$t$ 为当前时间，t_p 为上一次更新时间，$e^{-\lambda}$ 为时间衰减因子。为表示方便，在后续的内容中一致使用 $f(\bullet)$ 表示时间衰减函数，这种方法通常用在数据流实时处理场景中，可以保存经常更新的样本信息，删除长时间不更新的样本。

界标窗口：界标窗口是指在数据流中设置特定的界标（或标记），以便在处理数据时能够快速定位和管理数据。界标可以是时间戳、序列号等用于标识数据的起始和结束。

倾斜时间窗口：倾斜时间窗口是一种在时间窗口中引入倾斜的方式，允许在特定时间段内对数据进行加权处理。这种机制适用于需要对数据进行实时分析和处理的场景，能够更好地捕捉数据的变化趋势。但随着时间的推移方法的稳定性还是可能下降。

2.5.2 微簇结构

微簇结构在 Clustream 方法中被[25]提出，通过三元组的形式保存数据概要，例如：$\{\overrightarrow{CF1},\overrightarrow{CF2},n\}$，$\overrightarrow{CF1}$，$\overrightarrow{CF2}$表示线性和与平方和，$n$表示微簇内样本的数量。

定义 1（线性和，平方和）$\overrightarrow{CF1}$ 、$\overrightarrow{CF2}$表示的微簇内样本属性的线性和以及平方和，计算方法如式（2-12）、（2-13）所示，其中x_i表示样本的属性。

$$\overrightarrow{CF1}=\sum_{i=1}^{n} f(t-t_p)*x_i \tag{2-12}$$

$$\overrightarrow{CF2}=\sum_{i=1}^{n} f(t-t_p)*x_i^2 \tag{2-13}$$

定义 2（簇心）簇心代表微簇在特征空间中位置，采用加权求平均的方法计算簇心，可以更精确反映微簇内数据分布情况，如式（2-14）所示。

$$C_i=\frac{\overrightarrow{CF1}}{\omega_{(C_i,t)}} \tag{2-14}$$

其中，初始的簇心位置即为样本在空间中的位置。

定义 3（微簇权重）微簇用权重表示微簇内样本的密集程度，微簇在t时刻的权重是微簇内所有样本的权重之和。传统的数据流聚类方法如 Denstream 方法对每个新插入样本，权重默认为 1，更新权重的计算方法如式（2-15）所示。

$$\omega_{(C_i,t)}=f(t)*\omega_{(C_i,t_p)}+1 \tag{2-15}$$

但是，这种加权方法存在一个弊端，新的样本到达恰好处在聚类半径的边缘位置，此时该样本的权重会超过类簇中任何样本，簇心将朝边缘方向移动，导致簇心不能精确反映样本的分布情况。

定义 4（微簇半径）微簇半径是样本可插入的距离范围，到簇心距离小于半径的样本可以插入到微簇中。微簇半径随簇内样本数量增加而增加，为防止微簇无限制扩张，微簇的半径不能超出半径阈值μ，计算方法如式（2-16）所示。

$$r_i=\sqrt{\frac{|\overrightarrow{CF2}|}{\omega_{(C_i,t)}}-(\frac{|\overrightarrow{CF1}|}{\omega_{(C_i,t)}})^2} \qquad ,r_i<\mu \tag{2-16}$$

定义 5 （核心微簇(Core Micro-cluster:CMC)） 指一组包含大量邻近样本的

数据集合，微簇权重超过核心阈值ε对数据聚类性能影响大。

定义 6 （潜在微簇(Potential Core Micro-cluster:PMC)）指一组包含较多邻近样本的数据集合，微簇权重$\eta < \omega < \varepsilon$ 对数据聚类有一定影响，其中η为潜在微簇阈值。

定义 7 （离群微簇(Outlier Micro-cluster:OMC)）指一组包含较少样本的数据集合，权重低于潜在阈值η 由于这些数据和其他数据关联性弱，因此在聚类时应该将这些数据当作噪声处理。

2.5.3 异常检测基本概念

数据流异常检测方法大致分为三种：基于预测的异常检测方法、基于分布的异常检测方法、基于聚类的异常检测方法。聚类技术可以快速将特征相近的样本划分到一起，与其他类型的异常检测方法相比基于聚类方法拥有更强的泛化能力和特征学习能力。

基于聚类的异常检测方法从学习框架角度可以分为两类：基于在线框架的异常检测方法和基于两阶段框架的异常检测方法。

两阶段的异常检测方法，需要在内存中保存重要样本的数据概要，等待用户发出更新指令，用数据概要来更新方法的整体模型。与传统的两阶段聚类挖掘方法不同，在输出聚类结果后，异常检测模型需通过异常分析策略对聚类结果进行筛选，并向用户反馈是否存在异常。

在线异常检测方法通常采用增量学习策略。初始化方法后，模型能够自动学习新样本，更新聚类模型，无需额外设置缓冲区等待用户指令。模型更新是自动进行的，使得聚类信息可以实时输出无需用户操作干预。

这些异常检测方法通常需要满足四个条件[83]：

（1）实时性：数据流是高速传输的，假设当前时刻为t模型需要在$t+1$时刻前预测出系统的行为是否正常。

（2）准确性：准确性是衡量方法性能的重要标准，一个好的方法应该能够正确地确定异常，避免误报和漏报。

（3）动态适应性：流数据的特征随着时间的推移演变，这被称为概念漂移。方法必须适应这种动态变化，才能对当前数据点做出准确的判断。

（4）内存有限：流数据的数据量是无限的，但内存是有限的，所以模型不

能存储所有的历史数据。该方法应保留最有效的历史信息，并减少内存占用。

2.6 本章小结

本章对相关工作进行阐述，具体包括对数据流定义、概念漂移生成原因、数据流聚类标准及框架，最后介绍模型评价指标。后文将针对具体研究内容展开描述。

第 3 章　基于密度峰值的数据流动态聚类方法

数据流环境普遍存在数据形状不确定性和噪声等因素，这些因素影响数据流聚类模型的聚类性能，因此解决这些问题成为研究热点之一。为此本章提出一种基于密度峰值的数据流动态聚类方法。通过将密度峰值聚类算法数据流化，使模型具备识别任意形状数据和不易受噪声影响等特性。该方法采用在线－离线框架。在线阶段，实时响应处理不断到达的数据流，设计一种非均匀衰减策略，削弱历史数据对聚类结果的影响，依据样本到微簇的距离，采用动态加权方法对样本进行加权处理。离线阶段，基于密度峰值聚类原理，本章设计一种基于最近邻域自适应的局部密度计算方法，改良局部密度定义，降低 DPC 算法分配阶段“多米诺效应”影响，提高模型准确性。

3.1 引言

数据流是无限的，数据序列传输速度快，随着时间的推移而演变，这需要挖掘模型具有很高的响应速度。数据流中数据充满不确定性，存在大量任意分布的数据[78-79]和噪声[73]。这就使得一些传统的数据流聚类方法，识别数据效率低并容易受到噪声干扰，为降低噪声的干扰许多学者将目光投向密度峰值算法。DPC 算法对噪声有很好的抵抗能力，可以识别任意形状数据。但 DPC 算法复杂度过高，分配阶段的“多米诺效应”导致聚类性能不稳定，不适合数据流环境。

本章旨在解决传统数据流聚类方法，易受数据形状不确定性和噪声等因素影响，导致聚类性能降低的问题。通过改进 DPC 算法局部密度定义方法，降低分配阶段“多米诺效应”影响和将 DPC 算法数据流化，达成预定目标。具体工作如下：（1）设计微簇的不均匀衰减策略，更新权重信息降低历史数据对聚类效果的负面影响；利用动态加权分配策略，动态反应微簇在数据流中的位置变化，降低边缘样本对类簇的影响。（2）提出一种基于最近邻域的自适应局部密度计算方法，考虑邻居间的相互关系，降低密度峰值聚类算法分配阶段“多米诺效应”的影响。

3.2 基于密度峰值的数据流动态聚类方法

3.2.1 框架结构

本章为解决以下两个关键问题：（1）如何处理含噪声的数据流，减少噪声的影响；（2）如何使用 DPC 算法解决数据形状不确定性问题，降低 DPC 算法中“多米诺效应”的影响。结合上述知识，提出一种基于密度峰值的数据流动态聚类方法命名为 RDDPCStream （Robust Dynamic Density Peaks Clustering for Data Stream）方法。该聚类方法采用在线－离线框架，在线阶段包含两个模块用于构建和维护微簇；离线阶段，提出最近邻域和密度系数重新定义局部密度，利用微簇近邻关系降低分配阶段“多米诺效应”影响，以微簇为单位，对数据进行密度峰值聚类。RDDPCStream 方法框架，如图 3-1 所示，在线阶段分为两个模块，分别是在线合并模块和在线更新模块；离线阶段则是密度峰值聚类模块。

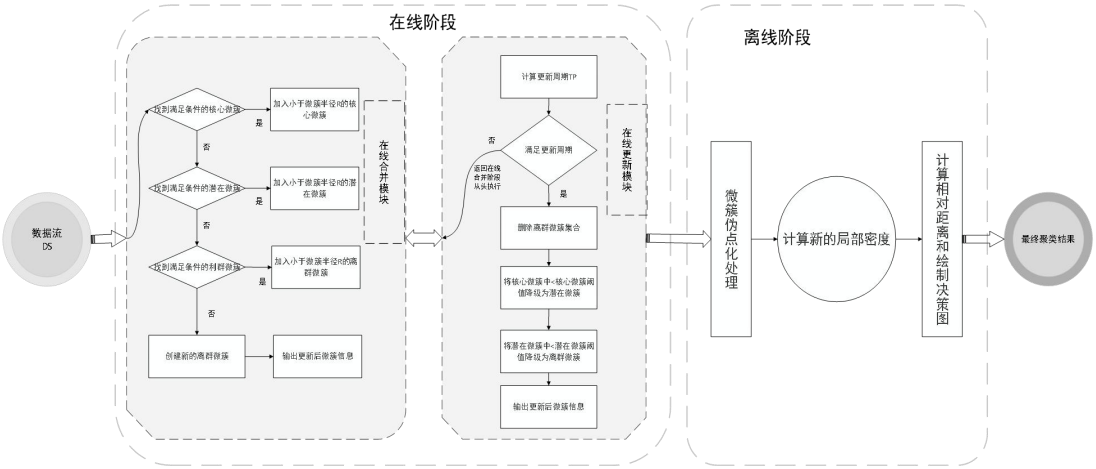


图 3-1 基于密度峰值的数据流动态聚类方法框架

Fig.3-1 A Framework for Dynamic Data Stream Clustering Based on Density Peaks

3.2.2 在线合并模块

传统的聚类方法按照式（2-15）给插入微簇的样本加权，没有考虑到样本和微簇中心的位置关系，可能导致聚类精准度不准确。本章设计的 RDDPCStream 方法，将样本分配给微簇时，考虑样本和微簇中心的位置关系对插入的样本动态加权，如式（3-1）所示。

$$\omega_{(x_i,t)} = 1 - \frac{dist(x_i, C_i)}{dist(x_i, C_i) + 1} \tag{3-1}$$

其中，$dist(x_i, C_i)$ 表示样本 x_i 与对应簇心的欧式距离，通过对应距离为准则，认为距离簇心越近的样本相似性越高，因此赋予的权重也应该越高。式（3-1）

描述的加权方法，考虑到样本和微簇中心的位置关系为样本动态赋权，当样本在插入阈值 μ 处插入时样本权重值最小，最小权重表示为 $\frac{1}{\mu+1}$，最大权重为 1。

数据流环境数据是动态变化的，一些时间久远的样本，会影响对当前样本聚类效果。对一些早期分散的样本，它们对当前样本的聚类没有帮助，甚至可能干扰聚类结果，这些样本应及时删除。然而，另一种情况是早期较为密集的样本，这部分样本对当前数据聚类的贡献，远大于前述的分散样本。在这种情况下，对每个微簇使用相同的时间衰减函数，控制微簇内样本的变化是不合理的，因为它们在当前聚类中仍然具有较高价值。为降低历史数据对当前聚类的影响，在线更新模块设计基于微簇的不均匀衰减策略，对上述两种情况灵活处理。

与 Clustream 方法的微簇结构不同，本章采用三种不同的微簇来表示微簇结构：核心微簇、潜在微簇、离群微簇。本章基于时间衰减窗口，提出一种不均匀的时间衰减策略。与传统数据流聚类方法不同，该策略并不直接删除潜在微簇，而是提供一个缓冲。当潜在微簇的权重低于潜在阈值时，会被降级为离群微簇，并保留一个周期进行观察。如果该微簇的权重在此期间未能超过潜在阈值，则最终删除该微簇。本章采用四元组的方式来表示微簇结构如： $\{\overrightarrow{CF1},\overrightarrow{CF2},n,\lambda\}$，其中 λ 表示衰减因子对于不同的微簇，采用不同的衰减因子，如式（3-2）所示。

$$\lambda=\begin{cases}\lambda_{high}\ ,\ \omega_{(C,t)}<\eta\\ \lambda_{mid}\ ,\ \eta<\omega_{(C,t)}<\varepsilon\\ \lambda_{low}\ ,\ \omega_{(C,t)}>\varepsilon\end{cases} \tag{3-2}$$

三种衰减因子代表数据流中的微簇在不同权重范围内的衰减状态，根据不均匀衰减策略，将样本 j 插入微簇 C_i 如式（3-3）所示。

$$\omega_{(C_i,t)}=f(t)*\omega_{(C_i,t_p)}+\omega_{(x_j,t)} \tag{3-3}$$

微簇的衰减因子，根据微簇实时权重的变化动态调整。对于核心微簇而言，该类微簇在数据流中，较高的概率存在同类样本。因此，衰减速度会被适当调低，以尽量保留有效的历史数据。当核心微簇退化为潜在微簇时，表示该微簇在数据流中出现同类样本的概率降低，因此可以提高衰减速度。这不仅有助于减少内存的浪费，还能有效保存有价值的微簇信息。通过精确控制衰减因子，能够有效识别噪声点并删除，提升方法的鲁棒性。

算法 1 详细阐述在线阶段微簇学习的算法伪代码，在此阶段模型将新到达的样本分配给距离最近且小于微簇半径的微簇而后更新微簇信息，若没有满足的微簇则为该样本创建新的微簇。

算法 1 在线合并

输入： x_i 、 ε 、 η 、 $\lambda_{low},\lambda_{mid},\lambda_{high}$ 、 μ

输出： CMC 、 PMC 和 OMC 集合

1. 初始化微簇；
2. **if** $dist(x_i,C_i)<r_i,C_i\in CMC$ **then**
3. 由式（3-1）计算 x_i 权重，插入微簇并更新微簇权重信息 $\omega_{(C_i,t)}$ ，半径 r_i ，衰减因子 λ ；
4. **else**
5. **if** $dist(x_i,C_j)<r_j,C_j\in PMC$ **then**
6. 由式（3-1）计算 x_i 的权重，插入微簇并更新微簇权重信息 $\omega_{(C_j,t)}$ ，半径 r_j ，衰减因子 λ ；
7. **if** $\omega_{(C_j,t)}>\varepsilon$ then
8. 从 *PMC* 列表中将微簇删除，将微簇加入 *CMC* 微簇列表和调整微簇对应的 λ ；
9. **end if**
10. **else**
11. **if** $dist(x_i,C_k)<r_j,C_k\in OMC$ **then**
12. 由（3-1）计算 x_i 权重，插入并更新新微簇权重信息 $\omega_{(C_k,t)}$ ，半径 r_κ ，衰减因子 λ ；
13. **if** $\omega_{(C_k,t)}>\eta$ then
14. 从列表中将微簇删除，并将微簇加入 *PMC* 微簇列表和调整微簇对应的 λ ；
15. **end if**
16. **else**
17. 为 x_i 创建一个新的潜在微簇更新微簇权重信息 $\omega_{(C_z,t)}$ ，半径 r_z ，
18. 衰减因子 λ ；
19. **end if**
20. **end if**
21. **end if**

3.2.3 在线更新模块

现阶段大多数数据流聚类中使用时间衰减窗口，在删除微簇时对每个微簇都判断一次删除或保留，但这种方法会对模型处理速度造成负担，影响实时响应性能。因此，为解决这一问题本章根据潜在微簇转化为核心微簇的最短时间，以及潜在微簇退化为离群微簇的最短时间，设计两个更新周期 TP_1 和 TP_2 ，计算方法如式（3-4）和（3-5）所示。定义 TP_1 为核心微簇退化为潜在微簇的最短时间， TP_2

为潜在微簇退化为核心微簇的最短时间。

$$TP_1 = \frac{1}{\lambda_{low}} * \log_2(\frac{\varepsilon}{\varepsilon - \frac{1}{\mu+1}}) + 1 \quad (3\text{-}4)$$

$$TP_2 = \frac{1}{\lambda_{mid}} * \log_2(\frac{\eta}{\eta - \frac{1}{\mu-1}}) + 1 \quad (3\text{-}5)$$

