基于改进密度峰值聚类的AIS数据航线挖掘方法

慕志颖[1,2]，李晓宇[2]，郑玉方[2]，郭森森[3]
1. 西北工业大学长三角研究院，江苏 太仓 215400；
2. 西北工业大学网络空间安全学院，陕西 西安 710072；
3. 西安电子科技大学通信工程学院，陕西 西安 710071

摘要
针对现有船舶轨迹挖掘方法特征点提取阈值确认难、轨迹相似度计算效率低下以及聚类算法处理海量AIS数据时耗时过长等问题，提出了基于改进密度峰值聚类的AIS数据航线挖掘方法。首先，针对特征点阈值难以确定的问题，设计自适应阈值特征点提取算法，通过计算航向变化率和速度变化率的动态阈值自动筛选关键轨迹点；其次，针对轨迹相似度计算效率低下的问题，引入FastDTW算法计算轨迹间相似度距离，克服了传统方法在时序性和轨迹数量关系上的局限性；最后，针对聚类算法处理海量数据耗时过长的问题，结合四叉树空间分区策略，提出改进的密度峰值聚类算法，实现自适应参数选择和多分区并行聚类。在厦门港周边海域1 020万条AIS数据上进行实验验证，结果表明该方法在处理海量AIS数据时，能够准确提取目标水域的主要航路，轨迹相似度计算与航线挖掘效率显著优于传统方法。

关键词
AIS数据；密度峰值聚类；空间分区

中图分类号：TP391　　文献标志码：A　　doi：10.11959/j.issn.2096-0271.2026048

AIS data route mining method based on improved density peak clustering

Mu Zhiying[1,2], Li Xiaoyu[2], Zheng Yufang[2], Guo Sensen[3]
1. Yangtze River Delta Research Institute, Northwestern Polytechnical University, Taicang 215400, China
2. School of Cybersecurity, Northwestern Polytechnical University, Xi'an 710072, China
3. School of Telecommunications Engineering, Xidian University, Xi'an 710071, China

Abstract
To address the issues of the difficulty in determining feature point thresholds, the low computational efficiency in trajectory similarity calculation and the high time complexity of clustering algorithms when processing massive AIS data in existing ship trajectory mining methods, this paper proposes an AIS data route mining method based on improved density peak clustering. Firstly, to tackle the problem of the difficulty in determining feature point thresholds, an adaptive threshold feature point extraction algorithm is designed, which automatically screens key trajectory points by calculating dynamic thresholds of course over ground (COG) change rate and speed over ground (SOG) change rate. Secondly, to address the low computational efficiency issue, the FastDTW algorithm is introduced to calculate similarity distances among trajectories, overcoming the limitations of traditional methods in terms of temporal characteristics and

trajectory quantity relationships. Finally, to solve the problem of high complexity and long processing time when clustering algorithms handle massive data, an improved density peak clustering algorithm based on the quadtree spatial partitioning strategy is proposed, which achieves adaptive parameter selection and multi-partition parallel clustering. Experimental validation is conducted using 10.2 million AIS data from waters around Xiamen Port, and the results demonstrate that the proposed method can accurately extract main routes in target waters when processing massive AIS data, significantly outperforming traditional methods in trajectory similarity calculation and route mining efficiency.

Key words
AIS data, density peak clustering, spatial partition

0 引言

随着经济全球化的深入发展，海上运输在国际贸易中的地位日益凸显，约占全球贸易运输总量的90%[1]。船舶交通量的显著增长给海上交通管理带来了前所未有的挑战，如何有效管控海域交通、保障航行安全成为海事管理部门面临的重要课题。船舶经验航线的准确提取对于了解船舶在区域内的行为模式、掌握船舶交通流的分布规律、识别共同港口和航线节点等具有重要意义，是实现智能海事监管和优化航路规划的关键技术基础。

传统的航线获取方法存在诸多局限性。人工测量方法成本高昂、更新速度缓慢、安全性差且易受环境因素影响，不适用于大规模航路测量。高精度遥感影像技术虽能提供实时监控，但需要通过卫星对船舶移动数据进行持续跟踪，成本过高且难以推广。基于图像识别技术的方法在陆地交通中应用效果良好，但在海上环境中很难有效区分航道区域和非航道区域。电子海图虽然能为船舶提供推荐航道，但由于天气条件、船舶行驶状况等因素的差异，无法准确反映船舶的实际经验航线[2]。

自动识别系统（automatic identification system，AIS）作为安装在船舶上的重要信息报告工具，不仅是引航员了解周围船舶航行意图的主要设备，也是海事监管部门对船舶进行跟踪和监管的核心技术手段[3]。AIS系统以每隔几秒或几分钟的频率持续刷新数据，产生了大量复杂的数据，为海上交通监控和可持续管理提供了丰富的数据源。然而，传统的数据处理技术在处理海量的AIS数据时面临严重挑战，海上交通监控和管理迫切需要一种高效的自动化解决方案来分析船舶运动模式并提取航线知识[4]。如何从复杂的AIS数据中准确提取高精度的路径模式已成为当前研究的重点[5-7]。

现有的AIS数据航线挖掘方法主要采用聚类分析技术，但在关键技术环节仍存在诸多瓶颈。首先，在特征点提取方面，现有方法主要包括基于压缩算法的道格拉斯-普克（Douglas-Peucker，DP）算法、基于滑动窗口的数据压缩方法以及基于航行参数变化的特征点选取方法等，但普遍存在阈值设定主观性强、压缩效果与轨迹完整性难以平衡等问题[8-9]。其次，在轨迹相似度计算方面，传统的基于几何距离的方法（如Euclidean距离、Hausdorff距离等）难以有效处理AIS数据固有的时序特性和轨迹长度差异问题，而基于动态时间规整（dynamic time warping，DTW）

的方法虽能解决时序对齐问题，但在处理大规模轨迹数据时计算复杂度过高[10-12]。最后，在聚类算法方面，现有研究主要采用K-means、DBSCAN（density-based spatial clustering of applications with noise）等传统聚类方法，这些方法在处理海量AIS数据时面临计算复杂度高、参数设置依赖人工经验、难以适应海域数据密度分布不均等问题[13]。

