# تشخیص داده‌های پرت در جریان داده با استفاده از خوشه‌بندی روش DenStream

## چکیده

کاوش در داده‌های جریانی به دنبال الگوریتم‌هایی است که بتوانند در زمان واقعی خوشه‌بندی داده‌های پویا را انجام دهند و همچنین در همین روند داده‌های پرت را نیز تشخیص دهد. مقاله حاضر بررسی عمیقی را در جهت ارائه‌ی روش DenStream برای خوشه‌بندی در داده‌های جریانی ارائه می‌دهد. این روش قابلیت تطبیق با محیط‌های مختلفی از هواشناسی گرفته تا حوزه‌ی مالی و کشاورزی را داراست. با در نظر گرفتن ویژگی‌های **داده‌های جریانی** مانند **پویایی** و **ناپایداری**، DenStream برآیند یک الگوریتم پویا و **زمان واقعی** برای خوشه‌بندی این داده‌ها است که امکان **شناسایی نقاط ناهنجار و پرت** را نیز داراست. این الگوریتم در مقایسه با روش‌های سنتی، **از جمله مزایایی نظیر عدم نیاز به فرض تعداد خوشه‌ها یا قابلیت کشف خوشه‌های با اشکال مختلف** را داراست. با این وجود، مسائلی نظیر حساسیت به پارامترها و پیچیدگی در تفسیر نیازمند تحقیقات بیشتر است تا به نتیجه بهتر و دلخواهی رسید. درضمن این روش یک روش تکاملی است که ممکن است راه‌اندازی اولیه با دقت‌تری برای شرایط مختلف بخواهد.

## واژه‌های کلیدی

کاوش داده‌های جریانی، خوشه‌بندی در زمان واقعی، DenStream، تشخیص نقاط ناهنجار، داده پرت، الگوریتم پویا، پارامترهای خوشه‌بندی، الگوریتم تکاملی

## مقدمه

مقدمه‌ای که به چالش‌های موجود در خوشه‌بندی داده‌های جریانی، اهمیت آن‌ها در زمینه‌های مختلف و نیاز به روش‌هایی همچون DenStream اشاره می‌کند. همچنین، معرفی ساختار این بخش‌ها و شرح کوتاهی از محتوای بخش‌های اصلی مقاله.

در حال حاضر روش‌های مختلفی در زمینه‌های مختلفی در راستای اهداف مختلفی وجود دارد اما در این مقاله هدف ما بیشتر ارتباط با الگوریتمی است که میتواند داده‌های پرت را در داده‌های جریانی تشخیص دهد همچنین این تشخیص در روش خوشه‌بندی باشد. پس می‌توان گفت که این روش در زمینه های بسیار زیادی که دارای داده‌های بی‌نهایت و سرعت بالا و همچنین زمان واقعی، کارآمد است.

در این روش چون داده‌های جریانی وجود دارد و داده‌ها نیز ممکن است تکاملی بسیار تغییر داشته باشند ، روش مورد بحث ما چندین مزیت دارد که بسیار مهم است:

* نیازی به تعیین تعداد خوشه‌ها نیست
* پیدا کردن شکل‌های متفاوتی از خوشه‌ها
* توانایی تشخیص داده‌های پرت

همچنین باید بدانید در این نوع مسائل به دلیل بی نهایت بودن داده‌ها ما محدودیت‌های دیگری مانند **محدودیت حافظه** نیز داریم.

به طور کلی در این روش نقش داده‌های پرت و خوشه‌های واقعی به طور مکرر جابجا خواهد شد که به دلیل داده‌های تکاملی است و بعد از مدتی این‌ها باهم ادغام میگردند و نتیجه‌ی آن خوشه بندی درست‌تر است. و هر مدت زمانی (که نیز محاسبه می‌گردد) حافظه پاکسازی می‌شود و این باعث می‌شود محدودیت حافظه نیز رفع شود.



Figure 1چالش‌های داده‌های جریانی

## کارهای مرتبط

تاکنون در رابطه با خوشه‌بندی ، روش‌های متعددی (مثلا clustream [1] ( برای کشف خوشه‌های داده‌ها با اشکال مختلف ارائه شده‌اند. اما این روش‌ها بیشتر فرض می‌کنند که تمام داده‌ها در دیسک سخت ذخیره شده (یعنی تمام داده‌ها آماده هستند و به همه‌ی داده‌ها به صورت کامل دسترسی داریم و می‌توانیم بیش از چند بار دسترسی داشته باشیم) و اطلاعات کلی مربوط به داده‌ها در هر زمان قابل دسترسی هستند، بنابراین برای پردازش جریان‌های داده که نیازمند یک گذر از مجموعه داده‌ها و خوشه‌بندی با استفاده از اطلاعات محلی هستند، کارایی ندارند. به عنوان مثال، روش IncrementalDBSCAN [2] برای برنامه‌های انبار داده مناسب است، اما تنها در محیط‌های نسبتاً پایدار عملکرد خوبی دارد و با جریان‌های داده‌ای که سریعاً تغییر می‌کنند(Drift)، سازگار نیست.

اخیرا خوشه بندی جریان داده توجه زیادی به خود جلب کرده است که چگونه می‌توان جریان‌های داده را به بهترین نحو خوشه‌بندی کرد. روش‌های یک‌گذره یا تکاملی موجود، نمی‌توانند به خوبی با اشکال مختلف خوشه‌ها در جریان‌های داده کار کنند و نتایج آن‌ها اغلب محدود به خوشه‌های کروی می‌شود. به‌طور کلی، این روش‌ها فرض می‌کنند که یک مدل یکتا به پایه داده‌ها وجود دارد و نمی‌توانند با تغییرات در توزیع داده‌ها سازگاری داشته باشند.