式（3-4）是由式$2^{-\lambda_{low}TP_1}\varepsilon + \frac{1}{\mu+1} < \varepsilon$推导得到，它表示核心微簇降级成潜在微簇所需要的最短时间，最终解得TP_1取值范围为$TP_1 > \frac{1}{\lambda_{low}} * \log_2(\frac{\varepsilon}{\varepsilon - \frac{1}{\mu+1}})$。同理，解得$TP_2 > \frac{1}{\lambda_{mid}} * \log_2(\frac{\eta}{\eta - \frac{1}{\mu-1}})$，由此得出式（3-5）潜在微簇的更新周期$TP_2$。

因潜在微簇的衰减速度大于核心微簇，潜在微簇更新周期TP_2小于核心微簇更新周期TP_1。那么核心微簇至少要进行两次降级才会被删除，且不会占用过多的内存空间。因此这种方法可以减少历史数据中的关键信息，不会对内存造成过高的负担。

3.2.4 密度峰值聚类模块

为提高聚类准确性，减少 DPC 算法分配阶段“多米诺效应”的影响。在处理微簇前，以微簇的权值为初始权重。提出一种新的局部密度计算方法，考虑到邻域间的相关性，用最近邻域代替传统的截断距离。在该方法中，若数据集中微簇C_i到另一微簇C_j的距离最短，那么认为C_i和C_j的都在对方的最近邻域中，为提高邻域内样本的相关性在计算局部密度时，引入密度系数概念ζ，计算方法如式（3-6）所示。

$$\xi_i = e^{(\frac{1}{K_{ij}+1})} - 1 \quad (3\text{-}6)$$

其中表示在最近邻域内的近邻情况，K_{ij}越小表示两者越接近，权重系数也越大，K_{ij}为 1 时表示互为最近邻，此时密度系ξ_i数取到最大值。

算法 2 中详细介绍在线阶段更新模块的算法伪代码，在此阶段模型判断自身是否需要更新微簇数量，当满足更新条件后检查核心微簇中是否有低于核心阈值的微簇，若有则将微簇降级为潜在微簇。当满足潜在微簇更新时间时，删除所有离群微簇，模型检查潜在微簇中是否有低于潜在阈值的微簇，若有则降级为离群微簇。

算法 2 在线更新

输入：CMC ,PMC 和OMC 集合、ε、η、t

输出：CMC 、PMC 集合

1. 按照式（3-4），（3-5）计算更新周期TP_1, TP_2；

2. **if**（$t-t_p$)%$TP_1==0$ **then**

3. 在 CMC 集合搜索低于核心阈值ε的微簇，移入 PMC 集合并更新衰减因子λ；

4. **end if**

5. **if**（$t-t_p$)%$TP_2==0$ **then**

6. 清空离群微簇OMC 集合；

7. 在 PMC 集合搜索低于潜在阈值η的微簇，移入离群微簇OMC 集合并更新衰减因子λ；

8. **end if**

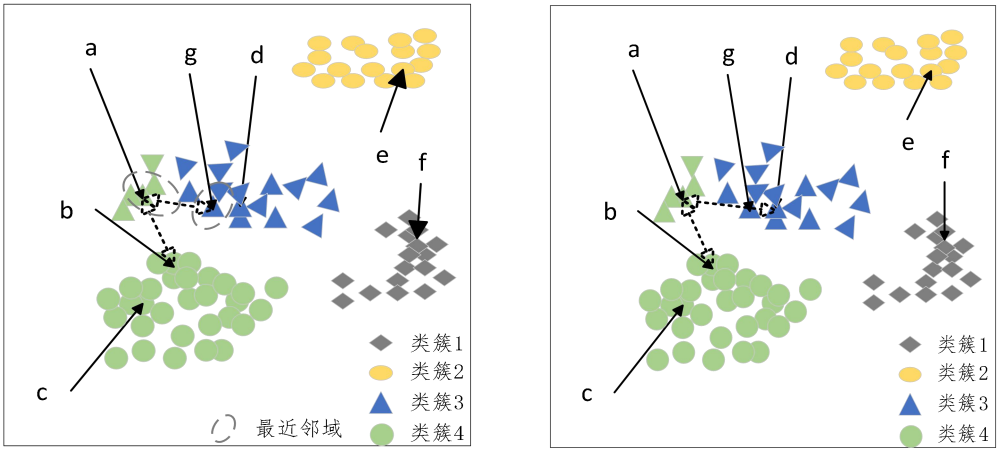
在计算局部密度时，除考虑微簇自身的权重外，还将邻域内微簇的权重纳入考量。具体而言，位置关系越接近的微簇，对局部密度的贡献权重越高，局部密度的计算方法，如式（3-7）所示。

$$\rho_i = \omega_{(C_i,t)} + \sum_{j\in\varpi_i} \omega_{(C_j,t)} \bullet \xi_i \qquad (3\text{-}7)$$

其中ϖ_i代表微簇C_i最近邻域，ξ_j表示邻域内伪点的密度系数。这种局部密度计算方法，考虑到每个微簇初始权重不同，对那些邻域内含有高权值样本但自身权值低的样本，在计算局部密度时可以得到较高的局部密度，但又不会超过邻域内高权值样本的局部密度。

传统的 DPC 算法使用截断距离，如计算局部密度需要手动输入截断距离参数，没有考虑到邻域信息，在算法的分配阶段容易发生“多米诺效应”。如图 3-2 所示，图中有四个簇，c、d、e、f为簇中心（初始权值高）a 是局部较大值样本，样本 g 存在最近邻域，在密度峰值算法中距离 a 样本最近的局部较大值样本为 b 样本，非同簇的 d 样本，因此 a 样本连带其余样本都错误划分给类簇 4。在 RDDPCStream 方法的分配策略考虑样本的邻域信息，能够自适应地计算出样本的局部密度，g 样本因为最近邻域内存在样本 d 贡献权值，也成为局部密度较

大值样本。比较最近较大值样本，会优先考虑 g 样本，避免分配错误。



(a)DADS　　(b)DPC

图 3-2 RDDPCStream 与 DPC 的分配策略图

Fig.3-2 Allocation strategy diagram for RDDPCStream and DPC

本章在算法 3 中详细阐述局部密度的算法伪代码，在此阶段模型将每个微簇视为一个样本，并将相交的微簇也看作一个样本。为每个微簇找到最近的微簇并将双方都加入最近邻域，微簇最近邻域里的样本越多说明越可能是聚类中心，最近邻域内部按照距离降序排列，计算密度系数，而后计算局部密度。算法 4 则是在执行完算法 3 后对已经计算好局部密度的样本们，计算相对距离和决策值，依据决策值选出聚类中心，完成二次聚类。算法 5 是整个聚类方法的运行步骤，先学习样本，再更新数据概要，最后根据更新的概要对样本二次聚类输出最终聚类结果。

算法 3 局部密度

输入： CMC、PMC 集合

输出： ϖ 和 ρ

1. **if** $PMC \neq \emptyset$ **then**

2. 合并相距不超过两倍半径的核心微簇与潜在微簇；

3. **end if**

4. **for each** $C_i \in C$ **do**

5. 找到最近的点标记为 C_j；

6. C_j 加入 C_i 的最近邻域 ϖ_i，将两者的距离记为 $dist(C_i, C_j)$ 存入 ϖ_i；

7. C_i 加入 C_j 的最近邻域 ϖ_j，将两者地距离记为 $dist(C_i, C_j)$ 存入 ϖ_j

8. **end for**

9. **for each** $\varpi_i \in \varpi$ **do**

10. 按距离降序排列，由公式（3-6）计算密度系数 ξ_j；

11. **end for**

12. **for each** $C_i \in C$ **do**

13. 根据微簇 C_i 的 ϖ_i 信息，由式（3-7）计算局部密度 ρ_i；

14. **end for**

算法 4 详细阐述密度峰值聚类部分伪代码

算法 4 密度峰值聚类模块

输入： CMC、PMC 集合、聚类中心个数 c

输出： 最终聚类结果

1. C_i 执行算法 3，计算局部密度和最近邻域；

2. **for each** $C_i \in C$ **do**

3. 由式（2-10）计算微簇 C_i 相对距离表示为 δ_i；

4. 由式（2-11）计算微簇 C_i 决策 λ_i；

5. **end for**

6. **for each** $C_i \in C$ **do**

7. 输出决策值前 c 的 C_i 作为聚类中心；

8. **end for**

9. 将剩余点分配给微簇的上级点，输出聚类结果

为方便理解，本章整理 RDDPCStream 方法的算法伪代码，如算法 5 所示。

算法 5 RDDPCStream

输入： x_i、核心阈值 ε、潜在阈值 η、半径阈值 μ、衰减因子 $\lambda_{low}, \lambda_{mid}, \lambda_{high}$

输出： 最终聚类结果

1. 调用算法 1

2. **if** 满足更新条件 **then**

3. 调用算法 2

4. **end if**

5. 调用算法 4

3.3 实验结果与分析

3.3.1 实验设置

本次仿真实验在 PyCharm 集成开发环境中进行。为模拟现实数据流环境的噪声，本章从噪声影响对象的角度出发，将噪声分为两种：属性噪声和标签噪声[73]。属性噪声：通常表现为数据中的单个或多个特征值不符合正常分布，或者受到外界因素干扰而产生偏差；属性噪声会直接影响数据的质量，进而降低数据分析和挖掘的效果。标签噪声：指数据流中样本标签的异常或错误。标签噪声可能是由人工标注错误、标注标准不一致、标签模糊不清或采集过程中的标注不当等原因引起的。标签噪声可能导致训练出的模型对数据流的预测不准确，因为模型在训练时依据错误的标签信息进行学习。

鉴于数据流聚类方法预测数据，无需考虑标签信息，本实验仅关注属性噪声，并未涉及标签噪声。实验所用的数据集包括人工合成数据集和真实数据集。人工合成数据集通过 Numpy 库生成，分为无噪声数据集和噪声数据集两种类型。其中，无噪声数据集为原始数据集；噪声数据集则是在原始数据集的基础上，按照不同的噪声比例（分别为 15%和 30%）添加干扰项。目的是满足聚类方法的需求，数据集包含 10 个簇中心、2 个特征，总计 50000 条记录，为低维数值型数据。为便于后文引用，数据集分别命名为：无噪声数据集 CDS1，含噪声数据集 CDS2（噪声比例为 15%）和 CDS3（噪声比例为 30%）。

Covtype 数据集包含 54 个数值型特征和 7 个类别，每条记录描述一个特定的森林区域，广泛用于评估数据流聚类方法的性能。为适应数据流环境，本章从 Covtype 数据集中随机抽取 35000 个实例，并进行标准化处理。数据被分为 35 个批次，每个批次包含 1000 条记录，以便进行分组和批次测试。为验证方法在真实数据集上的抗噪声能力，本章对 Covtype 数据集分别添加 15%和 30%的噪声，生成两个噪声版本的数据集，分别命名为 Covtype-N1（噪声比例 15%）和 Covtype-N2（噪声比例 30%），为方便描述将这些数据集统称为 Covtype 系列数据集。

在数据流环境下，本章提出的方法与文献中的 Denstream、Clustream 和 DBSTREAM[77]等方法进行对比。Denstream 和 Clustream 方法在前文有过详细介

绍，DBSTREAM 方法是一种在线的数据流聚类方法，采用微簇结构和共享密度对数据处理，最后采用 DBSCAN 方法对微簇聚类。各对比方法的参数设置如下：

Denstream 方法：密度阈值为 8，搜索半径为 2.5，衰减因子为 0.05；

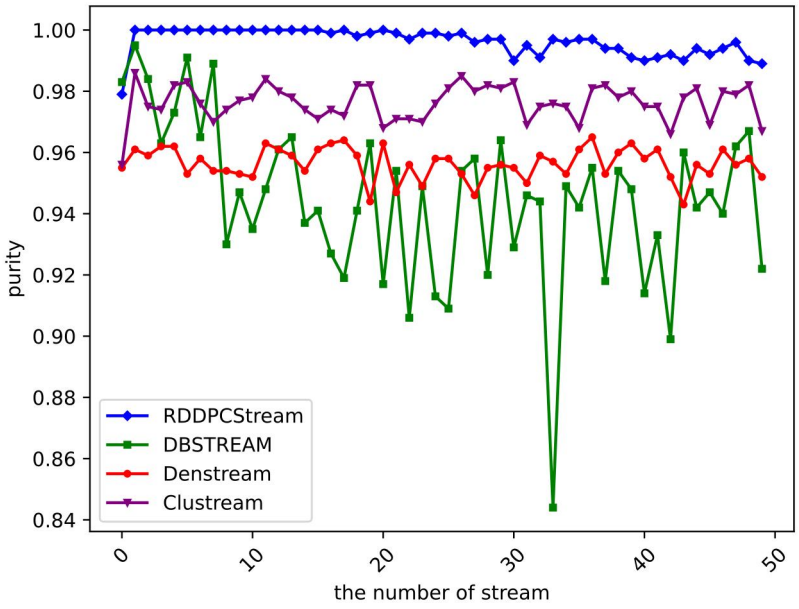
Clustream 方法：微簇数量为 50，衰减因子为 0.05；

DBSTREAM 方法：聚类密度阈值为 8，最小密度阈值为$\frac{8}{3}$，衰减因子为 0.05。

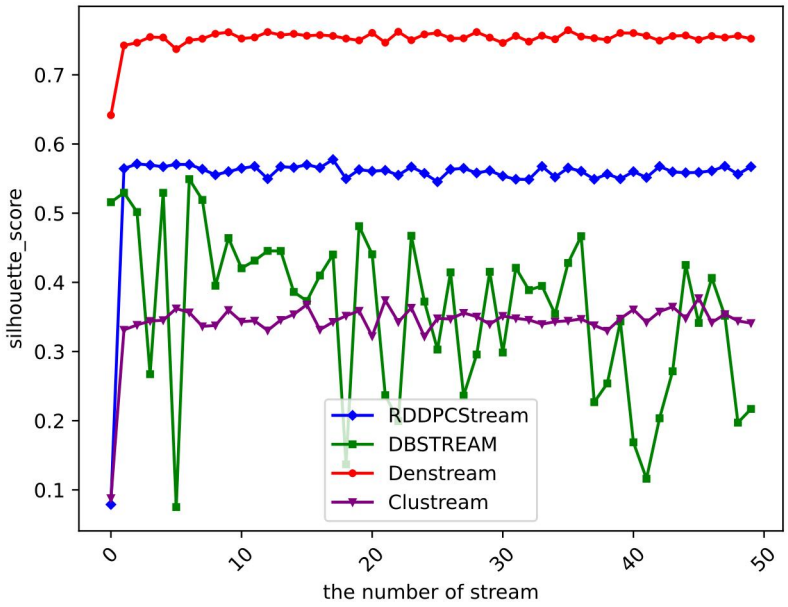
RDDPCStream 方法：核心阈值$\varepsilon=8$，潜在阈值为$\eta=\frac{8}{3}$，半径阈值$\mu=2.5$，衰减因子设置为$\lambda_{high}=0.01$，$\lambda_{\text{mid}}=0.03$，$\lambda_{\text{low}}=0.06$，人工数据集聚类中心个数 c=10，Covtype 系列数据集聚类中心个数 c=7。

3.3.2 人工数据集对比试验

本节通过记录聚类信息的纯净度和轮廓系数，衡量各流聚类方法的聚类性能。RDDPCStream 方法与对比方法在 CSD1、CSD2 和 CSD3 数据集上的实时纯净度和轮廓系数值分别如图 3-3、图 3-4 和图 3-5 所示。