本文提出了一种基于改进密度峰值聚类的AIS数据航线挖掘方法。首先，该方法设计了自适应阈值特征点提取算法，通过动态计算航向变化率和速度变化率阈值实现不同轨迹的自适应压缩；然后，引入快速动态时间规整（fast dynamic time warping，FastDTW）算法计算轨迹间相似度距离，有效克服了传统距离度量在时序性和计算效率方面的局限性；最后，结合四叉树空间分区，提出改进的密度峰值聚类算法，在Spark并行计算平台下，实现了自适应参数选择和多分区并行处理。在厦门港周边海域1 020万条AIS数据上进行实验验证，结果证实了该方法的有效性和实用性。

本文的章节组织如下：第1节回顾相关工作，第2节介绍本文采用的相关算法与模型，第3节在厦门港周边海域的AIS数据上进行实验验证，第4节进行总结与展望。

1 相关工作

船舶轨迹聚类技术作为AIS数据分析的核心方法，在海事热点识别和热门航道提取中具有重要意义[4]。现有的航线挖掘方法主要采用聚类算法。其中，Chu等[13]搭建了基于Spark、Hadoop、Mesos的AIS数据处理分析平台，利用K-means算法对船舶数据进行聚类分析，从而获得更清晰的船舶轨迹信息和提升数据处理效率。Zhang等[14]在实际船舶AIS轨迹数据上应用K-means算法和DBSCAN算法等常用聚类算法进行聚类分析，比较了各算法的优缺点和适用场景。Lei[15]使用基于DBSCAN的聚类算法对多个经纬度点阵进行聚类，在真实的AIS数据集上验证了聚类结果的有效性，但DBSCAN的部分参数选择具有不确定性。除了传统聚类方法之外，有研究探索了新兴算法的应用，如Gao等[16]将基于谱聚类算法的子轨迹聚类应用于行为模式识别，Dobrkovic等[17]采用遗传算法对船舶位置信息进行聚类以研究船舶路径特征。此外，在聚类稳定性与大规模数据处理能力方面，何玉林等[18]提出了一种基于标签迭代的聚类集成算法，在随机样本划分数据块上构建多个基聚类结果，并通过标签迭代机制进行融合与优化，从而提升聚类的一致性与稳定性，在大规模数据场景下表现出较好的处理能力。

在轨迹数据预处理方面，由于船舶航行过程中AIS轨迹点数量众多且存在异常数据，特征点提取成为其中的重要环节。Tang等[8]采用DP压缩算法进行轨迹特征点提取，压缩比超过95%，但得到的特征点数量较少，不利于设计规划详细的航路。Wei等[9]使用滑动窗口算法对AIS数据压缩，但未明确说明如何设置窗口大小。Sheng等[7]选取相邻航向或航速变化率较大的点作为特征点，但未说明阈值选取方法。这些方法普遍存在阈值设定主观性强的问题。

在轨迹相似度度量方面，传统的基于距离的方法多采用Euclidean距离、Hausdorff距离等。例如，Fernandez等[10]根据Hausdorff距离将语义路由划分

为不同的子路由，该方法可以更准确地估计船舶运动。然而，这些方法难以处理AIS数据的时序特性。为此，学者引入了动态时间规整方法，Jin等[11]提出了利用改进的动态时间规整从AIS数据中提取频繁航次的框架，Li等[12]采用DTW距离进行轨迹间相似度距离计算，无需考虑AIS各层级的数据分类问题。但DTW算法在处理大规模数据时计算复杂度较高。

针对海量AIS数据的计算挑战，Dobrkovic等[17]、Filipiak等[19]分别采用四叉树[20]和K-D树进行空间分区，加快了算法速度，也解决了不同密度区域的参数选取问题。王佳玉等[21]根据轨迹数据的空间分布特性，将整个数据空间划分为若干个局部矩形分区，实行分布式并行计算。杨帆等[22]结合Spark分析异常信息，检测出最终的实时流量异常数据，实验结果表明异常交通数据可以在短时间内被检测和分析。

综上，现有研究在AIS数据航线挖掘方面已取得一定进展，但仍存在以下关键问题：（1）在轨迹预处理与特征点提取方面，现有方法对于特征点选取、滑动窗口尺度等多依赖经验阈值或固定参数设置，主观性较强，缺乏自适应机制；（2）在轨迹相似度度量方面，传统距离度量难以兼顾时序特性，而DTW等方法虽能刻画时序差异但在大规模AIS数据下计算开销较高；（3）在聚类分析方面，部分算法在处理海量数据时复杂度高，参数设置依赖人工经验。

2 基于改进密度峰值聚类的AIS数据航线挖掘

本文提出了一种基于改进密度峰值聚类的AIS数据航线挖掘方法，该方法旨在通过分析船舶历史自报告定位数据，提取航道中最常见的航线，从而综合表征海上交通。本文方法包括数据预处理、相似度距离计算和聚类分析3个主要阶段，如**图1**所示。

在数据预处理阶段，针对原始AIS轨迹点数量庞大且冗余的问题，采用自适应阈值特征点提取算法从千万个AIS点轨迹中提取关键特征点组成特征轨迹。在相似度距离计算阶段，考虑到AIS数据的时序特性和轨迹长度差异，采用FastDTW算法计算轨迹间相似度距离来衡量两轨迹间的相似程度。在聚类分析阶段，由于特征点数量较多且分布不均匀，按照经纬度坐标对特征点进行四叉树空间分区，并基于Spark平台改进密度峰值聚类算法，最终提取船舶热门航线。

2.1 自适应阈值特征点提取

AIS系统传输与船舶相关的数据，包括船舶名称、呼号、海上移动通信业务标识码（maritime mobile service identity，MMSI）等静态数据和速度、航向等动态信息。虽然AIS数据包含多种类型的信息，但航路发现的最小信息集是船舶的位置（经度和纬度）。为了更好地挖掘航线信息并获得更好的聚类结果，需要将对地航向（course over ground，COG）和对地航速（speed over ground，SOG）等动态信息引入分析模型。船舶航道轨迹的表示如下。

$$\boldsymbol{T}_i=(\boldsymbol{P}_{t_1},\boldsymbol{P}_{t_2},\cdots,\boldsymbol{P}_{t_m})^i=[x,y,v,w]^i_{t_m} \quad (1)$$

其中，$\boldsymbol{T}_i$代表整条船舶轨迹，i代表MMSI编号。$\boldsymbol{P}_{t_m}$是同一船舶在时间戳t_m的位置信息，由特征向量$[x,y,v,w]_{t_m}$构成，(x,y)代表经度和纬度，v是SOG，w是COG。

船舶轨迹特征点的定义如下：当船舶

轨迹特征点丢失时，轨迹点集描述船舶轨迹的能力下降，船舶轨迹还原的完整性降低。选取轨迹代表点的基本标准有两个：简明性和精确性。简洁性是指选择的轨迹点数量应尽可能少，精确性是指原始轨迹与其代表集之间的差异应尽可能小。因此，寻找适当的阈值，既可以降低算法复杂度，又能保证轨迹不失真。

结合Sheng等[7]、肖潇等[23]提出的压缩方法的优缺点，本文提出了自适应阈值特征点提取算法。该算法不仅能够根据不同的船舶行驶情况（直线匀速行驶、弯道变向行驶）进行自适应压缩，还解决了手动设置阈值的问题，压缩比相比其他算法更高，得到的特征点较多，有利于设计更详尽的航路。具体而言，AIS系统在船舶COG或SOG短时间内发生变化时发送消息，因此定义航向变化率CRC和速度变化率CRS。

$$\mathrm{CRC}=\frac{|W_{P_{t_m}}-W_{P_{t_{m-1}}}|}{t_m-t_{m-1}} \tag{2}$$

$$\mathrm{CRS}=\frac{|V_{P_{t_m}}-V_{P_{t_{m-1}}}|}{t_m-t_{m-1}} \tag{3}$$

其中，$W_{P_{t_m}}$与$V_{P_{t_m}}$分别为t_m时刻的航向和速率信息，CRC是单一轨迹段时间上相邻AIS数据点的航向变化率，CRS是单一轨迹段时间上相邻AIS数据点的速度变化率。

如果直接对所有CRC、CRS求和，必然会受到极端变化率值的影响，因此须剔除极小值和极大值，再将剩下范围内合法值的均值作为标准阈值。由于静止轨迹段的CRC或CRS在0～0.1内变动，其阈值大小几乎为0，因此该方法对静止点的过滤非常有效。在具体实现中，首先按照MMSI编码将轨迹数据分成多个单条轨迹，根据UNIX时间戳对每条轨迹进行排序，计算相邻点的CRC和CRS；

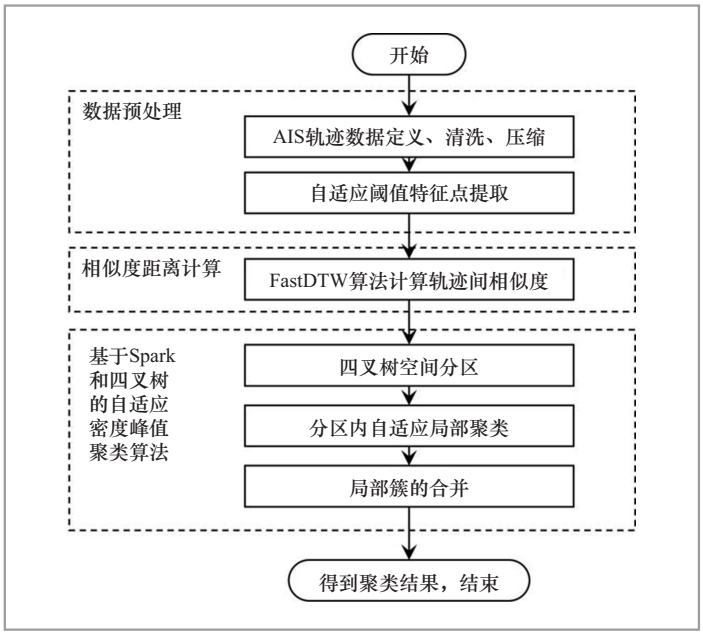

图1 经验航线挖掘流程

然后将CRC和CRS进行排序，剔除前20%较大的差值和后20%较小的差值，再计算CRC和CRS的平均值$\overline{\mathrm{CRC}}$和$\overline{\mathrm{CRS}}$作为阈值（该处理借鉴了稳健统计中的截断思想，通过排除极端异常差值和过小差值对统计结果的干扰，在保证样本充分性的同时提高了阈值估计的稳定性）；最后筛选出$\mathrm{CRC}>\overline{\mathrm{CRC}}$及$\mathrm{CRS}>\overline{\mathrm{CRS}}$的AIS点作为特征点。剔除前后各20%的数据，该截断比例属于中等截断强度，在保证样本数量充分的前提下，可有效抑制分布两端异常波动对统计量的影响。当截断比例过小时，异常值抑制能力有限；当比例过大时，可能造成有效样本信息损失。截断比例20%作为折中设置，在样本规模较大的情况下能够兼顾稳健性与统计效率。这既符合Sheng等[7]将航速和航向作为特征点提取依据的思想，也可以在面对不同船舶运动形态时将不同的$\overline{\mathrm{CRC}}$和$\overline{\mathrm{CRS}}$作为阈值，从而实现不同程度的轨迹压缩。

2.2 基于FastDTW算法的相似度距离计算

相似性度量是实现轨迹聚类和分类的基础[24]。轨迹中的每个位置都用一个多维特征向量表示。由于AIS数据是典型的不对齐离散时间序列，结合其空间目标的形状差异与相对位置差异，采用DTW算法计算轨迹间的相似度距离。选择DTW算法的主要原因如下：（1）AIS数据点时间先后顺序直接影响相似度距离计算；（2）不同船舶在同一海域存在不同的行驶轨迹；（3）相同船舶在同一海域不同时间的轨迹不同；（4）不同船舶轨迹点的数量不同；（5）该算法可解决欧式距离时序性、轨迹数量关系限制的问题，具有较好的鲁棒性。

DTW算法基于上述船舶特征向量，建立轨迹 $\boldsymbol{T}_{\mathrm{a}}$ 和轨迹 $\boldsymbol{T}_{\mathrm{b}}$ 的距离矩阵

$$\boldsymbol{D}_{ij}=\left\|\boldsymbol{T}_{a_i}-\boldsymbol{T}_{b_j}\right\| \tag{4}$$

其中，$i=1,2,3,\cdots,n$，$j=1,2,3,\cdots,m$，$\boldsymbol{T}_{\mathrm{a}}$ 和 $\boldsymbol{T}_{\mathrm{b}}$ 定义如下。

$$\boldsymbol{T}_{\mathrm{a}}=(\boldsymbol{P}_{t_1},\boldsymbol{P}_{t_2},\cdots,\boldsymbol{P}_{t_n})^i=[x,y,v,w]^i_{t_n}=[\mathrm{Lon,\ Lat,\ Sog,\ Cog}]^i_{t_n} \tag{5}$$

$$\boldsymbol{T}_{\mathrm{b}}=(\boldsymbol{P}_{t_1},\boldsymbol{P}_{t_2},\cdots,\boldsymbol{P}_{t_m})^j=[x,y,v,w]^j_{t_m}=[\mathrm{Lon,\ Lat,\ Sog,\ Cog}]^j_{t_m} \tag{6}$$