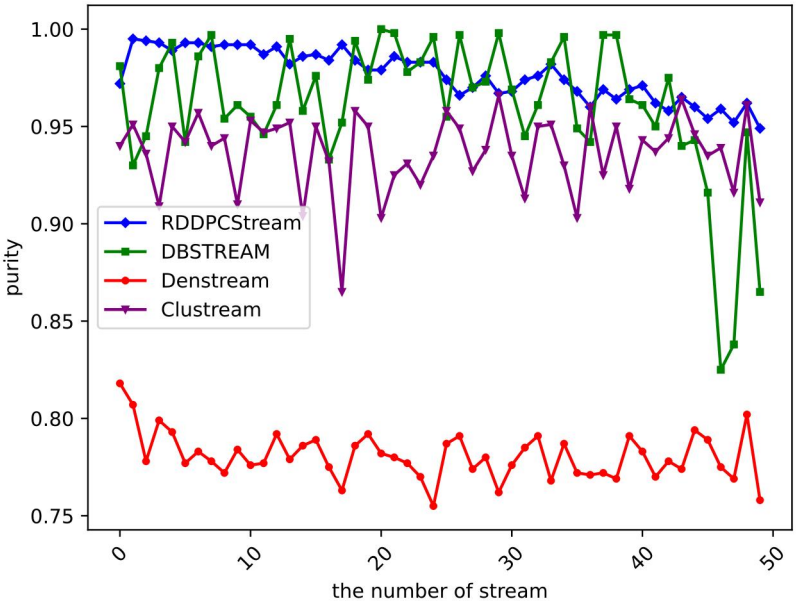
(a)CSD1 纯净度



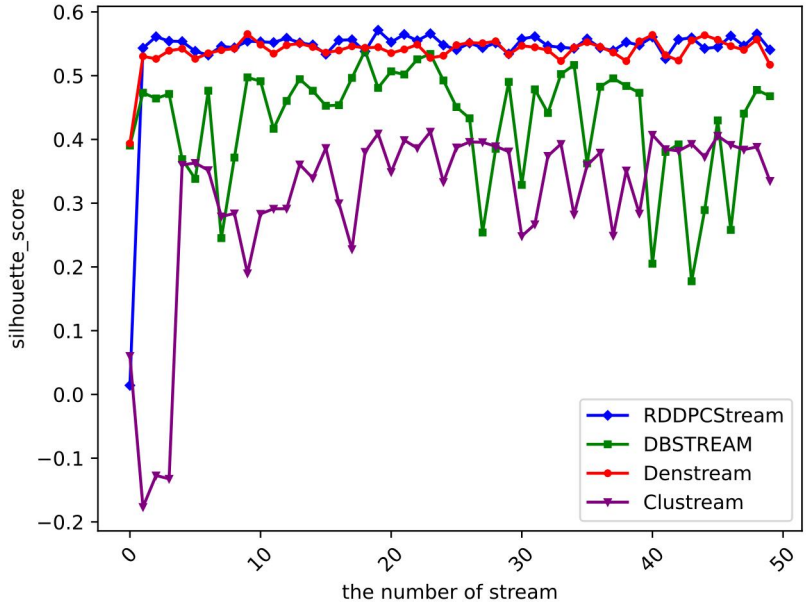
(b)CSD2 轮廓系数

图 3-3 CSD1 上的纯净度和轮廓系数对比

Fig.3-3 Comparison of purity and silhouette coefficientonCSD1



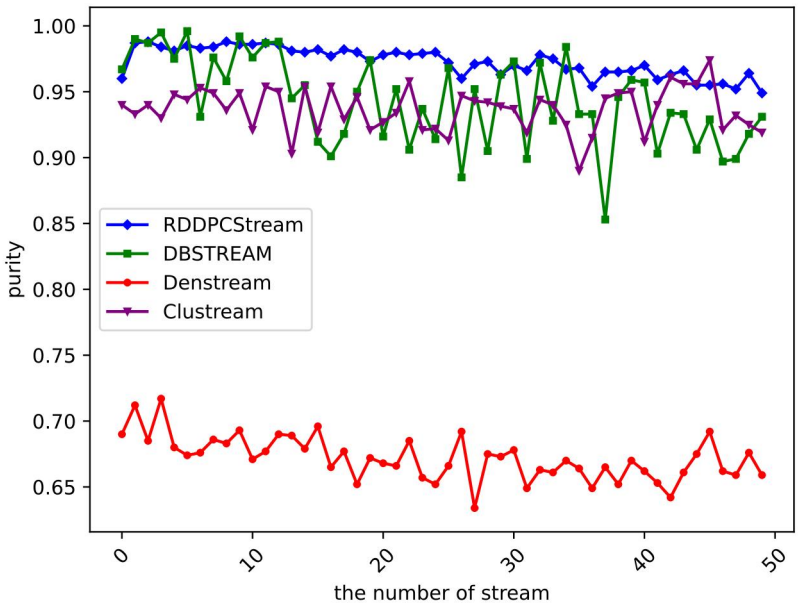
(a)CSD2 纯净度



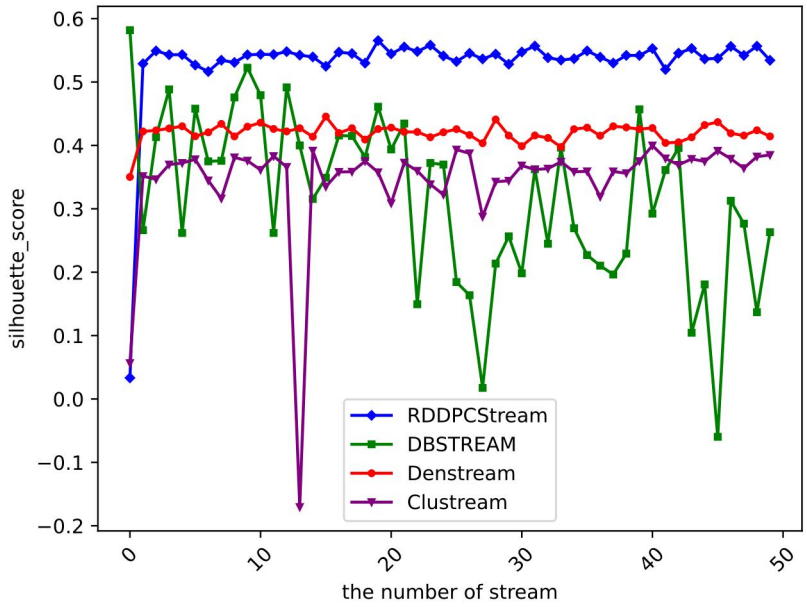
(b)CSD2 轮廓系数

图 3-4 CSD2 上的纯净度和轮廓系数对比

Fig.3-4 Comparison of purity and silhouette coefficientonCSD2



(a)CSD3 纯净度



(b)CSD3 轮廓系数

图 3-5 CSD3 上的纯净度和轮廓系数对比

Fig.3-5 Comparison of purity and silhouette coefficientonCSD3

在无噪声环境下，如图 3-3 所示，四种流式聚类方法均表现出较好的基础性能，四种方法纯净度指标（*Pur*）均超过 0.84。具体而言，Denstream 方法凭借动态密度阈值机制，在 CSD1 数据集中表现出最优的轮廓系数，同时维持高纯净度稳定性，这得益于 Denstream 方法基于核心微簇的增量更新策略，对数据分布的准确刻画；DBSTREAM 方法虽然在部分数据组上纯净度达到 0.99，但受限于固定半径阈值的设计，性能呈现出显著波动，如纯度极差 0.14，轮廓系数极差 0.4 等，表明该方法对初始参数具有较高的敏感性；Clustream 方法通过 K-Means 变体的线性时间特性，实现稳定的高纯度，在部分数据组纯净度超过 0.96，但由于固定簇数的约束轮廓系数始终较低，这揭示该方法的簇结构辨识能力不足。

RDDPCStream 方法在数据流的前十五组数据，表现出理论最优纯净度，并在后续数据流中因概念漂移，产生不到 5%的性能衰减。RDDPCStream 方法基于相对密度的动态剪枝策略，在保持高纯净度的同时，有效维持轮廓系数高于 0.5。尽管在概念漂移的影响下略有性能衰退，RDDPCStream 仍在低噪声环境中表现出较强的鲁棒性。综上所述，Denstream 和 RDDPCStream 在低噪声场景下的表现均处于性能第一梯队。

当数据集加入噪声时，如 CSD2 和 CSD3 数据集，如图 3-4 和图 3-5 所示，各方法在数据集上的稳定性差异明显，综合而言 RDDPCStream 方法的稳定性及性能最佳。Denstream 方法受噪声干扰最为严重，在 CSD3 数据集中的纯净度下降幅度达 20.2%（纯净度从 0.94 下降至 0.75），轮廓系数也下降至 0.5 以下，这与密度聚类方法对离群点的固有敏感性直接相关；DBSTREAM 方法虽高于 0.8，但轮廓系数在 CSD3 出现幅度超过 0.52 的极端波动，反映出该方法阈值自适应机制在噪声环境中的失效；Clustream 方法通过静态簇中心约束将纯净度波动控制在 11%以内，但噪声的引入导致轮廓系数方差扩大近 3.2 倍，表明传统划分式方法在复杂数据流环境中的局限性；相比之下，RDDPCStream 方法展现显著的优势，在 CSD3 数据集上，RDDPCStream 方法保持纯净度高于 0.95 且标准差小于 0.02，轮廓系数稳定在 0.5 到 0.6 区间内。这一优异表现得益于双重鲁棒机制——基于最近邻域的局部密度估计有效地过滤随机噪声，不均匀衰减策略则有效缓解噪声积累对簇演化的影响。

本章统计四种方法在 CSD1、CSD2 和 CSD3 数据集上平均纯净度和平均轮廓系数如表 3-1 表 3-2 所示。

表3-1　人工数据集上的纯净度

Table3-1 Purity on artificial datasets

数据集/方法	RDDPCStream	Denstream	DBSTREAM	Clustream
CSD1	**0.996**	0.956	0.944	0.976
CSD2	**0.976**	0.780	0.960	0.936
CSD3	**0.972**	0.671	0.943	0.936

表3-2　人工数据集上的轮廓系数

Table3-2 Silhouette coefficients on artificial datasets

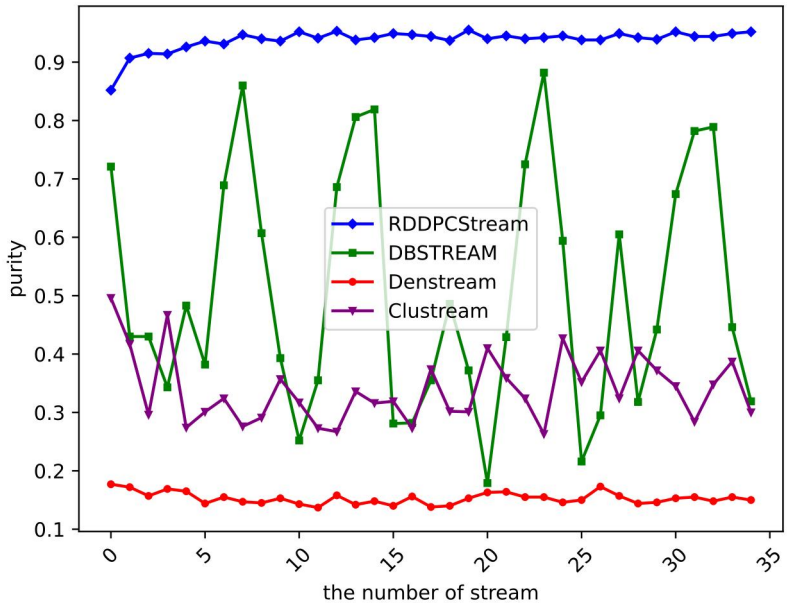
数据集/方法	RDDPCStream	Denstream	DBSTREAM	Clustream
CSD1	0.551	**0.752**	0.359	0.341
CSD2	**0.539**	**0.539**	0.414	0.311
CSD3	**0.531**	0.419	0.315	0.354

结合表 3-1、表 3-2 以及前述的聚类性能折线图分析，可以得出结论：RDDPCStream 方法的抗噪声能力和聚类性能在四个数据流聚类方法中最佳。在

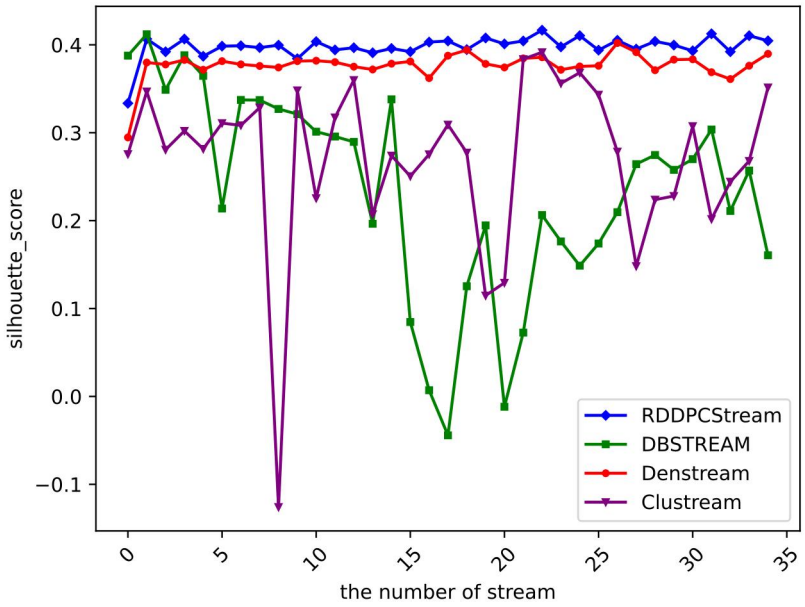
无噪声干扰的情况下，Denstream 方法在人工合成数据集上的聚类性能表现最为出色，但当数据集受到噪声影响时，聚类性能显著下降，表明该方法对噪声较为敏感；DBSTREAM 方法在人工合成数据集上展现较强的抗噪声能力，但聚类性能相较于 RDDPCStream 方法略显不足，尤其是在面对更复杂的数据流环境时；Clustream 方法在人工合成数据集上的抗噪声能力仅优于 Denstream 方法，但在聚类性能上未能超越 RDDPCStream 方法，表明该方法在处理高噪声场景下的能力有所局限；相比之下，RDDPCStream 方法在人工合成数据集上不仅展现出卓越的抗噪声特性，且在聚类性能上也表现最优。综上所述，可以证实本章提出的 RDDPCStream 方法在人工合成数据集上的有效性和优越性。