构造后的规整路径为 W，路径长度为 K，第 k 个元素为 w_k。

$$W=w_1,w_2,\cdots,w_k \tag{7}$$

其中，$w_k=(i,j)$ 表示 $\boldsymbol{T}_{\mathrm{a}}$ 的第 i 个轨迹点与 $\boldsymbol{T}_{\mathrm{b}}$ 的第 j 个轨迹点相匹配。若存在最优规整路径，设规整路径 W 的距离代价为 $\mathrm{Dist}(W)$，$\mathrm{Dist}(w_{ki},w_{kj})$ 表示 $\boldsymbol{T}_{\mathrm{a}}$ 的第 i 个轨迹点和 $\boldsymbol{T}_{\mathrm{b}}$ 的第 j 个轨迹点进行匹配后的距离。

$$\mathrm{Dist}(W)=\sum_{k=1}^{K}\mathrm{Dist}(w_{ki},w_{kj}) \tag{8}$$

验证 W 为最优规整路径的有效性条件如下：

（1）单调性：单调性表示规整路径的顺序点应向前匹配，不能存在“回头”“交叉”现象，即 $(w_{k+1}-w_k)\in\{(0,1),(1,0),(1,1)\}$，$w_k=(i,j)$；

（2）边界性：边界性要求两匹配轨迹的首尾对应，即 $w_1=(1,1)$，$w_k=(n,m)$；

（3）连续性：若找到最优规整路径，即使 $\mathrm{Dist}(W)$ 最小，构造累积距离矩阵 $\boldsymbol{D}_{M\times N}$，遍历划分网格的距离代价之和：

$$\boldsymbol{D}(i,j)=\mathrm{Dist}(i,j)+\min[\boldsymbol{D}(i-1,j),\boldsymbol{D}(i,j-1),\boldsymbol{D}(i-1,j-1)] \tag{9}$$

其中，M 和 N 分别为网格的长和宽，$i=1,2,3,\cdots,n$，$j=1,2,3,\cdots,m$，$\boldsymbol{D}(0,0)=0$，$\boldsymbol{D}(i,0)=\infty$，$\boldsymbol{D}(0,j)=\infty$。

当两条船舶轨迹的AIS数量点过多（超过1 000个）时，采用DTW算法计算相似度距离所需时间将倍增。因此，本文在DTW算法的基础上，针对AIS单条轨迹数量点过多引起的轨迹间匹配时间过长问题，进一步采用FastDTW算法[25]对计算进行加速。具体而言，FastDTW采用了粗粒度化、投影和细粒度化3种策略。首先，对原始的时间序列进行数据抽象，数据抽象可以迭代执行多次（1/1→1/2→1/4→1/8），粗粒度数据点是其对应的多个细粒度数据点的平均值。其次，在较粗粒度上对时间序列运行DTW算法。最后，将在较粗粒度上得到的规整路径经过的方格进一步细粒度化到新的时间序列上。除了进行细粒度化之外，还可以额外在较细粒度的空间内向外（横向、竖向、斜向）扩展 ϑ 个粒度，ϑ 为半径参数。

FastDTW算法的执行过程如**图2**所

示。其中，第1个子图展示了以1/8的粒度执行DTW算法，而第2个子图显示了将1/8的粗粒度空间计算得到的对齐路径延续至方格细粒度化后的结果。全局最优扭曲路径可能不完全包含在投影路径中，为了提高找到全局最优解的可能性，用半径参数来控制投影路径每一侧上的额外单元数，这些单元格在规整路径优化过程中被纳入计算。**图2**中的半径参数为1，由于半径参数而在规整路径细化过程中的单元格被轻微着色，再执行DTW算法得到规整路径。然后，规整路径以1/4分辨率细化，将投影到1/2分辨率，并以半径1扩展，然后再细化。最后，规整路径被投影到全分辨率（1/1分辨率）矩阵，扩展半径后再细化，得到精确的规整路径结果。

在算法复杂度方面，当两条序列长度均约为N时，传统DTW算法的时间复杂度为$O(N^2)$，而FastDTW通过多分辨率递归搜索策略将计算复杂度降低至线性复杂度$O(N)$，从而有效提升了大规模AIS航迹相似度计算的效率。在标准数据集上，选取两集合点的数目为20、200、2 000、4 000，DTW算法和FastDTW算法的运行时间（秒）见**表1**。当集合点的数目小于200时，DTW算法和FastDTW算法的运行时间几乎相同；当数目超过2 000时，FastDTW算法的效率显著高于DTW算法。

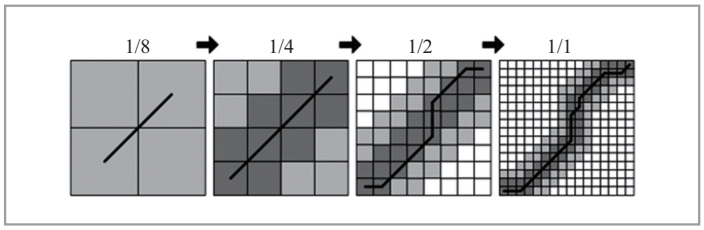

图2 FastDTW算法的执行过程

表1 DTW算法与FastDTW算法的运行时间对比

算法	两集合点数目			
	20	200	2 000	4 000
DTW	0.001	0.098	10.023	40.195
FastDTW	0.001	0.015	0.218	0.407

2.3 基于四叉树的自适应密度峰值聚类算法

密度峰值聚类算法（density peaks clustering，DPC）[26]是Rodriguez等最早提出的聚类算法，可以实现任意形状和大小的聚类，但该算法在进行海量数据计算时复杂度较高，聚类效率相对较低，不适合大规模数据聚类。通常情况下，一片海域在某一时间段内的AIS数据点的数量会达到千万级别，轨迹数量有上万条，即使经过清洗、压缩仍会保留大量的AIS数据点。若想要保留较为真实的船舶航迹以设计更详尽的航路，必须对基础的密度峰值聚类算法进行改进，使其适用于船舶轨迹规划领域。因此，本文提出四叉树的自适应密度峰值聚类算法（改进DPC算法），该算法通过空间分区缩小单次聚类的数据规模，通过Spark并行计算平台提升处理效率，通过自适应参数选择减少人工干预。具体实现包括空间分区、分区内局部聚类和局部簇合并3个主要步骤。