3.3.3 真实数据集对比实验

RDDPCStream 方法和对比方法在 Covtype 系列数据集上实时纯净度和轮廓系数值，如图 3-6、图 3-7 和图 3-8 所示。



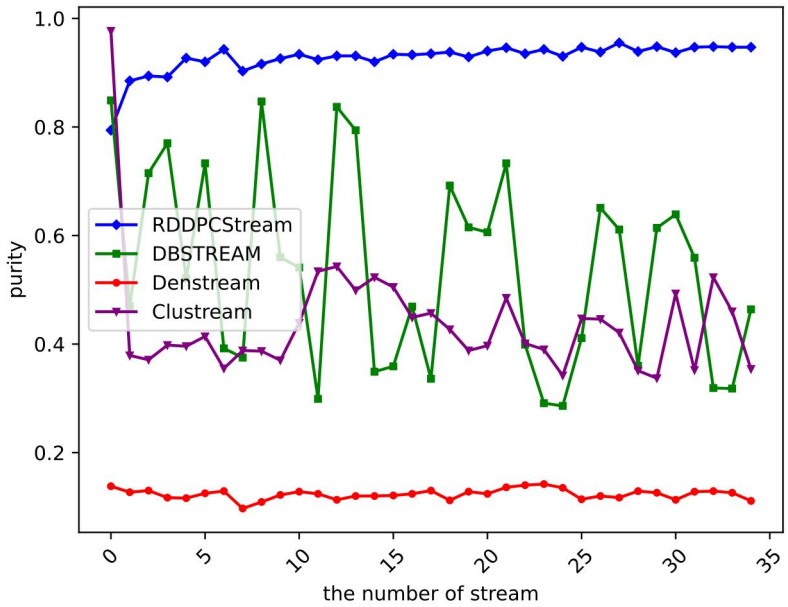
(a)Covtype 纯净度



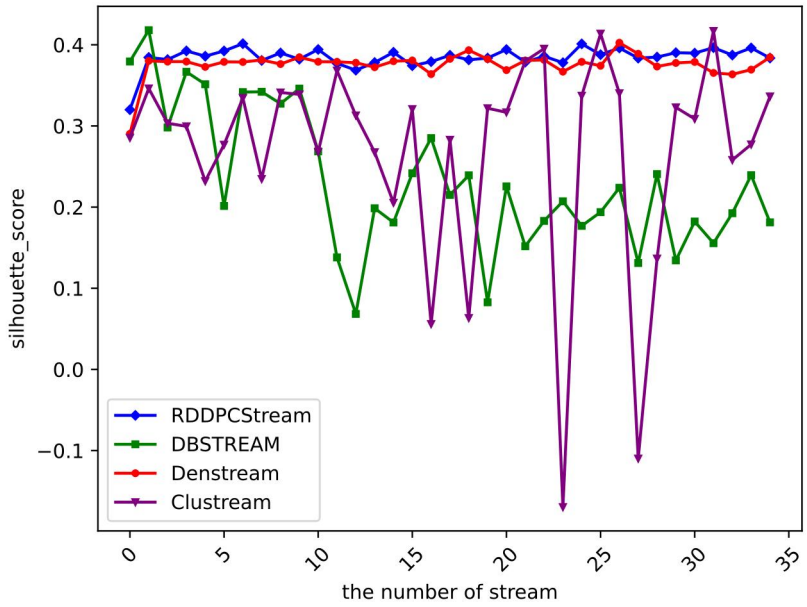
(b)Covtype 轮廓系数

图3-6 Covtype上纯净度和轮廓系数对比图

Fig.3-6 Comparison of purity and Silhouette coefficients on the Covtype



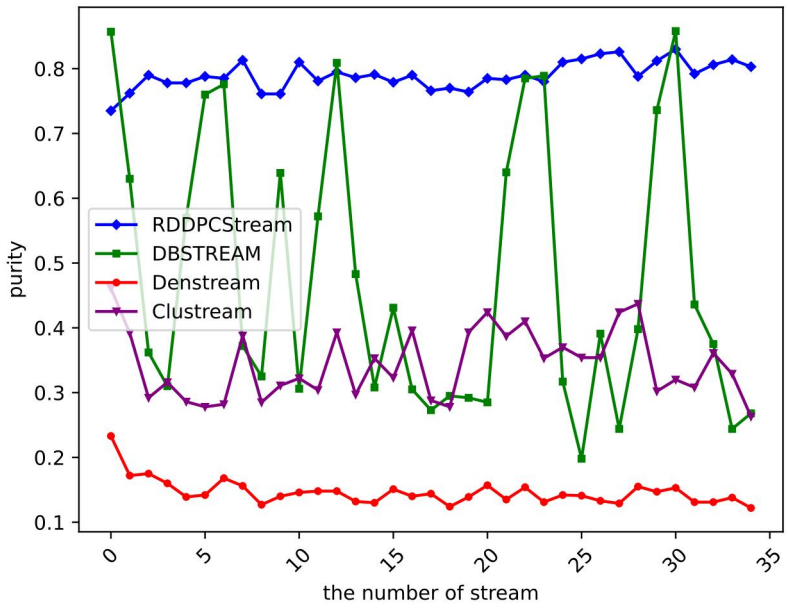
(a)Covtype-N1 纯净度



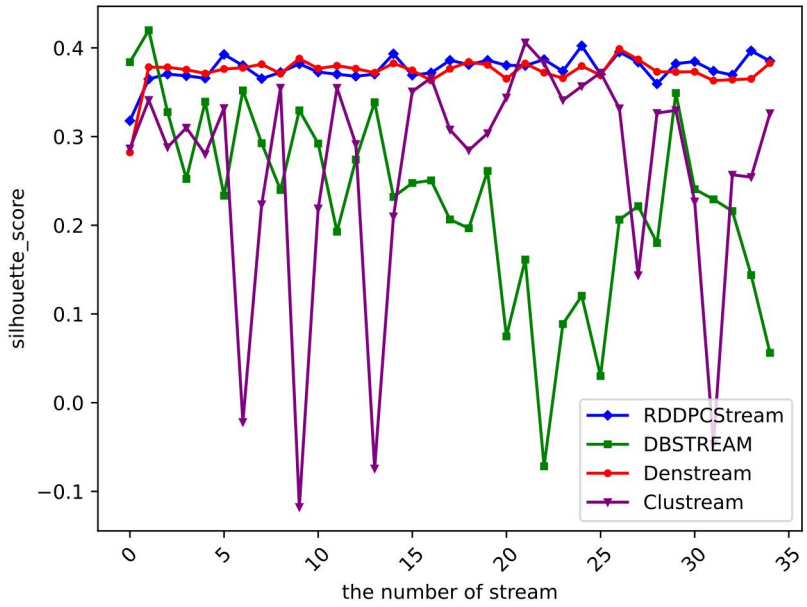
(b)Covtype-N1 轮廓系数

图3-7　Covtype-N1上纯净度和轮廓系数对比图

Fig.3-7 Comparison of purity and Silhouette coefficients on the Covtype-N1



(a)Covtype-N2 纯净度



(b)Covtype-N2 轮廓系数

图3-8 Covtype-N2 噪声纯净度和轮廓系数对比图

Fig.3-8 Comparison of purity and Silhouette coefficients on the Covtype-N2

根据图 3-6、图 3-7、图 3-8 给出的信息，所有方法在 Covtype 系列数据集上的纯净度和轮廓系数均呈现不同程度的下降，但 RDDPCStream 方法的下降幅度最小稳定性最佳。

在无噪声环境下，如图 3-8 所示，各方法的性能差异显著。Denstream 方法表现最为欠佳，纯净度指标普遍较低，虽然轮廓系数保持稳定，但始终低于 0.35，说明该方法在无噪声条件下的聚类效果较差。DBSTREAM 方法表现出较大的性能波动，纯净度最大波动幅度超过 0.6，轮廓系数波动幅度较大，显示出方法在真实数据集上的不稳定性。Clustream 方法的纯净度指标普遍低于 0.5，且轮廓系数不仅低于 0.4 还表现出明显的不稳定性，表明该方法在数据集上的聚类效果不足。相比之下 RDDPCStream 方法展现出优异的聚类性能，纯净度指标始终稳定在 0.8 以上且轮廓系数波动幅度最小，表明该方法在无噪声环境下具有显著的性能优势。

在 Covtype-N1 数据集和 Covtype-N2 数据集，如图 3-7 和图 3-8 所示，所有方法的聚类性能在 Covtype 数据集的基础上均受到不同程度的影响。Denstream 方法在噪声环境下的纯净度仍低于 0.2，但该方法的轮廓系数能够稳定保持在

0.35 以上，这一表现优于 Clustream 和 DBSTREAM 方法。Clustream 方法在 Covtype-N1 数据集的第一组数据纯净度超过 0.9，但随着数据量的增加，性能急剧下降，后续若干组数据的纯净度均低于 0.6。在 Covtype-N2 数据集上，Clustream 方法的纯净度除第一组数据超过 0.5 外，其余数据组均低于 0.5，且轮廓系数波动显著，最低值低于 0.1。值得注意的是，RDDPCStream 方法在噪声数据集上的表现，虽较原始数据集有所下降，但下降幅度最小且最为稳定。在 Covtype-N1 和 Covtype-N2 数据集上，RDDPCStream 方法能保持较高的纯净度和稳定的轮廓系数，展现出优越的鲁棒性和聚类性能。

表 3-3、表 3-4 给出 RDDPCStream 方法和对比方法在 Covtype、Covtype-N1 和 Covtype-N2 数据集上的平均纯净度和平均轮廓系数。

表3-3　Covtype系列数据集上的纯净度

Table3-3 Purity on the Covtype family dataset

数据集/方法	RDDPCStream	Denstream	DBSTREAM	Clustream
Covtype	**0.937**	0.152	0.506	0.339
Covtype-N1	**0.927**	0.123	0.536	0.439
Covtype-N2	**0.789**	0.146	0.475	0.343

表3-4　Covtype系列数据集上的轮廓系数

Table3-4 Silhouette coefficients on the Covtype family dataset

数据集/方法	RDDPCStream	Denstream	DBSTREAM	Clustream
Covtype	**0.397**	0.376	0.234	0.271
Covtype-N1	**0.384**	0.375	0.231	0.269
Covtype-N2	**0.367**	0.373	0.225	0.264

结合表 3-3、表 3-4 和上述实时聚类信息，RDDPCStream 方法的聚类性能、稳定性和抗噪声能力在四个数据流聚类方法中最佳。Denstream 方法在 Covtype 系列数据集上的聚类性能最差；Clustream 方法的聚类稳定性差且纯净度和轮廓系数低；DBSTREAM 方法的平均纯净度仅次于 RDDPCStream 方法，但组间波动大对噪声抵抗能力较弱；RDDPCStream 方法在无噪声和含噪声情况下聚类性能均高于对比方法。综上所述，RDDPCStream 方法不仅在无噪声环境下表现出优异的聚类性能，在噪声环境下也展现出良好的聚类性能和鲁棒性。

3.4 本章小结

本章设计一种基于密度峰值的数据流动态聚类方法，解决流环境中识别数据

形状不确定性和噪声等因素影响聚类性能的问题。该方法包含在线阶段和离线阶段。在线阶段，提出微簇的不均匀衰减策略和数据的动态加权策略处理样本。当一个新的样本被合并到相应的微簇时，通过样本和微簇中心距离关系动态加权，有助于更准确地描述微簇，降低噪声干扰。合并后，计算微簇的权重根据权重调整对应的衰减因子，通过不均匀衰减策略能够有效保留重要的历史数据，删除噪声，经实验证明该方法在噪声环境下具备较好的鲁棒性。在离线阶段，为处理任意形状的聚类并快速找到聚类中心，提出基于最近邻域的局部密度计算方法。该方法能够自适应地计算局部密度，减少参数输入，通过近邻关系突出簇间差异性以此降低 DPC 算法分配阶段“多米诺效应”的影响，提高流聚类模型的聚类质量。

第 4 章　基于密度峰值聚类和微簇结构的数据流异常检测方法

本文第三章工作针对数据流环境存在的噪声、数据形状不确定性两个因素影响模型聚类问题，提出针对性的解决方法。但仍存在些许不足之处，如数据流环境不仅存在噪声和数据形状不确定性，还存在异常数据和概念漂移等情景。在实际应用中，上述的情景往往同时发生，影响数据流聚类方法的聚类性能。因此，本章在第三章的工作基础上，改进聚类模型适应更加复杂的数据流场景，提高聚类性能。

在数据流聚类模型的应用过程中，异常和概念漂移等因素对流聚类模型的聚类性能产生不利影响，导致聚类效果不佳。然而，异常与噪声有所不同，重要性通常大相径庭，尽管数量较少，但往往包含着特殊且有价值的信息。因此，如何在数据流聚类过程中有效检测异常，提升聚类模型的性能，成为一个亟待解决的问题。为解决这一问题，本章提出一种基于密度峰值聚类与微簇结构的数据流异常检测方法，旨在通过对聚类模型设计异常检测机制，提高聚类模型在含异常的动态数据流中聚类的性能。该方法由微簇学习模块和异常检测模块构成，微簇学习模块将数据映射至微簇；异常检测模块则将微簇作为分析单元，通过二次聚类对微簇划分区域检测异常。

4.1 引言

在数据流挖掘中，异常检测旨在识别并过滤出虚假、有害和包含特殊信息的数据，提高数据流挖掘模型的挖掘质量。鉴于数据流中常有标签缺失问题，多数异常检测方法采用半监督或无监督学习策略[81-82]。但这些方法及算法的计算量大，导致延迟高，因此不适用于大规模数据。聚类算法不依赖标签且更易发现未知类型样本，研究者们提出基于聚类的异常检测方法。这类方法通常认为不属于任何簇的数据点、距离聚类中心较远的数据点，以及数据点稀疏的簇是异常[83]。大多数这类方法采用两阶段框架，其中离线阶段的算法计算量大耗时高，难以满足数据流处理的时间要求。为解决这一问题，在线框架的异常检测算法应运而生。尽

管，这类方法虽然能够满足即时数据处理的需求，但随着数据流的概念漂移的出现。初始预测模型无法准确预测新的数据分布，不能有效检测出异常数据，导致数据流挖掘模型的挖掘性能受到影响。

为此本章提出一种基于密度峰值聚类和微簇结构的数据流异常检测方法，命名为 DADS（Density peak clustering and micro-cluster structure-based anomaly d0etection algorithm for data stream）方法。该方法采用在线框架由两个主要模块（微簇学习模块和异常检测模块）构成。微簇学习模块以微簇结构为基础，将数据点分配到微簇中，并通过时间衰减策略保留微簇的有效历史数据。该模块定义微簇稀疏度，微簇稀疏度有助于捕捉数据流中的异常。异常检测模块采用 K 近邻思想[84]改进局部密度定义方法，对所有微簇进行区域划分，并设计一种评估局部和整体离群信息的异常评分计算方法，筛选出异常。DADS 方法通过微簇结构优化计算复杂度，仅在微簇删除或新增时才需重新划分区域，显著减少检测时间。此外，该模型能够通过无监督聚类方法，自适应数据流中的概念漂移，能有效检测异常，提高模型聚类质量。

4.2 基于密度峰值聚类和微簇结构的数据流异常检测方法

4.2.1 框架结构

本章提出的 DADS 方法侧重于解决以下两个问题：（1）实时有效地检测数据流中的异常；（2）降低数据流概念漂移对模型聚类的影响。该方法的整体框架如图 4-1 所示，DADS 方法采用在线框架处理数据，分为两个模块：微簇学习模块和异常检测模块。微簇学习模块将流中数据映射到微簇中，在此模块中定义了微簇稀疏度 κ 和自适应权重阈值 μ，稀疏度从内部反映微簇的局部异常程度，自适应权重阈值减少参数输入，可以自适应判断哪些微簇需要删除。异常检测模块对微簇进行基于 K 近邻思想改进局部密度定义方法，按照这种定义方法重新设计密度峰值聚类划分区域，并提出一种新的异常评分计算方法，在区域内计算微簇的异常评分，根据微簇的异常评分筛选出异常微簇和正常微簇。

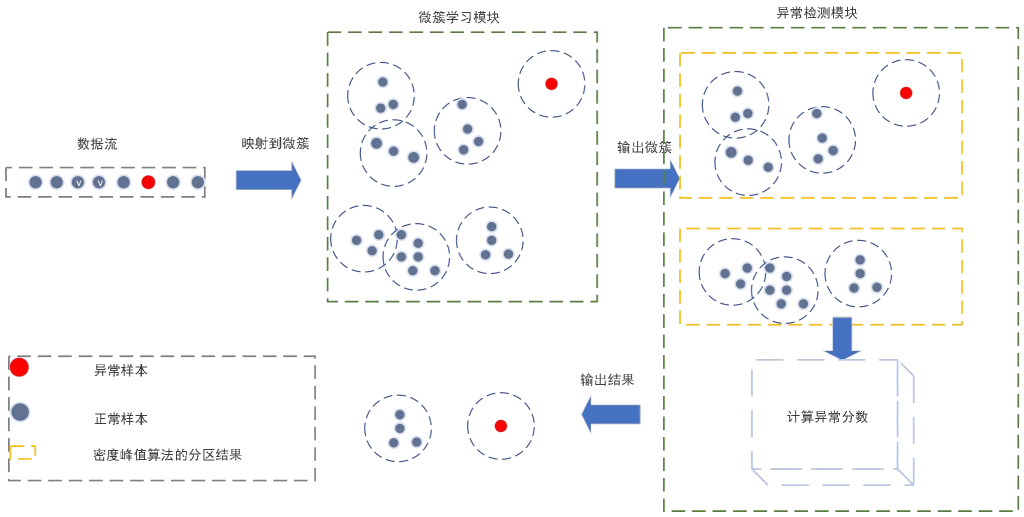


图 4-1 DADS 方法框架图

Fig.4-1 Framework of the DADS Algorithm

4.2.2 微簇学习模块

在数据流聚类挖掘模型中检测异常时，检测效率将大幅影响聚类的性能。多数基于聚类的异常检测方法将单个样本作为处理单元。这种方法在小规模数据集上能够保持较高的准确性，但在数据流场景中由于数据量庞大且传输速度快，这类方法往往导致检测时间增加，从而影响检测的实时性。为此 DADS 方法使用微簇结构，改进保存的实例信息，每个微簇以多元组形式存储样本，多元组保存的信息包括：数量 N 、权重 ω 、最近一次更新时间 t_p 、簇心 c 、更新周期 T 和稀疏度 κ 等。对于每个微簇 DADS 方法设置插入阈值 a 和核心插入阈值 $\frac{a}{2}$，将微簇内部分成核心区域和边界区域。当新样本被处理时，首先与现有微簇进行匹配，如果样本匹配到多个微簇，则选择距离最优的微簇进行合并；如果未能匹配到任何微簇，则为该样本创建一个新的微簇。本章设计微簇时间衰减函数 $f(\bullet)$，设计自适应微簇权重阈值 μ，微簇权重低于 μ 时删除微簇，保留数据流中有价值的历史信息，时间衰减函数和微簇权重的计算方法见式（4-1）和式（4-2）。

$$f(t-t_p)=e^{-(t-t_p)\times\lambda} \tag{4-1}$$

$$\mu=\frac{1}{|C|}\sum_{i\in C}\omega_i \bullet f(T) \tag{4-2}$$

式（4-1）中的 t 为当前时间，t_p 为样本上一次更新时间，λ 表示时间衰减因

子。式（4-2）中 T 为更新周期，即每过 T 时间对微簇集体更新状态，$|C|$ 为微簇总数量，C 为微簇集合 DADS 方法对刚加入的微簇的样本，权重有两种赋值方式如图 4-2 所示，当样本处于微簇的核心区域时微簇的权重更新方法如式（4-3）所示，当样本处于微簇的边界区域时微簇的权重更新方式如式（4-4）所示。

$$\omega_{i\text{core}} = N_{i\,core} \times f(t-t_p)+1 \tag{4-3}$$

$$\omega_{i\,bor} = \sum_{i\in N_{ibor}} e^{(-\frac{dis(i,c_i)}{1+dis(i,c_i)})} \times f(t-t_p) + e^{(-\frac{dis(i,c_i)}{1+dis(i,c_i)})} \tag{4-4}$$

$$\omega_i = \omega_{i\,core} + \omega_{i\,bor} \tag{4-5}$$

$$\omega_{\frac{a}{2}} = \sum_{i\in N_{ibor}} e^{\frac{(-\frac{a}{2})}{1+\frac{a}{2}}} \times f(t-t_p) \tag{4-6}$$