2.3.1 基于四叉树的空间分区

在进行数据分区时，对于靠近分区边界的单元网格内部点，其密度不仅与同一网格内部点的数据对象有关，也与该单元网格扩展区域的扩展点有关。因此，在计算局部密度时，需要考虑该单元网格的扩展点。在实际计算中，扩展点也属于该网格单元的数据分区，将单元网格内部点及

其扩展点共同载入内存进行计算。二者的区别是扩展点仅辅助内部点计算局部密度，不会成为聚类中心，而内部点才有可能成为该单元网格的聚类中心。在具体分区过程中，g_i 的数据分区可表示为 $P=\{\text{Point}\,|\text{Point}\in g_i+\lambda\}$，其中，$\lambda$ 为扩展阈值，内部点 $\text{Point}_i=\{x|x\in g_i\wedge x\text{可成为中心点}\}$，扩展点 $\text{Point}_j=\{x|x\in g_i\wedge x\text{不可成为中心点}\}$，内部点和扩展点共同组成数据分区 g_i 的数据对象。

数据空间的划分如**图3**所示。数据空间通过四叉树方法划分为多个重叠的分区。这种分区策略的优势在于：一方面，可以将海量数据分解为多个规模较小的子问题，降低单次聚类的计算复杂度；另一方面，通过分区间的重叠区域可以在后续步骤中进行局部簇的合并，避免因分区边界导致的聚类结果分割问题。

2.3.2 分区内自适应局部聚类

在数据分区完成后，对每个分区内的序列采用 FastDTW 算法进行相似度距离计算以提高运行效率。同时，对密度峰值聚类算法进行改进以实现分区内局部聚类，全过程在 Spark 平台上实现，多个分区同时进行局部聚类从而加快聚类速度。

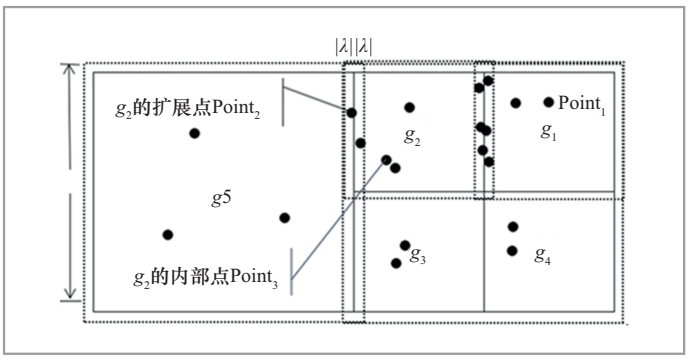

图3 数据空间的划分

局部密度和相对距离优化：针对原始 DPC 算法计算局部密度 ρ_i 和相对距离 δ_i 时需要进行全局遍历的问题，将全局计算转为分区内局部计算。重新定义在单数据分区 P 上的计算方式，只有内部点才可能成为聚类中心，扩展点只能辅助计算。

$$\rho_i=\sum_{i\neq j}\exp[-(\frac{d_{ij}}{d_c})^2\ ],\ i,j\in P \tag{10}$$

$$\delta_i=\begin{cases}\max\limits_{i\neq j}(d_{ij}),\rho_i=\max(\rho)\\ \min\limits_{j:\rho_j>\rho_i}(d_{ij}),\rho_i\neq\max(\rho)\end{cases},\ i,j\in P \tag{11}$$

其中，d_c 为截断距离，d_{ij} 为数据分区 P 内的两点间的距离。

聚类中心数目的自适应优化：针对 DPC 原算法无法自动确定聚类结果的簇数目、常常需要人为手工干预的问题，使用决策值 $\gamma_i=\rho_i\times\delta_i$ 来自动确定聚类中心。决策值 γ_i 越大，成为聚类中心的可能性也越大。对 $\{\gamma\}_{i=1}^{N}$ 进行降序排序，找到排序后整体变化最大的点。设临界点 Q 是 $\gamma_{[1\sim Q]}$ 和 $\gamma_{[Q\sim n]}$ 前后整体变化最大的点，用斜率表示变化程度，则点 Q 满足以下条件。

$$\begin{aligned}&\alpha(j)=\sum_{j=1}^{n-2}\left|\left|k_j\right|-\left|k_{j+1}\right|\right|\\&\beta=\alpha(j)/(n-2)\\&P=\max\{i\left|\left|\left|k_j\right|-\left|k_{j+1}\right|\right|\geq\beta,i=1,2,\cdots,n-2\}\end{aligned} \tag{12}$$

其中，k_j 为点 j 和点 $j+1$ 之间形成的线段的斜率，$\alpha(j)$ 为相邻两点间斜率差的总和，β 表示排序后相邻两点斜率差的平均值，临界点 P 是相邻两点间斜率差不小于 β 的序号最大的点。

2.3.3 局部簇合并

在数据分区阶段将数据集划分为若干

组互相重叠的数据分区，重叠区域必然是多个数据分区的扩展空间。如果聚簇中心点 $\text{Point}_1 \in C_1$，$\text{Point}_2 \in C_2$，Point_1，$\text{Point}_2 \in P_1 \Lambda P_2$，那么合并聚类簇 C_1 和聚类簇 C_2，指定新的聚簇中心点成为合并后新簇的中心。如**图 4** 所示，在进行局部簇合并时，主要对象是数据分区后扩展空间上的点，不会改变其他内部簇的聚类结果，从全局范围减轻了因分区不合理而对聚类结果造成的影响。此外，在分区边界点较为稀疏或分区之间重叠区域不足的情况下，在局部簇合并过程中进一步结合簇整体的空间分布特征进行判断，避免仅依据扩展空间内的少量重叠点进行合并，从而减少簇分割或重复聚类现象的发生。

综上，本文方法由自适应阈值策略、相似度计算优化及改进的峰值密度聚类框架构成。其中，自适应阈值策略用于实现特征点的稳定筛选；FastDTW 算法用于降低轨迹相似度计算复杂度；峰值密度聚类框架负责簇结构划分。三者分别在密度估计稳定性、计算效率与结构识别 3 个关键环节中发挥作用，共同构成了完整的航线挖掘流程。