在式（4-5）中单个微簇的权重 ω_i，由核心区域权重 ω_{icore} 和边界区域权重 ω_{ibor} 两个部分组成，N_{icore} 和 N_{ibor} 分别表示核心区域与边界区域样本总数，以 $\frac{a}{2}$ 为界限，$dis(\bullet)$ 表示距离函数，采用欧式距离为计算标准，a 表示半径阈值。当处理新的样本时，判断样本到微簇中心的距离，小于核心插入阈值 $\frac{a}{2}$，插入核心区域，反之则插入边界区域。

式（4-6）定义微簇在边界区域的理想权值 $\omega_{\frac{a}{2}}$ 表示，为能及时获得微簇内的离散信息，本章通过核心区域和边界区域权重赋值不均匀关系，提出一种局部离散权重即微簇稀疏度 κ_i。当稀疏度越高时表示微簇内部样本更少更分散，稀疏度越低表示微簇内部样本越紧凑，稀疏度的取值范围[0,1]单个微簇稀疏 κ_i 的计算方法，如式（4-7）所示。

$$\kappa_i = \begin{cases} 1-\frac{1}{2}(\frac{\omega_{icore}}{N_{icore}\times f(t-t_p)} + \frac{\omega_{ibor}+1}{\omega_{\frac{a}{2}}+1}) & \omega_i > \mu \\ 1 & \omega_i \le \mu \end{cases} \tag{4-7}$$

当微簇权重大于权重阈值 μ 时，计算核心区域样本数量占整体数量的比例和计算边界区域样本与边界区域理想权值比值，因两者上限均为 1，以做差方式计

算微簇内部稀疏度。微簇内的样本都集中在核心区域则微簇的稀疏度为 0，当微簇权重低于 μ 时则认为微簇稀疏度为 1。

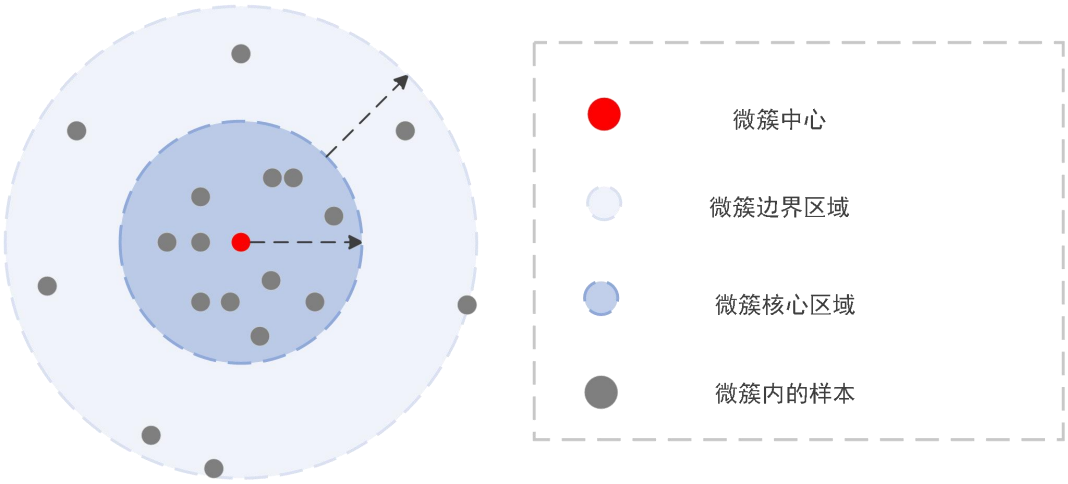


图 4-2 微簇内部分布图

Fig.4-2 Internal Distribution Diagram of Microclusters

算法 1 给出微簇更新模块的详细步骤。在此阶段模型接受数据流中样本并更新到微簇中，满足更新周期后更新微簇信息，删除权重低于阈值的微簇，输出更新后的微簇。

当数据流数据分布发生变化时，微簇也会跟着变化，微簇会随着时间的推移不断调整微簇中心在数据空间中的位置，同时微簇的稀疏度 κ 也会发生变化。因此，检测模型需要及时更新，确保始终基于最新的数据信息进行处理。这一动态更新机制，使得模型能够更准确地反映数据流的实时变化。为获得更准确的微簇内部离散信息，每当微簇的状态发生变化时，必须同步更新微簇稀疏度 κ。稀疏度 κ 反映微簇内样本分布的密度变化。实时更新稀疏度 κ，有助于更准确地捕捉微簇的结构变化，提升异常检测的效果。

算法 1 微簇学习模块

输入：样本 x_i，插入阈值 a，微簇更新周期 T

输出：微簇列表 C

1. C = 空列表
2. **for each** $x_i \in X$ **do**
3. **if** $c_i \in C$ *and* $dis(c_i, x_i) < a$ **then**
4. 将样本 x_i 加入微簇 c_i；
5. 更新微簇 c_i 的权重 ω_i；
6. 更新微簇 c_i 的稀疏度 κ_i；
7. **else**
8. 创建新的微簇 c_j；
9. 将样本 x_i 加入微簇 c_j 将 c_j 加入微簇列表 C；
10. **end if**
11. **end for**
12. **if** $t\%t_p == 0$ **then**
13. **for each** $c_i \in C$ **do**
14. 更新微簇 c_i 的权重 ω_i 和稀疏度 κ_i；
15. **endfor**
16. **for each** $c_i \in C$ **do**
17. **if** $\omega_i < \mu$ **then**
18. 从 C 中删除微簇 c_i；
19. **end if**
20. **end for**
21. 输出 C

此外，对长时间未更新的微簇进行清理，这些不活跃的微簇对当前数据点的判别影响不大。每个微簇是独立的，一个微簇的更新不会影响其他的微簇。更新频率高的微簇，保留的信息更有价值，更新频率低的微簇，更容易被删除。微簇跟随数据变化而变化，从而降低传统分类方法需要预训练的滞后性，并且该方法能处理概念漂移场景。

4.2.3 异常检测模块

微簇的稀疏度 κ 能够反映微簇中心半径 a 圆形区域内的离散程度，当一个微簇周边存在多个微簇时，仅通过 κ 不能直接判定哪个微簇是异常。因此，为准确识别全局的异常数据，需进一步设计异常检测机制。本章设计异常检测模块，该模块采用密度峰值聚类方法将微簇划分为若干区域，并通过分析微簇与中心位置的关系，筛选出全局异常微簇。通过此方法，能够有效识别数据流中具有显著异常特征的微簇，提高整体异常检测的准确性。

在异常检测模块中，相邻的微簇是否属于同一分区，由自身局部密度和微簇中心间的相对距离决定的。在 DADS 方法中微簇的局部密度，指微簇的自身权重加上微簇最近的 K 个近邻的部分权值组成。在数据分布较为均匀情况下，局部密度越高，表示该微簇和邻域内的微簇越紧密。若在微簇分布不均匀情况下，局部密度越高表示该微簇可能是邻域内的中心微簇，或是潜在的聚类中心。反之若局部密度较低则表示微簇在数据空间中处于边缘位置，且微簇内样本较少。基于这一理论，本章在原有的局部密度计算方法上加入 K 近邻信息，代替原有的核密度估计方法，微簇 c_i 的局部密度 ρ_i 计算方法，如式（4-8）所示。

$$\rho_i = \omega_i + \sum_{j \in KNN} \frac{\omega_j}{dis(c_i, c_j)} \tag{4-8}$$

其中 KNN 表示 K 近邻域，微簇 i 的局部密度为邻域微簇权值和距离之比的求和，再加上自身权重。但仅确定微簇的局部密度，仍不能决定两个微簇是否应该合并为同一类簇。因此，在密度峰值聚类方法中，聚类中心的局部密度应高于其他区域。每个聚类中心相距较远，且局部密度不是最高的样本应服从于最近的、局部密度高于它的样本，并将两者间的距离定义为相对距离。将这点应用在微簇空间中，非局部密度最大的微簇应分配给最近的且局部密度更高的微簇，并按照式（2-10）计算两者的相对距离。

当前微簇是局部密度最大值微簇时，相对距离为到数据空间中最远微簇的距离，按照式（2-10）计算每个微簇的相对距离，通过微簇的局部密度与相对距离按照式（2-11）计算单个微簇的决策值 λ_i。

按决策值分布规律，选择决策值较高的微簇作为分区中心，以此为依据筛选出分区中的全局异常微簇。为实现这一目标，本章提出一种全局异常微簇筛选策略。计算微簇的向心距离即到所在分区中心的距离，同时计算分区内微簇的平均局部密度 ρ_{avg}，如式（4-9）所示，其中 $|C|$ 表示微簇的总个数。根据该筛选策略，异常微簇的数据占比应显著低于正常微簇，并且异常微簇应远离正常的微簇群。分区内部微簇的平均局部密度应当高于异常微簇的局部密度。本章定义密度系数单个微簇的密度系数 η_i 计算方法如式（4-10）所示，密度系数表示微簇局部密度和平均局部密度的比例关系。

$$\rho_{avg}=\frac{1}{|C|}\sum_{i=0}^{|C|}\rho_i \tag{4-9}$$

$$\eta_i=\frac{\rho_i}{\rho_{avg}} \tag{4-10}$$

异常评估方法是依据微簇的向心距离，密度系数和微簇自身的稀疏度。向心距离越高，密度系数越低，稀疏度越高则微簇的异常评分就越高。单个微簇的异常分数计算方法如式（4-11）所示，其中 P_i 表示当前微簇所在分区中局部密度最高微簇的中心位置。

$$Ascore_i=\frac{dis(P_i,c_i)}{\eta_i}\times\kappa_i \tag{4-11}$$

为更清晰表达异常检测模块的功能，算法 2 详细解析异常检测方法的整体步骤伪代码如下。在此阶段调用算法 1 获取最新的微簇信息，计算局部密度与相对距离，计算微簇的决策值，对微簇二次聚类划分区域，计算微簇异常评分，筛选出当前的异常微簇。

算法 2 DADS

输入：样本 x_i、插入阈值 a、微簇更新周期 T、近邻参数 K、分区个数 D

输出：异常微簇

1. $C=$ 微簇列表；
2. 调用算法 1；
3. **for each** $c_i \in C$ **do**
4. 计算 c_i 的局部密度 ρ_i；
5. 更新 c_i 的局部密度加入微簇列表 C；
6. **for each** $c_i \in C$ **do**
7. 计算 c_i 的相对距离 δ_i；
8. 更新 c_i 的相对距离 δ_i 加入微簇列表 C；
9. **for each** $c_i \in C$ **do**
10. 计算 c_i 的决策值；
11. 更新 c_i 的决策值到微簇列表 C；
12. 选择决策值前 D 个的微簇作为分区中心 p_i 更新到分区列表 P；
13. **for each** $p_i \in P$ do
14. **for each** $c_i \in p_i$ do
15. 计算 c_i 到 p_i 的向心距离；

16. 更新 c_i 的向心距离保存至微簇列表 C；
17. **end for**
18. **end for**
19. **for each** $p_i \in P$ **do**
20. 计算分区 p_i 的平均局部密度；
21. 更新分区 p_i 的平均局部密度到微簇列表 C；
22. **end for**
23. **for** $c_i \in C$ **do**
24. 计算微簇 c_i 的密度系数 η_i；
25. 计算微簇 c_i 的异常评分 $Ascore_i$；
26. **end for**
27. 输出异常分数最高的微簇；

4.3 实验结果与分析

4.3.1 实验设置

本章所有实验均在 PyCharm 开发环境中进行。实验分为两个部分：人工数据集上的参数敏感性分析实验、人工数据集和真实数据集上的对比分析试验。本次实验的人工数据集，均由 MOA[85]生成，通过 RBF 和 Hyperplane 生成器生成两种类型的数据集分别命名为 RBF1、RBF2、HP1、HP2，其中 RBF2 和 HP2 数据集加入了不同尺度的概念漂移。

RBF1 通过 RBF 生成器，生成 9500 条正常数据和 500 条异常数据，异常比例为 5%，每条数据包括两个特征，共计一万条数据不含概念漂移。RBF2 数据集在此基础上加入概念漂移。

HP1 通过 Hyperplane 生成器，生成 9500 条正常数据和 500 条异常数据，每个数据包括两个特征。Hyperplane 生成器通过属性控制概念漂移尺度，HP1 为无概念漂移数据集。HP2 数据集在此基础上加入单属性概念漂移。

因第三章选用的 Covtype 数据集包含七个类别表示多个森林覆盖区域适合聚类使用，差异性上不如 Shuttle 数据集，在异常检测时对模型性能影响较小。故本次实验的真实数据集为 Shuttle 数据集。Shuttle 数据集：是一个由 NASA 提供的包含 58000 个样本的机器学习数据集，每个样本具有 9 个特征，涵盖飞机飞行过程中的正常飞行数据和传感器读数，旨在用于分类任务，尤其是飞行状态的识别。数据集被划分为 56000 个训练样本和 2000 个测试样本，类别标签有 7 种

反映不同的飞行状态，其中有 5 种类别被认为是异常。适用于研究分类方法、异常检测和故障诊断等领域，具有实际应用价值和类别不平衡的特点。在 Shuttle 数据集，随机抽取 10000 条数据作为实验数据集。其中控制正常与异常比例为 19:1，在每组中都插入一次异常数据，且每组只出现同一种异常。对实验数据集做出一系列初始化操作如：将数据集划分为 39 个组以每个组看成一个阶段，在每个阶段放入少量异常数据，模拟现实数据流；在每个组中模型在线学习数据，每组数据结束时计算当前阶段的异常精准度；除第一组包含 500 个样本，用于模型的初始化其余每组包含 250 个样本。

本次实验分为两个部分，参数实验和对比实验。在 DADS 方法中需要输入多个参数，如近邻参数 K，分区个数 c 和插入阈值 a 和微簇更新周期 T 。在这些参数中插入阈值 a 对模型的影响最明显，这是因为 a 的值越大模型生成的微簇就越少，a 越小生成的微簇就越少。a 的取值对其他参数都有不同程度的影响，比如 a 值的大小会影响，同一 K 值的精准性、方法的处理速度、a 值设置过大会导致大部分微簇的密度都很高，最极端的情况会导致一个分区中可能只存在一个微簇。因此，参数实验部分针对 a 的选取进行分析，验证模型的性能。

对比实验部分，列举 DADS 方法和 PYSAD[86]框架中的 KNNCAD，LOOP 等方法和 HST[87]方法，在 RBF1、RBF2、HP1、HP2 与 Shuttle 数据集进行对比分析。

KNNCAD 方法是基于 K 近邻的异常检测方法，核心思想是计算样本和其 K 近邻之间的欧式距离，使用这些距离来评估样本的异常程度，在 PYSAD 框架中，KNNCAD 主要用于流数据异常检测，它通过持续更新近邻信息来检测时间序列中的异常变化。

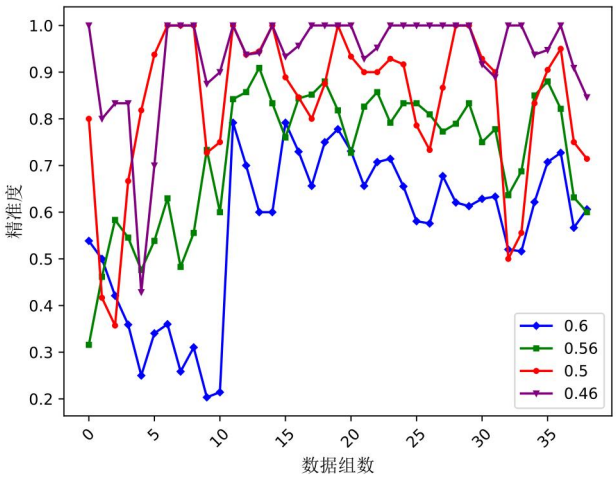
LOOP 方法是一种基于离群因子的异常检测方法，通过计算每个数据的局部密度与近邻点的密度进行比较，通过局部离群概率计算异常评分，使得评分在不同数据分布下更具稳定性。

HST 方法是一种无监督异常检测方法，属于随机投影树 (Random Projection Trees) 类别。其核心思想是：通过随机划分数据空间来构造一组半空间树，异常点在树的叶子节点中往往更靠近根部，计算数据点的平均路径长度，较短的路径长度表示异常程度较高。

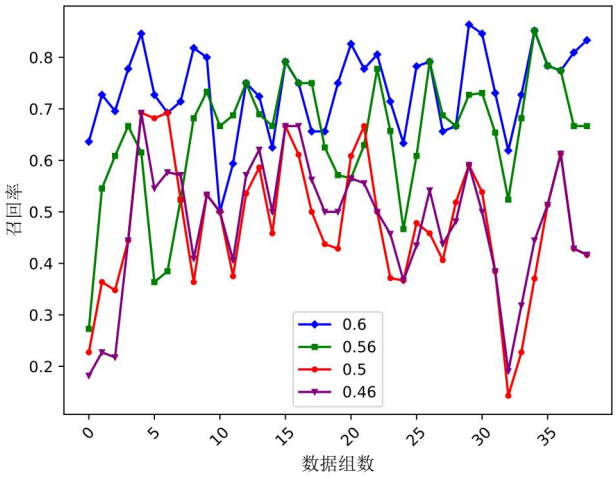
本章实验参数设置如下，参数实验：DADS 方法，近邻参数 K=5、分区个数 $D=2$、微簇更新周期 T=2 和插入阈值 $a \in [0.6,0.56,0.5,0.46]$。对比试验：KNNCAD，LOOP 等方法和 HST 方法，均按照文献设置参数，DADS 方法近邻参数 K=5、密度阈值 μ、分区个数 $D=2$、微簇更新周期 T=2 和插入阈值 $a=0.56$。

4.3.2 参数实验与分析

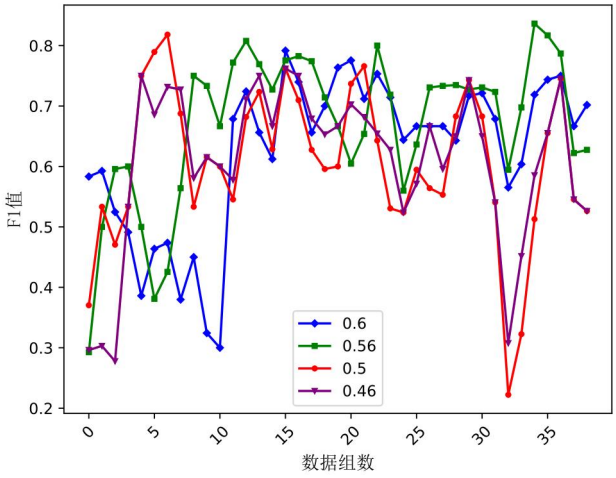
在本节实验中，针对微簇插入阈值 a，分析对模型整体性能的影响。通过在不同 a 值条件下，测试模型的方法性能，并采用三个常用指标：精准度、召回率和 *F1* 值进行分析。本节实验采用 RBF1 数据集和 RBF2 数据集，验证本方法在应对概念漂移情况下的参数有效性。



(a)精准度



(b)*Recall*

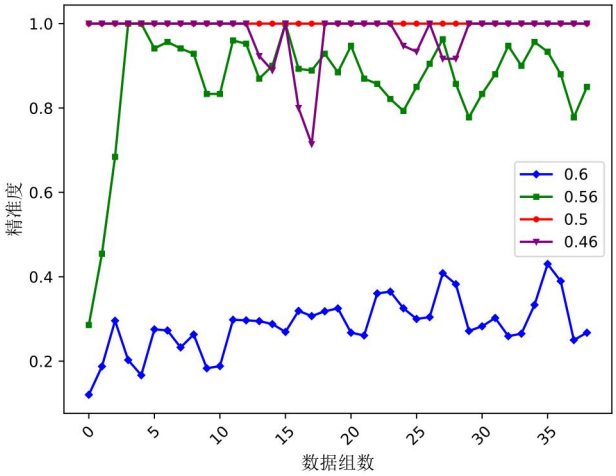


(c)*F1* 值

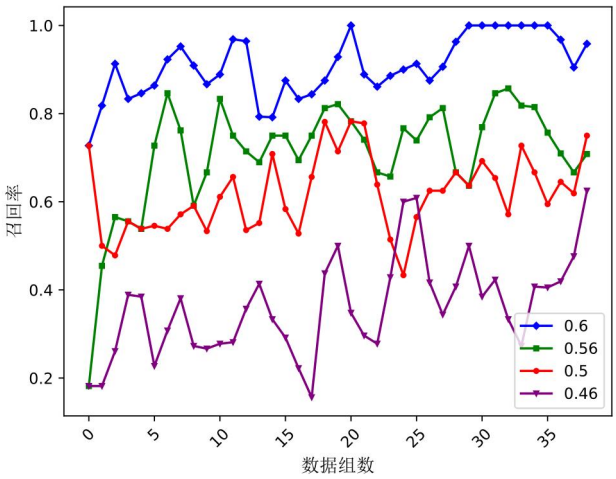
图 4-3 DADS 方法在 RBF1 数据集上参数实时折线图

Fig.4-3 Real-time Line Chart of DADS Algorithm Parameters on RBF1 Dataset

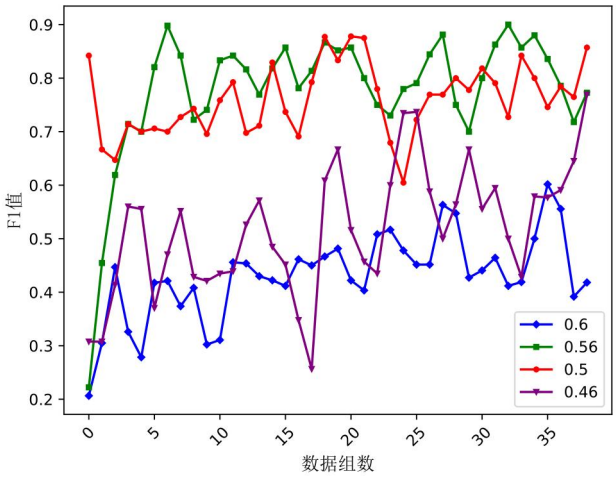
图 4-3 和图 4-4 给出方法在有概念漂移和无概念漂移情境下，不同插入阈值参数 *a*，DADS 方法性能的实时折线图，根据图中信息当 *a* 为 0.56 模型的综合性能最佳。图 4-3(a)显示，当参数值 *a* 为 0.6 和 0.56 时，DADS 方法的精准度降至最低，表明这时模型的聚类性能最低。该现象产生的主要原因，是随着微簇包含样本数量的增加，异常被错误地划分到正常样本的微簇中，导致聚类精准度显著降低。图 4-3(b)进一步表明，当参数值 *a* 为 0.5 和 0.46 时召回率表现最差。这是因为较低的参数值导致微簇内包含的样本数量减少，使得 DADS 方法倾向于将异常程度最高的微簇判定为异常微簇，从而导致召回率下降。当参数值 *a*=0.56 时，模型的召回率达到最高值，表明模型将大部分异常都成功检测，但较低的精准度表明模型在检测异常时存在误判。图 4-3(c)展示 *F1* 值的变化情况 *F1* 值反映模型当前检测的稳定性。从图 4-3(c)可以看出当参数值 *a* 为 0.5 和 0.56 时，检测方法的 *F1* 值最为稳定，表现最佳。这表明在这些参数设置下，方法能通过发现异常让模型的聚类性能达到最优。



(a) 精准度



(b)*Recall*



(c)*F1* 值

图 4-4 DADS 方法在 RBF2 数据集上的参数实时折线图

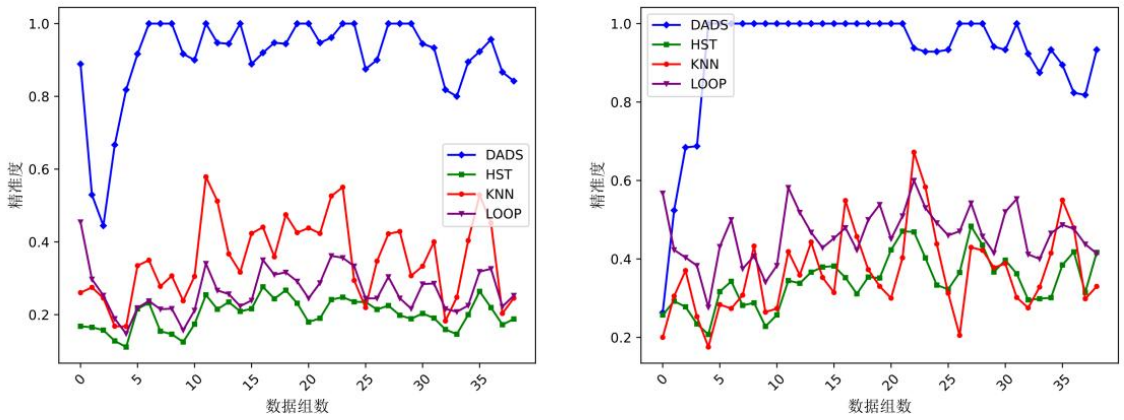
Fig4- 4 Real-time Line Chart of DADS Algorithm Parameters on RBF2 Dataset

图 4-4 展示 DADS 方法，在含有概念漂移的 RBF1 数据集上，实时异常检测结果。如图 4-4(a)和图 4-5(b)所示，当参数值 *a* 设置为 0.46 时，在检测异常时，模型受到概念漂移的影响稳定性较差。其中，召回率波动较大且整体偏低，但精准度较高，这表明模型通过检测出部分异常，大幅度提高模型聚类性能，召回率偏低说明，当前的插入阈值不能有效地检测出所有的异常。当参数值 *a* 设置为 0.6 时，模型的召回率显著提高，但精准度却大幅降低。这是由于微簇范围的扩大，使得模型将许多正常样本错误地识别为异常，说明模型没有在异常检测精准性和聚类性能之间找到平衡，过分追求检测异常降低模型原本的聚类性能。

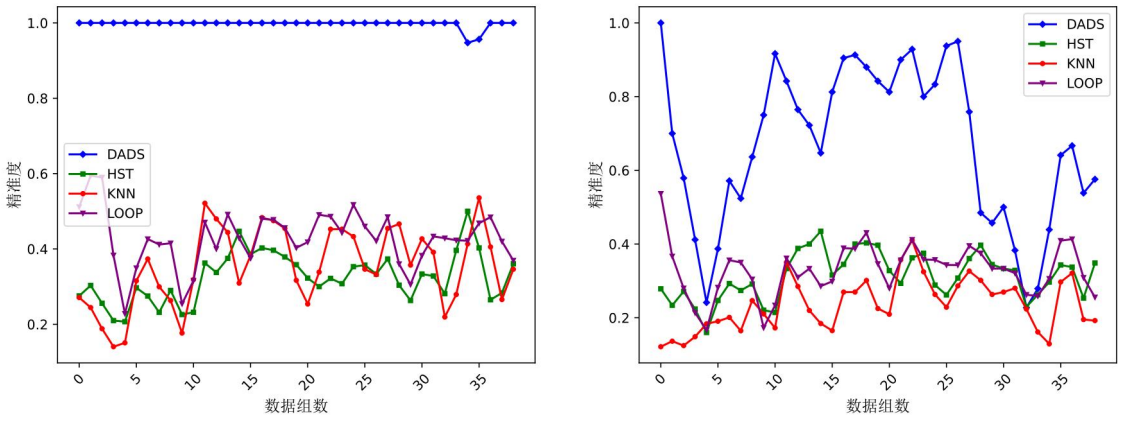
当参数值 *a* 降低至 0.56 时，方法能够学习更多的数据更好地适应概念漂移。此时，精准度和召回率的组间波动显著减小，模型能更准确地识别异常。如图 4-4(c)所示，当参数值 *a* 为 0.56 时，*F1* 值优于其他参数设置，并且与无概念漂移情况相比，精准度、召回率和 *F1* 值三个指标均略有提升。这一结果表明，DADS 方法能够通过调节微簇范围参数 *a*，让模型在检测异常正确性和聚类性能间达到一个平衡，使得模型通过发现异常来提高聚类性能。

4.3.3 对比实验与分析

本节实验旨在通过和多个异常检测方法在实验数据集上对比，验证本章提出的方法通过检测异常提高模型聚类性能的可行性。图 4-5 给出各个方法在所有人工数据集上的精准度，图 4-6 给出在所有人工数据集上的召回率，最后图 4-7 给出在所有人工数据集上的 *F1* 值。



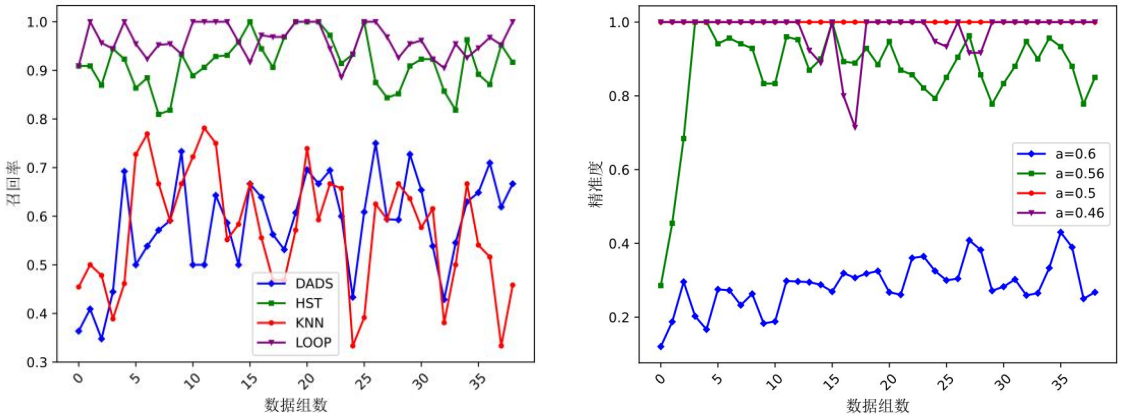
(a)RBF1　　(b)RBF2



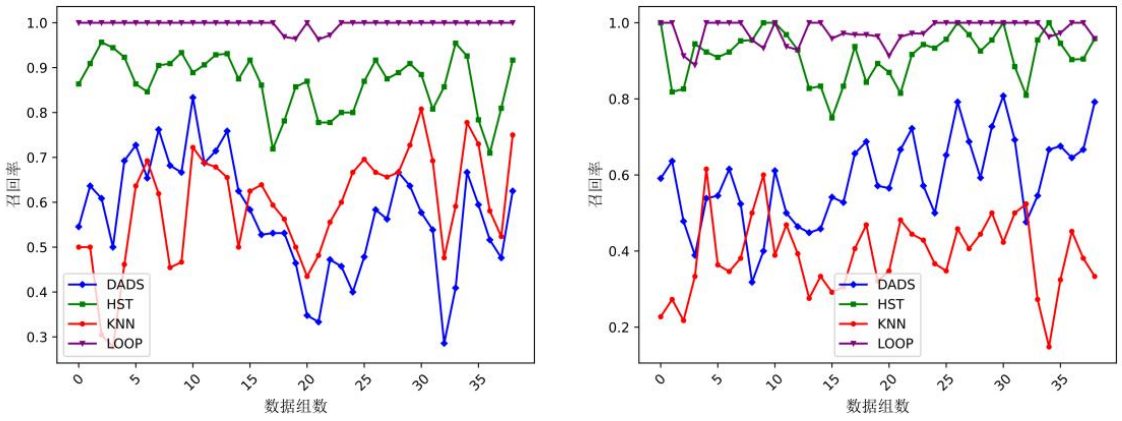
(c)HP1　　(d)HP2

图 4-5 四个方法在人工数据集上精准度的实时折线图

Fig4-6 Real-time Line Chart of Accuracy for Four Algorithms on the Synthetic Dataset



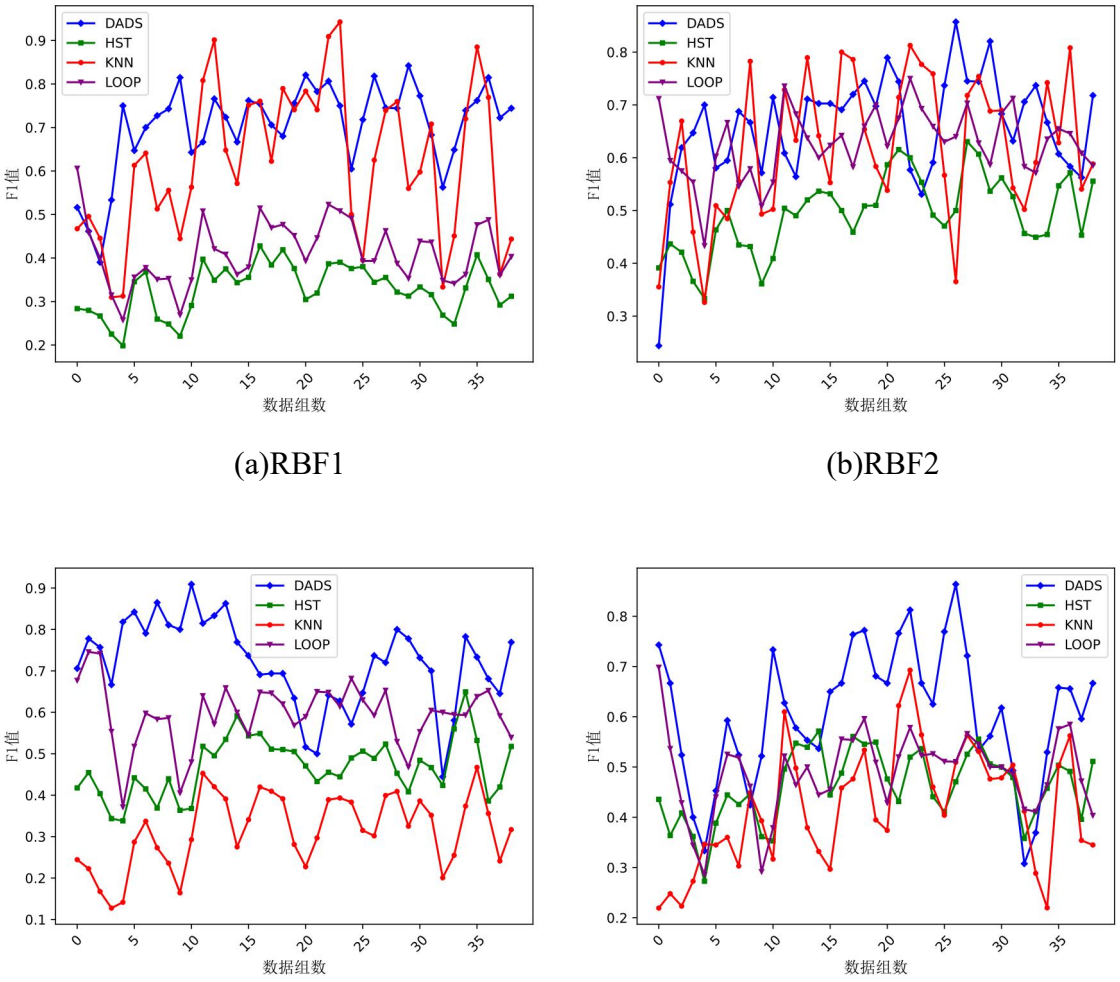
(a)RBF1　　(b)RBF2



(b)HP1　　(b)HP2

图 4-6 四个方法在人工数据集上召回率的实时折线图

Fig 4-6 Real-time Line Chart of Recall Rate for Four Algorithms on the Artificial Dataset