算法 1 改进密度峰值聚类航线挖掘算法

输入：AIS 数据 D

输出：航迹聚类结果 C

1：计算每条轨迹相邻点的差值序列，并根据式（2）、式（3）计算 CRC 与 CRS；

2：根据 CRC 与 CRS 对轨迹点进行筛选，并基于筛选后的特征点重构特征轨迹集合 S_T；

3：基于空间分布构建四叉树空间分区：$(P_1, P_2, \cdots, P_k)$；

4：for i = 1 to k do

5：在分区 P_i 内采用 FastDTW 算法计算轨迹间的相似度距离；

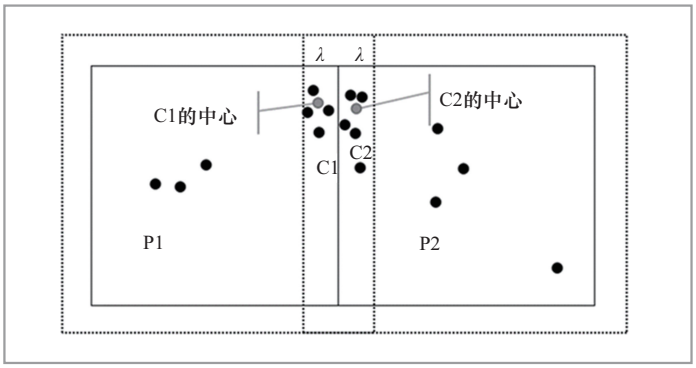

图 4　局部簇合并示意图

6：基于相似度距离计算局部密度 ρ 和相对距离 δ；

7：进行自适应局部聚类，得到局部聚类结果 C_i。

8：end for

9：对所有 C_i 进行局部簇合并，得到全局聚类结果 C；

10：return C

2.3.4　复杂度分析

设数据集 D 的对象点个数为 N，集群共有 M 个计算结点并行计算。在数据分区阶段时，多个计算结点可同时运行，时间复杂度为 $\mathrm{O}\left(\frac{N}{M}\right)$。在分区内局部聚类阶段，假设每个空间分区的网格点数量为 n，局部聚类时需要计算 n 个数据对象的局部密度和相对距离，每个数据分区的时间复杂度为 $\mathrm{O}(n^2)$，因此该阶段的时间复杂度为 $\mathrm{O}\left(\frac{N}{n} \times \frac{n^2}{m}\right) = \mathrm{O}\left(\frac{Nn}{m}\right)$。最后进行局部簇的合并，需要遍历全部扩展点找到聚类中心点，时间复杂度为 $\mathrm{O}\left(\frac{N}{M}\right)$，总时间复杂度如下。

$$T = \mathrm{O}\left(\frac{N}{M}\right) + \mathrm{O}\left(\frac{Nn}{m}\right) + \mathrm{O}\left(\frac{N}{M}\right) \quad (13)$$

3 实验与验证

3.1 实验设置

（1）实验数据

为了验证改进密度峰值聚类算法在标准数据集上的有效性，本文分别在开源数据集和真实数据上进行实验。其中，选取的开源数据集包括Flame、R15、D31、Compound、Aggregation共5个数据集；真实AIS数据为厦门港及周边水域2018年12月21日到2019年1月3日的真实船舶AIS数据。

（2）实验环境

所有实验在完全相同的实验环境中运行，Spark集群包括3台计算机，系统为Centos 7，其中一台为Master结点（充当Driver角色），两台为Slave结点（充当Executor角色），具体配置见**表2**。

（3）基线方法

本文选取当前船舶轨迹聚类使用较为广泛的DBSCAN与OPTICS聚类算法作为基线方法。其中，DBSCAN把具有足够高密度的区域划分为簇，并可在噪声的空间数据库中发现任意形状的聚类；OPTICS聚类算法是一种对空间中的数据进行聚类的算法，该算法旨在基于密度分布自动确定数据集的聚类数量和结构，并输出一个按照密度值进行排序的样本队列。

需要指出的是，现有轨迹聚类方法在研究目标和技术路线上存在一定差异。部分方法侧重于轨迹分段与形状模式挖掘，而本文关注的是基于整体轨迹相似度的热点航线结构提取，并重点解决大规模AIS数据场景下的特征点提取阈值确认和计算效率问题。因此，在基线方法选择上，主要选取与本文问题设定一致的密度类聚类算法进行对比分析。

（4）评价指标

本文主要采用邓恩指数（Dunn validity index，DVI）和戴维森堡丁指数（Davies-Bouldin index，DBI）来进行聚类评价，DVI由任意两个簇元素的最短距离（类间距离）除以任意簇中的最大距离（类内距离）得到。邓恩指数与聚类效果呈正相关，DVI越大意味着类间距离越大同时类内距离越小。

$$\mathrm{DVI}=\frac{\min\limits_{1<i<j\leqslant n} d(i,j)}{\max\limits_{1\leqslant k\leqslant n} d(k)} \tag{14}$$

其中，该方法将类簇分为n类，$d(i,j)$表示在n类中任意两类距离的最小值，$d(k)$为n类中类内距离的最大值。

DBI又称为分类适确性指标，该指标的目的是度量每个簇类最大相似度的均值。戴维森堡丁指数越小，则类内距离越小，类间距离越大，其计算如下。

$$\mathrm{DBI}=\frac{1}{k}\max_{j\neq i}\left(\frac{\overline{C_i}+\overline{C_j}}{\left\|\omega_i-\omega_j\right\|_2}\right) \tag{15}$$

表2 Spark集群配置

主机	IP地址	服务	系统
Master	192.168.33.11	Zookeeper Mesos Hadoop Spark	Centos 7
Slave 1	192.168.33.12	Zookeeper Mesos Hadoop Spark	Centos 7
Slave 2	192.168.33.13	Zookeeper Mesos Hadoop Spark	Centos 7

其中，n表示将数据集分成n类，$\overline{C_i}+\overline{C_j}$表示$n$类簇的边缘点到中心点的和，$\|\omega_i-\omega_j\|$表示任意类簇间的差的绝对值。

3.2 基于真实AIS航线数据的聚类结果分析

为了全面评估改进密度峰值聚类算法的性能，本文在10个分区上进行了对比实验。实验采用本文方法、DBSCAN算法和OPTICS算法3种方法，从戴维森堡丁指数、邓恩指数和簇数目3个维度对聚类效果进行定量评估。

表3给出了3种算法在10个分区上的详细实验结果。从表中数据可以看出，本文方法在绝大多数分区上都取得了更优的聚类性能。具体而言，在戴维森堡丁指数方面，本文方法在9个分区中均获得了最小值，相对基线算法均产生了更紧凑、分离度更好的聚类结果。在邓恩指数方面，本文方法在9个分区中表现最优，显著高于对比算法，说明本文方法的聚类结果可实现更小的类内距离和更大的类间距离。