(a)RBF1 (b)RBF2

(c)HP1 (d)HP2

图 4-7 四个方法在人工数据集上 *F1* 值的实时折线图

Fig4-7 Real-time Line Chart of *F1* Score for Four Algorithms on the Artificial Dataset

图 4-5、图 4-6、图 4-7 中(a)、(b)、(c)、(d)分别表示 RBF1、RBF2、HP1 和 HP2 四个数据集上的对比实验信息，DADS 方法的精准度和 *F1* 值最佳，LOOP 方法的召回率最佳。通过图 4-6 的对比分析可以发现，DADS 方法在四个数据集上的精准度，均显著优于其他对比方法，表明模型及时检测异常提高聚类性能。其余对比方法在包含概念漂移的数据集（如 RBF2 和 HP2）上，精准度明显低于无概念漂移的数据集（如 RBF1 和 HP1），DADS 方法则展现出优异的鲁棒性，无论是在概念漂移数据集还是无概念漂移数据集，均能保持稳定的高精准度。虽

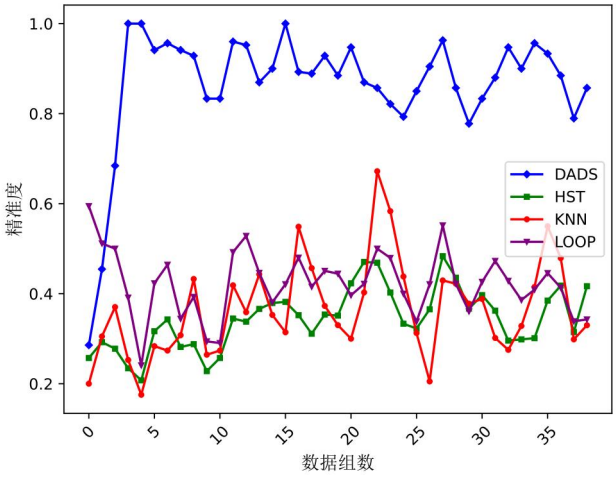
然，DADS 方法在 HP2 数据集上的精准度存在一定波动，但整体表现仍明显优于其他方法，这充分证明该方法在处理概念漂移问题时优秀的适应能力。

从图 4-6 展示的实时召回率来看，HST 和 LOOP 方法在四个数据集上的召回率值均处于领先位置。然而，通过深入分析可以发现，这种高召回率是牺牲模型挖掘的精准度为代价的，方法倾向于将部分正常样本误判为异常样本，这就大大降低模型对数据流的挖掘性能。这种检测策略虽然提高召回率，但会导致方法在实际应用中产生大量误报，不仅影响检测结果的可靠性还会造成计算资源的严重浪费。因此在评估方法性能时需要综合考虑多个评价指标，以避免单一指标的片面性。

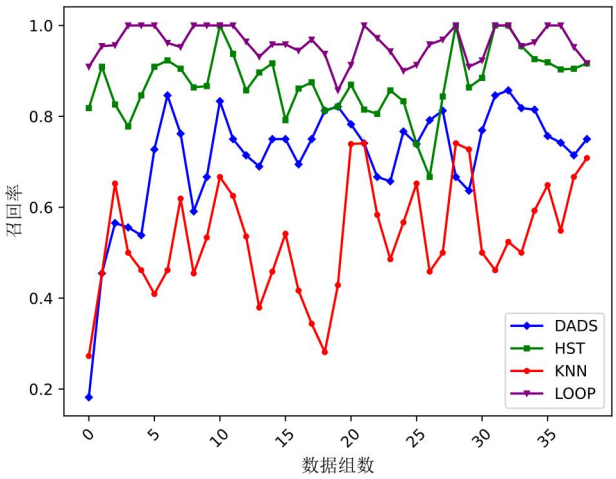
图 4-7 展示 RBF1、RBF2、HP1 和 HP2 四个数据集上的实时 *F1* 值，可以看出 DADS 方法在四个数据集上的 *F1* 值最佳且最稳定。具体如下：DADS 方法和 KNNCAD 方法在 RBF1 和 RBF2 数据集上均取得较好的 *F1* 值，其中 DADS 方法的稳定性明显优于 KNNCAD 方法。在更具挑战性的 HP1 和 HP2 数据集上，KNNCAD 方法的 *F1* 值出现显著下降，DADS 方法虽然也表现出一定程度的波动，但仍保持相对稳定的性能水平。LOOP 方法和 HST 方法在所有数据集上的 *F1* 值均明显落后于其他两个方法。KNNCAD 方法在四个数据集上的整体表现仅次于 DADS 方法。综上所述，DADS 方法虽然在处理复杂数据集时存在轻微的性能波动，但该方法综合性能仍然显著优于其他对比方法。

通过综合精准度、召回率和 *F1* 值三个关键指标的评价结果可以得出结论：DADS 方法在动态数据流异常检测任务，展现出显著的优势。该方法不仅能够有效处理概念漂移问题，还能通过检测异常提高模型聚类性能和聚类质量，这为数据流聚类挖掘容易受到异常影响，提供一个解决思路。

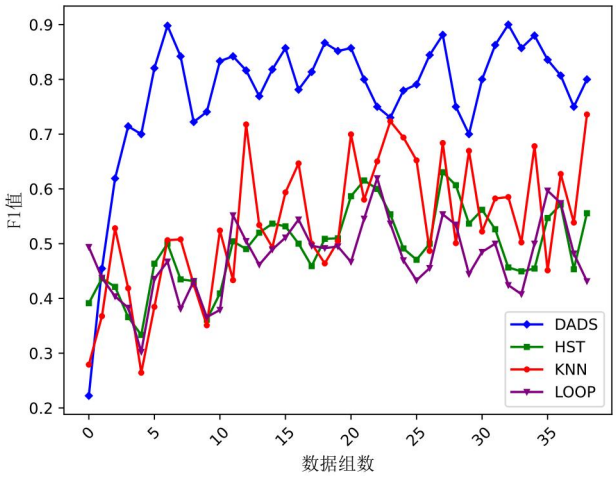
图 4-8 给出四个方法在 Shuttle 数据集上各项指标的实时折线图，其中(a)代表精准度、(b)代表召回率、(c)代表 *F1* 值，表 1 给出四个方法在人工数据集和真实数据集上的平均精准度、召回率、*F1* 值。



(a)精准度



(b)*Recall*



(c)*F1* 值

图 4-8 四个方法在 Shuttle 数据集上精准度、召回率、*F1* 值的实时折线图

Fig.4-8 Real-time Line Chart of Precision, Recall, and *F1* Score for Four Algorithms on the Shuttle Dataset

依据图 4-8 的信息，DADS 方法在 Shuttle 数据集上实时检测和挖掘性能最佳。具体表现如下：HST 方法的精准度和 *F1* 分数均表现不如其他方法，特别是与召回率表现优异的 LOOP 方法相比；LOOP 方法在稳定性和性能方面均优于 HST 方法，与 KNNCAD 方法持平，该方法召回率在四款方法中位居首位，但在精准度和 *F1* 分数上不及 KNNCAD 方法和 DADS 方法；KNNCAD 方法的精准度和 *F1* 分数仅次于 DADS 方法，召回率与 DADS 方法相当；DADS 方法在精准度和 *F1* 分数上均优于其他方法，并表现得更加稳定，尽管 DADS 方法召回率与 KNNCAD 方法相似，低于 LOOP 和 HST 方法。

表4-1四个方法在所有数据集上的各项平均指标

Tab4- 1 Average Metrics of Four Algorithms on All Datasets

	DADS			**HST**			**KNNCAD**			**LOOP**		
	精准度	召回率	*F1* 值	精准度	召回率	*F1* 值	精准度	召回率	*F1* 值	精准度	召回率	*F1* 值
RBF1	**0.906**	0.582	**0.704**	0.200	0.915	0.327	0.353	0.572	0.619	0.265	**0.960**	0.412
RBF2	**0.922**	0.521	**0.659**	0.344	0.900	0.494	0.365	0.574	0.620	0.460	**0.995**	0.627
HP1	**0.997**	0.573	**0.719**	0.323	0.868	0.467	0.353	0.593	0.314	0.427	**0.996**	0.594
HP2	**0.670**	0.588	**0.605**	0.312	0.915	0.461	0.236	0.387	0.418	0.328	**0.976**	0.488
Shuttle	**0.865**	0.712	**0.778**	0.344	0.874	0.494	0.365	0.534	0.538	0.422	**0.960**	0.473

结合表 4-1 与图 4-8 中的综合数据来看，DADS 方法的精准度和 *F1* 值在所有实验数据集上都是最高，这说明无论是人工数据集还是真实数据集，DADS 方法都可以在检测异常和提升聚类性能间找到平衡，通过检测异常提高模型的整体聚类性能。HST 方法的平均召回率较高但精准度和 *F1* 值较低，这说明该方法在多个数据集上，都是通过牺牲模型挖掘性能换取异常检测正确性。KNNCAD 方法平均精准度较低，平均召回率和 *F1* 值在不同数据集上波动较大，这说明该方法在异常数据集上，模型分类性能较差但异常检测效率稳定。LOOP 方法平均召回率最高，但平均精准度较低且平均 *F1* 值在不同数据波动较大，这说明该方法与 HST 方法一样，通过牺牲模型挖掘性能，提高异常检测正确率。综上所述，证明 DADS 方法，无论在人工数据集或真实数据集，都可以做到检测异常提高聚类质量。

4.4 本章小结

本章设计一种基于密度峰值和微簇结构的数据流异常检测方法，旨在改进聚类模型，并通过检测异常提升聚类质量。该方法采用在线框架，由微簇更新模块和异常检测模块构成，其中微簇学习模块采用时间衰减窗口控制微簇数量，提出微簇稀疏度定义，通过微簇稀疏度和动态加权的微簇权重判断局部异常微簇。异常检测模块基于 KNN 的局部密度定义方法，将微簇聚类划分为多个区域，并提出一种新的全局异常分数计算方法，以筛选出最有可能的异常微簇。经验证，DADS 方法在含异常的动态数据流环境，能通过检测异常提高模型聚类性能。

第 5 章 总结与展望

本章对全文研究工作和成果进行总结，对数据流动态聚类研究方向做进一步探讨。

5.1 总结

本文研究内容聚焦于基于密度方法的数据流聚类任务。现实场景中，数据的生成环境受众多因素的影响，数据流聚类面临数据形状不确定性、噪声以及挖掘模型自适应等问题。本文使用微簇结构和密度峰值聚类等方法构建数据流聚类模型，研究数据之间的关系，了解概念漂移产生的原因，探索数据流聚类新的发展方向。具体工作内容总结如下：

（1）首先，本文从数据空间动态分布角度将数据映射到一个个微簇中，为保留有效的数据概要设计不均匀衰减策略，减少长时间不更新微簇占用内存的时间。延长样本数量多，短时间不更新的微簇存在时间，保留更有价值的数据概要，更合理地安排内存空间，降低历史数据对当前聚类的影响和提高模型的鲁棒性。

（2）其次，本文为提高数据流聚类模型，抗噪和识别任意形状数据的能力，借鉴两阶段框架将密度峰值聚类算法融入数据流聚类模型中。利用不均匀时间衰减策略和微簇结构，减少密度峰值方法需要处理的对象数量减少迭代次数，降低密度峰值方法的复杂度。根据样本的邻居关系，提出基于最近邻域的局部密度计算方法，在分配阶段提高最近邻样本的权重，降低“多米诺效应”的影响，强化模型聚类精准性。经实验结果证明，本文设计的数据流聚类模型，具备较高的聚类性能和抗噪声能力

（3）最后，提出一种基于密度峰值聚类和微簇结构的数据流异常检测方法。该方法对原有的聚类模型进一步优化，加入微簇空间的分布检测机制，使得模型可以自适应检测到微簇空间发生变化，更新聚类模型。这种方法能及时发现概念漂移，并在发生概念漂移时做出更新，削弱概念漂移带来的影响。在此基础上采用增量学习的思想，将原有的两阶段聚类框架改进为在线聚类框架。并发挥密度峰值聚类方法的特性对微簇空间分局，提出全局异常筛选策略，既考虑样本的局

部信息，也考虑到样本周围的整体信息，快速准确地检测异常，降低异常数据对聚类的影响。经在多个数据集上实验证明 DADS 方法，能有效稳定地检测到异常，提升模型聚类质量。

5.2 展望

本文基于密度聚类思想，设计基于密度峰值的数据流动态聚类方法，并在聚类方法的基础上改进设计基于密度峰值和微簇结构的异常检测方法。本文研究工作虽然取得一些进展但在方法挖掘效率、自适应性和时间复杂度上均有上升空间。

（1）概念漂移处理问题。在本文工作中遇到概念漂移时，被动地分析数据分布信息以识别概念漂移并更新方法模型。对概念漂移的产生原理和相应处理方式，并没有更加深入地探讨。且概念漂移检测方面检测敏感性，可以进一步探索，比如在遇到特殊的数据流环境，可能会造成检测机制的误判使得模型产生不必要的更新，增耗时间成本。

（2）不适合处理高维数据集问题，在本文的聚类方法中虽然通过微簇结构和不均匀衰减策略，能在一定程度上提高内存利用率，减少处理时的对象数量但随着对象的特征维数增加，模型的迭代次数也会增加整体的处理时间就会增长。因此，是否可以从聚类模型的更新角度，减少聚类阶段的时间复杂度，如改进密度峰值方法聚类中心的选择机制可以更切合实际地设置聚类中心。

（3）异常检测时精准性和提前性，本文中异常检测方法检测到异常时，通常是该异常样本已经被方法学习后，这种检测是较为被动的。对能否更加主动的预判异常本文没有更深入的探讨，以及微簇结构在筛选异常时通常会不可避免地发生连带错误，能否探索一种复杂度较低对簇内样本在筛选的局部异常判断机制，以提高检测准确性减少误判这些问题都值得进一步的讨论。

参考文献

[1] 金澈清,钱卫宁,周傲英.流数据分析与管理综述[J].软件学报,2004, (08):1172-1181.

[2] 孙玉芬,卢炎生.流数据挖掘综述[J].计算机科学,2007, (01):1-5+11.

[3] 李海林,张丽萍.时间序列数据挖掘中的聚类研究综述[J].电子科技大学学报,2022, 51(03): 416-424.

[4] 王涛,李舟军,颜跃进等.数据流挖掘分类技术综述[J].计算机研究与发展,2007(11):1809-1815.

[5] 冯辉宁. 云计算环境下的多路数据流分层模块化建模与设计[J]. 系统工程理论与实践, 2013,33(06): 1570-1576.

[6] Tareq M, Sundararajan E A, Mohd M, et al. Online Clustering of Evolving Data Streams Using a Density Grid-Based Method[J]. IEEE Access, 2020, 8: 166472-166490.

[7] 贾涛,韩萌,王少峰等.数据流决策树分类方法综述[J].南京师大学报(自然科学版), 2019, 42(04): 49-60.

[8] Ukil A,Bandyoapdhyay S,Puri C,Pal A.IoT healthcare analytics:the importance of anomaly detection[C].in: 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), 2016, 994–997.