此外，在簇数目方面，本文方法在大多数分区中识别出了更多的聚类簇。例如，在分区4中识别出9个簇，分区5中识别出7个簇，而DBSCAN和OPTICS算法通常只能识别出2～3个簇，这表明本文方法具有更强的细粒度聚类能力，能够发现更多潜在的航道信息，可为航路规划提供更丰富的选择。

3.3 基于开源数据集的聚类效果分析

本文方法改进了DPC算法，为充分验证改进后的DPC算法的有效性，在Flame、R15、D31、Compound、Aggregation共5个标准数据集上使用匈牙利算法[27]评估改进前后的DPC算法的聚类准确性，实验结果见**表4**。

Flame数据集簇数目较少，改进DPC算法的准确率能够达到0.983，效果非常好；R15数据集的簇数目有15个，改进DPC算法的准确率为0.935，说明局部簇合并能够提高聚类效果；D31数据集的类簇数目较多，存在较为复杂的数据分布，聚类准确率为0.895，准确率稍有下降；Compound和Aggregation是典型的数据

表3　在10个分区上聚类结果对比

分区	本文方法			DBSCAN聚类			OPTICS聚类		
	戴维森堡丁指数	邓恩指数	簇数目/个	戴维森堡丁指数	邓恩指数	簇数目/个	戴维森堡丁指数	邓恩指数	簇数目/个
1	**2.38**	6.27	2	3.22	**9.65**	**3**	2.79	6.40	**3**
2	**1.48**	**26.80**	2	2.02	3.44	2	3.83	16.46	**3**
3	2.18	**12.87**	3	**0.99**	3.26	2	2.73	11.38	**5**
4	**2.26**	**6.78**	**9**	4.24	3.29	2	2.83	5.68	2
5	**1.65**	**12.45**	**7**	1.67	2.46	2	2.01	8.46	3
6	**2.44**	**6.01**	**3**	—	—	1	3.01	3.25	2
7	**1.19**	**9.57**	**2**	1.56	3.06	**2**	1.72	5.83	**2**
8	**1.48**	**26.02**	**3**	2.18	3.24	2	7.07	13.29	2
9	**1.51**	**14.56**	**2**	2.46	3.13	**2**	2.24	**14.56**	**2**
10	**1.96**	**11.17**	**4**	2.29	3.30	2	3.32	9.38	2

表4 改进前后的DPC算法的聚类准确性对比

数据集	DPC算法	本文方法
Flame	1.0	0.983
R15	0.996	0.935
D31	0.968	0.895
Compound	0.832	0.824
Aggregation	0.997	0.937

密度分布不均匀的数据集，改进的DPC算法的聚类准确率也能达到0.824和0.937。综上所述，改进DPC算法在处理簇数目较多、数据密度分布不均匀的数据时仍能保持较高的准确率，具有处理不规则且复杂多变的船舶运动轨迹的能力。

此外，计算效率是评估算法实用性的重要指标，特别是在处理海量AIS数据时，算法的运行时间直接影响其可行性。在4个不同规模的目标坐标点数据集上测试DCP算法和本文方法的聚类运行时间（秒），分别为A1（1 000个坐标点）、A2（5 000个坐标点）、A3（10 000个坐标点）、A4（20 000个坐标点），实验结果如**表5**所示。

当测试数据集的数据量为1 000和5 000时，原始DPC算法的运行时间较短，分别为5 s和119 s，这是因为原始DPC算法无需进行数据分区和通信结点间的信息交换。相比之下，本文方法由于需要进行四叉树空间分区等预处理步骤，运行时间相对较长。然而，在数据集的数据量较大时，本文方法的运行时间明显较短。在A3数据集上，原始DPC算法的运行时间急剧增加至485 s，而本文方法仅需235 s；在A4数据集上，原始DPC算法需要1 938 s，而本文方法仅需257 s，运行时间仅占原始DPC算法的13.26%。

表5 改进前后的DPC算法的聚类运行时间对比

方法	A1（1 000）	A2（5 000）	A3（10 000）	A4（20 000）
DCP算法	5	119	485	1 938
本文方法	223	230	235	257

综上，本文方法在聚类准确性上较原始DPC算法略有降低但相差不大。本文方法的相似度距离计算仅在单个分区内进行，在处理大规模数据时具有显著的效率优势，更适合海量AIS数据的实际应用场景。

4 结束语

本文针对船舶轨迹识别中特征点提取阈值难以确定和聚类算法处理海量数据时复杂度过高等问题，提出了自适应阈值特征点提取算法，能够对不同数量级的船舶轨迹进行不同程度的的压缩，在保留轨迹特征的同时降低船舶轨迹点密度；引入FastDTW算法，解决了欧式距离等时序性和轨迹数量关系限制问题，该算法具有较好的鲁棒性；结合四叉树提出了改进密度峰值聚类算法，在Spark平台下能够并行处理百万数据点，且聚类结果准确度和计算效率得到了提升。在厦门港周边海域AIS数据上进行实验验证，结果表明本文方法能够准确提取目标水域的主要航路，聚类的结果更具实际意义。下一步将尝试结合实时计算组件（如Spark Streaming），实时接收和处理AIS数据，为实际船舶航行提供指导。

参考文献：

[1] Kondratenko A A, Zhang M Y, Tavakoli S, et al. Existing technologies and sci-

entific advancements to decarbonize shipping by retrofitting[J]. Renewable and Sustainable Energy Reviews, 2025, 212: 115430.

[2] Lu N H, Liang M H, Zheng R T, et al. Historical AIS data-driven unsupervised automatic extraction of directional maritime traffic networks[C]//Proceedings of the 2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA). Piscataway: IEEE Press, 2020: 7-12.

[3] Wang Y M. Application of neural network in abnormal AIS data identification [C]//Proceedings of the 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA). Piscataway: IEEE Press, 2020: 173-179.

[4] 胡昕源, 谢磊, 常吉亮, 等. 改进Quick-Bundles算法在船舶轨迹聚类中的应用[J]. 中国航海, 2023, 46(3): 145-152.
Hu X Y, Xie L, Chang J L, et al. Application of improved QuickBundles algorithm in ship trajectory clustering[J]. Navigation of China, 2023, 46(3): 145-152.