[9] Habeeb R A A,Nasaruddin F,Gani A, Hashem I A T,Ahmed E,Imran M.Real-time big data processing for anomaly detection: A survey[J].Int.J.Inf. Manag.2019,45:289–307.

[10] 吕庆礼. 基于计算机数据算法模型的交通流数据分析 [J]. 计算技术与自动化, 2024, 43 (01): 160-166.

[11] SILVA J A, FARIA E R, BARROS R C, et al. Data stream clustering: A survey[J]. ACM Computing Surveys (CSUR),2013, 46(1): 13.

[12] LI Y, YANG G, HE H, et al. A study of large-scale data clustering based on fuzzy clustering[J]. Soft Computing, 2016, 20(8): 3231-3242.

[13] ZHANG P, SHEN Q. Fuzzy c-means based coincidental link filtering in support of inferring social networks from spatiotemporal data streams[J]. Soft Computing, 2018, 22(21): 7015-7025.

[14] Rodriguez A, Laio A. Clustering by fast search and find of density peaks[J]. science, 2014, 344(6191): 1492-1496

[15] 黄东福,刘立群. 基于图神经网络的异源图像配准方法综述 [J]. 软件工程, 2025, 28 (01): 1-7.

[16] 张春昊,解滨,张佳豪. 融合改进 VAE 与 BiLSTM 的无监督时序数据异常检测方法 [J/OL]. 计算机工程, 1-14

[17] 霍冠英,盛兴琳,施淑娴. 基于全局背景光强度估计和对比度调整的无监督水下图像增强 [J/OL]. 东南大学学报(自然科学版), 1-11.

[18] 彭焘,肖遥,冯时,等. 基于多特征表示的无监督机器异常声音检测 [J]. 复旦学报(自然科学版), 2024, 63 (06): 703-710.

[19] CARNEIN M, ASSENMACHER D, TRAUTMANN H. An empirical comparison of stream clustering algorithms[C]//Proceedings of the Computing

Frontiers Conference. New York: ACM Press, 2017: 361-366
[20] DUBEY A K, GUPTA R, MISHRA S. Data stream clustering for big data sets: A comparative analysis[C]//Proceedings of IOP Conference Series: Materials Science and Engineering.[S.l.]: IOP Publishing, 2021, 1099(1): 012030.
[21] ZUBAROĞLU A, ATALAY V. Data stream clustering: A review[J]. Artificial Intelligence Review, 2021, 54(2): 1201-1236.
[22] O'CALLAGHAN L, MISHRA N, MEYERSON A, et al. Streaming-data algorithms for high-quality clustering[C]// Proceedings of 18th International Conference on Data Engineering. [S.l.]: IEEE, 2002: 685-694.
[23] PUSCHMANN D, BARNAGHI P, TAFAZOLLI R. Adaptive clustering for dynamic IoT data streams[J]. IEEE Internet of Things Journal, 2016, 4(1): 64-74.
[24] DE ANDRADE SILVA J, HRUSCHKA E R, GAMA J. An evolutionary algorithm for clustering data streams with a variable number of clusters[J]. Expert Systems with Applications, 2017, 67: 228-238.
[25] AGGARWAL C C, HAN J, WANG J, et al. A framework for clustering evolving data streams[C]//Proceedings of the 29th International Conference on Very Large Data Bases. Netherlands: Elsevier Science Ltd, 2003: 81-92.
[26] AGGARWAL C C, HAN J, WANG J, et al. A framework for projected clustering of high dimensional data streams[C]//Proceedings of the Thirtieth International Conference on Very Large Databases. Netherlands: Elsevier Science Ltd, 2004:852-863.
[27] ZHOU A, CAO F, QIAN W, et al. Tracking clusters in evolving data streams over sliding windows[J]. Knowledge and Information Systems, 2008, 15(2):181-214.
[28] UDOMMANETANAKIT K, RAKTHANMANON T, WAIYAMAI K. E-stream: Evolution-based technique for stream clustering[C]//Proceedings of International Conference on Advanced Data Mining and Applications. Berlin: Springer, 2007:605-615.
[29] LÜHR S, LAZARESCU M. Incremental clustering of dynamic data streams using connectivity based representative points[J]. Data & Knowledge Engineering, 2009, 68(1):1-27.
[30] CAO F, ESTERT M, QIAN W, et al. Density-based clustering over an evolving data stream with noise[C]//Proceedings of the 2006 SIAM International Conference on Data Mining. [S.l.]: Society for Industrial and Applied Mathematics, 2006: 328-339.
[31] NTOUTSI I, ZIMEK A, PALPANAS T, et al. Density-based projected clustering over high dimensional data streams[C]// Proceedings of the 2012 SIAM International Conference on Data Mining. [S.l.]: Society for Industrial and Applied Mathematics, 2012: 987-998.
[32] FAHY C, YANG S, GONGORA M, et al. Ant colony stream clustering: A fast density clustering algorithm for dynamic data streams[J]. IEEE Transactions on Systems, Man, and Cybernetics, 2019, 49(6): 2215-2228.
[33] HYDE R, ANGELOV P, MACKENZIE A R. Fully online clustering of evolving data streams into arbitrarily shaped clusters [J]. Information Sciences, 2017, 382: 96-114

[34] AMINI A, SABOOHI H, HERAWAN T, et al. Mudi-stream: A multi density clustering algorithm for evolving data stream [C]//Proceedings of Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on. [S.l.]: IEEE, 2013.
[35] YIN C, XIA L, ZHANG S, et al. Improved clustering algorithm based on high-speed network data stream[J]. Soft Computing, 2018, 22(13): 4185-4195.
[36] HASSANI M, SPAUS P, CUZZOCREA A, et al. I-hastream: Density-based hierarchical clustering of big data streams and its application to big graph analytics tools[C]//Proceedings of 2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid). [S.l.]: IEEE, 2016: 656-665
[37] CHEN Y, TU L. Density-based clustering for real-time stream data[C]//Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. [S.l.]: ACM, 2007: 133-142.
[38] WAN L, NG W K, DANG X H, et al. Density-based clustering of data streams at multiple resolutions[J]. ACM Transactions on Knowledge Discovery From Data (TKDD), 2009, 3(3): 14. [44] PARK N H, LEE W S. Statistical grid-based clustering over data streams[J]. ACM Sigmod Record, 2004, 33(1): 32-37.
[39] 武培成. 基于网格密度和属性依赖的数据流异常检测[D]. 太原科技大学, 2024.
[40] 赵浩钧,邹德清,薛文杰,等. 基于 BERT 与自编码器的概念漂移恶意软件分类优化 [J/OL]. 软件学报, 1-17.
[41] 吴勇华,梅颖,卢诚波. 基于增量加权的概念漂移数据流分类算法 [J/OL]. 上海交通大学学报, 1-22.
[42] 周彦冰,马士伦,文益民. 基于图结构的概念漂移检测 [J/OL]. 山东大学学报(工学版), 1-9.
[43] Schlimmer JC, Granger RH. Incremental learning from noisy data. Machine Learning, 1986, 1(3): 317-354.
[44] Webb GI, Hyde R, Cao H, et al. Characterizing concept drift. Data Mining and Knowledge Discovery, 2016, 30(4): 964-994.
[45] 朱永南. 开放场景下的鲁棒半监督学习算法研究[D].南京大学, 2021.
[46] Yaohui L, Zhengming M, Fang Y. Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy[J]. Knowledge-Based Systems, 2017, 133: 208-220.
[47] Parmar M, Wang D, Zhang X, et al. REDPC: A residual error-based density peak clustering algorithm[J]. Neurocomputing, 2019, 348: 82-96.
[48] Chen Y, Hu X, Fan W, et al. Fast density peak clustering for large scale data based on kNN[J]. Knowledge-Based Systems, 2020, 187: 104824
[49] 张清华,周靖鹏,代永杨等.基于代表点与 K 近邻的密度峰值聚类算法[J/OL].软件学报,2023(12):5629-5648.
[50] D.M. Hawkins, Identification of Outliers, vol. 11, Springer, 1980.
[51] 张靖. 基于数据分布的零样本学习方法及应用研究[D]. 北京交通大学, 2023.
[52] A. Blázquez-García, A. Conde, U. Mori, J.A. Lozano, A review on outlier/anomaly detection in time series data, ACM Comput. Surv. 54 ,2021.

[53] M. Munir, S.A. Siddiqui, M.A. Chattha, A. Dengel, S. Ahmed, FuseAD: unsupervised anomaly detection in streaming sensors data by fusing statistical and deep learning models, Sensors 19 ,2019.
[54] M. Zhang, J. Guo, X. Li, R. Jin, Data-driven anomaly detection approach for time-series streaming data, Sensors 20 ,2020.
[55] S. Ahmad, A. Lavin, S. Purdy, Z. Agha, Unsupervised real-time anomaly detection for streaming data, Neurocomputing 262 (2017) 134–147.
[56] Y. Yang, L. Chen, C. Fan, ELOF: fast and memory-efficient anomaly detection algorithm in data streams, Soft Comput. 25 (2021) 4283–4294.
[57] M. Salehi, C. Leckie, J.C. Bezdek, T. Vaithianathan, X. Zhang, Fast memory efficient local outlier detection in data streams, IEEE Trans. Knowl. Data Eng. 28 (2016) 3246–3260.
[58] A. Degirmenci, O. Karal, Efficient density and cluster based incremental outlier detection in data streams, Inf. Sci. 607 (2022) 901–920.
[59] B. Sielly Jales Costa, C.G. Bezerra, L.A. Guedes, P.P. Angelov, Online fault detection based on typicality and eccentricity data analytics, in: 2015 International Joint Conference on Neural Networks (IJCNN), 2015, 1–6.
[60] K. Wu, K. Zhang, W. Fan, A. Edwards, P.S. Yu, Rs-forest: a rapid density estimator for streaming anomaly detection, in: 2014 IEEE International Conference on Data Mining, 2014, 600–609.
[61] W. Xiaolan, M. Manjur Ahmed, M. Nizam Husen, Z. Qian, S.B. Belhaouari, Evolving anomaly detection for network streaming data, Inf. Sci. 608 (2022) 757–777.
[62] D. Krleža, B. Vrdoljak, M. Brˇci´c, Statistical hierarchical clustering algorithm for outlier detection in evolving data streams, Mach. Learn. 110 (2021) 139–184.
[63] M. Jain, G. Kaur, V. Saxena, A k-means clustering and SVM based hybrid concept drift detection technique for network anomaly detection, Expert Syst. Appl.193 (2022) 116510
[64] A. Otero, P. Félix, D.G. Márquez, C.A. García, G. Caffarena, A fault-tolerant clustering algorithm for processing data from multiple streams, Inf. Sci. 584 (2022),649–664.
[65] L. Tu, Y. Chen, Stream data clustering based on grid density and attraction, ACM Trans. Knowl. Discov. Data 3 (2009).
[66] R. Hyde, P. Angelov, A. MacKenzie, Fully online clustering of evolving data streams into arbitrarily shaped clusters, Inf. Sci. 382–383 (2017) 96–114
[67] M.K. Islam, M.M. Ahmed, K.Z. Zamli, A buffer-based online clustering for evolving data stream, Inf. Sci. 489 (2019) 113–135
[68] Pevný T, Loda: Lightweight on-line detector of anomalies, Mach. Learn. 2016, 102: 275–304
[69] Islam M.K., Ahmed M.M., Zamli K.Z., A buffer-based online clustering for evolving data stream, Inf. Sci. 2019, 489: 113–135
[70] Kashani E.S., Shouraki S.B., Norouzi Y., Evolving data stream clustering based on constant false clustering probability, Inf. Sci. 2022, 614: 1–18
[71] 张忠平,李森,刘伟雄等.基于快速密度峰值聚类离群因子的离群数据检测算法

[J].通信学报,2022,43(10):186-195.
[72]文益民,刘帅,缪裕青等.概念漂移数据流半监督分类综述[J].软件学报,2022,33(04):1287-1314.
[73]赵光华,杨焘,付冬梅.数据流形边界及其分布条件的增量式降维算法[J].智能系统学报,2023,1-10.
[74]朱颖雯,陈松灿.数据流聚类算法研究[J].数据采集与处理,2022,37(04):894-908.
[75]P.S. Maciag, M. Kryszkiewicz, R. Bembenik, J.L. Lobo, J. Del Ser, Unsupervised anomaly detection in stream data with online evolving spiking neural networks,Neural Netw. 139 (2021) 118–139.
[76]刘晋霞,张曦. STK：基于对比学习嵌入的聚类方法 [J]. 计算机科学, 2024, 51 (S2): 631-636.
[77]Xu C, Shen J, Du X, et al. An intrusion detection system using a deep neural network with gated recurrent units[J]. IEEE Access, 2018, 6: 48697-48707.
[78]屠莉,陈峻.衰减窗口中的不确定数据流聚类算法[J].计算机应用研究,2021,38(09).
[79]张华,杨磊.基于密度梯度的滑动窗口数据流任意形状聚类[J].计算机仿真,2022,39(04):316-3230.
[80]Michael Hahsler,Matthew Bolaños:Clustering Data Streams Based on Shared Density between Micro-Clusters. IEEE Trans. Knowl. Data Eng,2016. 28(6): 1449-1461 .
[81]李军,杨飞帆,龚胜,周科宇.基于视觉的轻量化路面异常检测算法[J].吉林大学学报(工学版),2024,1-9.
[82]向万,陈绪君,郑有凯,房可.基于多尺度特征记忆增强的异常行为检测算法[J].计算机工程与设计,2024,45(09): 2634-2640.
[83]Wang X,Manjur A,NizamHusen M,Qian Z,Belhaouari S B.Evolving anomaly detection for network streaming data[J].Inf.Sci.2022,608:757–777.
[84]刘壮,凌能祥.基于函数型 k 近邻分类模型的 PM2.5 研究[J].合肥工业大学学报(自然科学版),2024,47(07):967-970.
[85]Bifet A,Holmes G,Pfahringer B.MOA:Massive online analysis, a framework for stream classification and clustering[C]//Proceedings of the first workshop on applications of pattern analysis.2010,WAWP:44-50.
[86]Yilmaz,S F,Kozat,S S .PySAD:A Streaming Anomaly Detection Framework in Python[C].CoRR ,2020,abs/2009:02572 .
[87]Tan S C,Ting,K M,Liu T F,Fast anomaly detection for streaming data[J].In Twenty-Second International Joint Conference on Artificial Intelligence 2011,IJCAI:1511-1516.

攻读学位期间参与的科研项目

[1] 安徽省高校自然科学研究重点项目（项目号 KJ2021A0516）

攻读学位期间发表的学术论文目录

[1] 张国一,刘三民.基于密度峰值的数据流动态聚类算法研究[J].长春理工大学学报(自然科学版),2024,47(04):82-93.（学生一作导师二作）

[2] 张国一,刘三民.密度峰值与微簇结构相结合的数据流异常检测算法[J].天津理工大学学报(自然科学版).（已录用,学生一作，导师二作）

[3] 朱健,刘三民,皇苏斌,张国一,Transfer learning method based on cluster for drifting data stream classification,cluster computing.（已录用，中科院三区，学生四作）

致 谢

时光荏苒，三年的研究生学习生涯已接近尾声。在安徽工程大学的三年硕士生涯中，我不仅积累了丰富的学术知识，还获得了许多宝贵的人生经验。通过这段学习旅程，我学会了如何以坚定的决心和积极的心态面对生活中的困难，如何以严谨的态度和创新的精神投入学术研究中，并在工作中保持谦逊、公正和平等的立场。“尚德敏学，唯实唯新”的人生态度将伴随我走过未来的每一个阶段。临别母校之际，我要向所有曾给予我帮助、支持和关怀的人表达我最真诚的感谢。

首先，我要深深感谢我的导师刘三民教授。从我入学的那一刻起，刘老师就耐心指导我进行学术规划，帮助我明确每个学习阶段的任务，给了我明确的方向。三年来，刘老师每周都会进行学术辅导，帮助我及时发现并改正研究中的错误。从开题报告到中期检查，再到论文的完成，刘老师的悉心指导为我的研究提供了重要支持。衷心感谢刘老师的教诲与关怀。

其次，感谢计算机与信息学院的所有老师，特别是汪军、刘涛、王忠群、卢桂馥、皇苏斌、章平、陶皖、强俊、王勇、孔超、李钧、谷灵康以及戴家树等老师。感谢你们三年来无私的帮助与支持，你们的教诲不仅增强了我的专业能力，还让我在综合素质上得到了极大的提升。

最后，我要感谢我的室友和计算机研 22 班的同学们，感谢你们给予我陪伴与支持。感谢数据挖掘与知识工程课题组的每一位同学，正是你们的支持和鼓励，让我的研究生生活充满了色彩，学术上的交流和合作也让我受益匪浅。感谢你们在我求学道路上的陪伴和帮助，是你们让我不断进步，取得了今天的成绩。你们的理解和支持将始终是我前进的动力，在此向你们表示衷心的感谢。

2025 年 3 月 于安徽工程大学