[5] Fujino I, Claramunt C, Boudraa A O. Extracting 4-attributes vessel courses from AIS data with PQK-means and topic model[C]//Proceedings of the 3rd International Conference on Big Data Research. New York: ACM, 2020: 129-135.

[6] Wang G L, Meng J L, Han Y B. Extraction of maritime road networks from large-scale AIS data[J]. IEEE Access, 2019, 7: 123035-123048.

[7] Sheng P, Yin J B. Extracting shipping route patterns by trajectory clustering model based on automatic identification system data[J]. Sustainability, 2018, 10 (7): 2327.

[8] Tang C H, Wang H, Zhao J H, et al. A method for compressing AIS trajectory data based on the adaptive-threshold Douglas-Peucker algorithm[J]. Ocean Engineering, 2021, 232: 109041.

[9] Wei Z K, Xie X L, Zhang X J. AIS trajectory simplification algorithm considering ship behaviours[J]. Ocean Engineering, 2020, 216: 108086.

[10] Fernandez Arguedas V, Pallotta G, Vespe M. Maritime traffic networks: from historical positioning data to unsupervised maritime traffic monitoring[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19(3): 722-732.

[11] Jin X Y, Yang Y, Qiu X S. Framework of frequently trajectory extraction from AIS data[C]//Proceedings of the 7th International Conference on Computer Engineering and Network (CENet2017). Trieste: Sissa Medialab (PoS), 2017: 098.

[12] Li H H, Liu J X, Liu R, et al. A dimensionality reduction-based multi-step clustering method for robust vessel trajectory analysis[J]. Sensors, 2017, 17 (8): 1792.

[13] Chu X M, Lei J Y, Liu X L, et al. KMEANS algorithm clustering for massive AIS data based on the spark platform[C]//Proceedings of the 2020 5th International Conference on Control, Robotics and Cybernetics (CRC). Piscataway: IEEE Press, 2020: 36-39.

[14] Zhang Z Y, Ni G X, Xu Y G. Comparison of trajectory clustering methods based on K-means and DBSCAN[C]//Proceedings of the 2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA). Piscataway: IEEE Press, 2020: 557-561.

[15] Lei B. A DBSCAN based algorithm for

ship spot area detection in AIS trajectory data[C]//Proceedings of the 3rd International Conference on Mechanical, System and Control Engineering. Les Ulis: EDP Sciences, 2019: 01008

[16] Gao M, Shi G Y. Ship-handling behavior pattern recognition using AIS sub-trajectory clustering analysis based on the T-SNE and spectral clustering algorithms[J]. Ocean Engineering, 2020, 205: 106919.

[17] Dobrkovic A, Iacob M E, Van Hillegersberg J. Using machine learning for unsupervised maritime waypoint discovery from streaming AIS data[C]//Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business. New York: ACM, 2015: 1-8.

[18] 何玉林, 杨锦, 黄哲学, 等. 基于标签迭代的聚类集成算法[J]. 智能科学与技术学报, 2024, 6(4): 466-479.
He Y L, Yang J, Huang Z X, et al. Label iteration-based clustering ensemble algorithm[J]. Chinese Journal of Intelligent Science and Technology, 2024, 6(4): 466-479.

[19] Filipiak D, Węcel K, Stróżyna M, et al. Extracting maritime traffic networks from AIS data using evolutionary algorithm[J]. Business & Information Systems Engineering, 2020, 62(5): 435-450.

[20] Finkel R A, Bentley J L. Quad trees a data structure for retrieval on composite keys[J]. Acta Informatica, 1974, 4(1): 1-9.

[21] 王佳玉, 张振宇, 褚征, 等. 一种基于轨迹数据密度分区的分布式并行聚类方法[J]. 中国科学技术大学学报, 2018, 48(1): 47-56.
Wang J Y, Zhang Z Y, Chu Z, et al. A trajectory data density partition based distributed parallel clustering method[J]. Journal of University of Science and Technology of China, 2018, 48(1): 47-56.

[22] 杨帆, 何正伟, 刘力荣. 基于Spark和小波分析的水上交通异常数据实时检测方法研究[J]. 陕西理工大学学报(自然科学版), 2019, 35(1): 35-41.
Yang F, He Z W, Liu L R. Research on real-time detection method of water traffic abnormal data based on Spark and wavelet analysis[J]. Journal of Shaanxi University of Technology (Natural Science Edution), 2019, 35(1): 35-41.

[23] 肖潇, 邵哲平, 潘家财, 等. 基于AIS信息的船舶轨迹聚类模型及应用[J]. 中国航海, 2015, 38(2): 82-86.
Xiao X, Shao Z P, Pan J C, et al. Ship trajectory clustering model based on AIS data and its application[J]. Navigation of China, 2015, 38(2): 82-86.

[24] 郭倩, 赵津, 过弋. 基于分层聚类的个性化联邦学习隐私保护框架[J]. 信息网络安全, 2024, 24(8): 1196-1209.
Guo Q, Zhao J, Guo Y. Hierarchical clustering federated learning framework for personalized privacy-preserving[J]. Netinfo Security, 2024, 24(8): 1196-1209.

[25] Salvador S, Chan P. Toward accurate dynamic time warping in linear time and space[J]. Intelligent Data Analysis, 2007, 11(5): 561-580.

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

[27] 钮焱, 李星, 李军, 等. 基于DTW和改进匈牙利算法的句子语义相似度研究[J]. 计算机与数字工程, 2021, 49(2): 242-247.
Niu Y, Li X, Li J, et al. Research on sentence semantic similarity based on DTW and improved Hungarian algorithm [J]. Computer and Digital Engineering, 2021, 49(2): 242-247.

作者简介

慕志颖（1994–），女，博士，西北工业大学长三角研究院助理研究员，主要研究方向为数据挖掘、风格迁移、对话系统、机器翻译、文本分类。

李晓宇（1980–），男，博士，西北工业大学网络空间安全学院副研究员，主要研究方向为数据挖掘、人工智能、大数据。

郑玉方（1997–），男，博士，西北工业大学网络空间安全学院硕士生（已毕业），主要研究方向为数据挖掘、人工智能。

郭森森（1990–），男，博士，西安电子科技大学通信工程学院助理研究员，主要研究方向为数据挖掘、人工智能安全。

收稿日期: 2025-08-25

通信作者: 李晓宇,lixiaoyu@nwpu.edu.com

基金项目: 国家自然科学基金项目(No.U23B2041,No.62074131,No.62272389,No.62372069)

Foundation Items: The National Natural Science Foundation of China (No.U23B2041, No.62074131, No.62272389, No.62372069)