رویکردهای تکاملی تلاش می‌کنند تا تغییراتی که در جریان داده رخ می‌دهد را در نظر بگیرند، اما روش‌های مانند CluStream یک تعداد ثابت از میکرو-خوشه‌ها را پیش‌فرض می‌کنند که ممکن است منجر به ادغام خوشه‌های طبیعی مختلف یا حتی تقسیم یک خوشه طبیعی به دو بخش شود. روش HPStream [3] مفهوم خوشه‌های تصویری را به جریان‌های داده معرفی می‌کند، اما قادر نیست شکل‌های دلخواه خوشه‌ها را در جریان‌های داده تشخیص دهد. هرچند که مکانیزم‌های ساده‌ای برای مدیریت داده‌های پرت‌ در این روش‌ها معرفی شده است، اما ناهنجاری‌ها همچنان تأثیر زیادی بر شکل میکرو-خوشه‌ها دارند.

## روش پیشنهادی (DenStream)

الگوریتم DenStream یک الگوریتم خوشه‌بندی تکاملی برای داده‌های جریانی است که به وسیله‌ی میکروخوشه‌ها و خوشه‌های هسته‌ای این امکان را فراهم می‌کند تا داده‌ها به صورت پویا در طول زمان خوشه‌بندی شوند. این الگوریتم به صورت پیوسته میکروخوشه‌ها را به روزرسانی می‌کند و با توجه به ورودی داده‌های جدید، خوشه‌های جدید تشکیل می‌دهد یا میکروخوشه‌های موجود را تغییر می‌دهد. همچنین همیشه داده‌ها یک داده پرت شناسایی می‌شوند مگر اینکه در طول پنجره گیری مشخص شود که یک داده از یک خوشه است. یا در نهایت مشخص شود که واقعا داده‌ی پرت بوده است و حافظه پاکسازی شود. همچنین در این روش از damped window استفاده می‌شود.

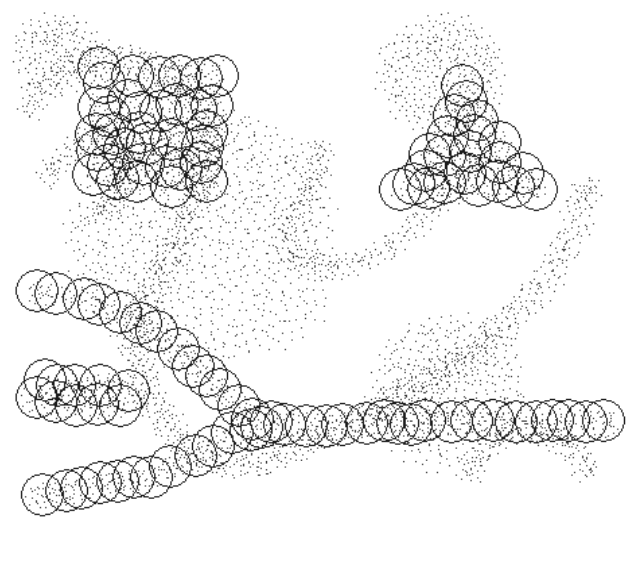


Figure 2 خوشه بندی با استفاده از DenStream

## میکروخوشه‌ها

میکروخوشه‌ها به عنوان نماینده‌های کوچک از داده‌ها عمل می‌کنند و بر اساس میزان چگالی و ارتباط با داده‌های جدید شکل می‌گیرند. هر میکروخوشه شامل دو بخش است: میکروخوشه هسته‌ای (C-Micro-Cluster) که نقطه‌ای اصلی در خوشه را نمایندگی می‌کند و میکروخوشه بالقوه (O-Micro-Cluster) که شامل داده‌های کمتر ارتباطی با خوشه دارد و به عنوان نقاط پرت مد نظر قرار می‌گیرد. فرآیند تکامل میکروخوشه‌ها:

### تشکیل اولیه میکروخوشه‌ها

- با شروع جریان داده، چند نقطه اولیه به عنوان میکروخوشه‌های هسته‌ای انتخاب می‌شوند.

### به‌روزرسانی میکروخوشه‌ها با ورود داده جدید

- هر نقطه داده جدید که وارد جریان می‌شود، با میکروخوشه‌های موجود مقایسه شده و بهترین میکروخوشه برای اختصاص دادن آن انتخاب می‌شود.

- اگر نقطه‌ای نتواند به میکروخوشه‌های موجود اضافه شود، یک میکروخوشه جدید تشکیل می‌شود یا نقطه به عنوان نقطه پرت در نظر گرفته می‌شود.

### تشکیل خوشه‌ها از میکروخوشه‌ها

- با استفاده از میکروخوشه‌های هسته‌ای و بالقوه، خوشه‌های اصلی شناسایی می‌شوند.

### شناسایی نقاط پرت

- با بررسی انحراف نقاط(استفاده از فاصله اقلیدسی) از میکروخوشه‌های هسته‌ای، نقاط پرت شناسایی می‌شوند و در نظر گرفته می‌شوند.

### تطبیق و تنظیم پارامترها

- پارامترهای DenStream از جمله epsilon، lambda، beta، و mu به طور پویا با تغییرات در جریان داده تنظیم می‌شوند تا الگوریتم بتواند به طور بهینه با داده‌های جدید سازگاری پیدا کند.

## نتایج و آزمایش‌ها

در این بخش، نتایج آزمایش‌های مختلفی که با استفاده از DenStream انجام شده است، به صورت جزییاتی گزارش می‌شود. معیارهای ارزیابی استفاده شده برای سنجش عملکرد DenStream، مقایسه نتایج با الگوریتم‌های دیگر و تحلیل عملکرد DenStream در مقایسه با آن‌ها در این بخش بیان می‌شود. همچنین، مطالعات موردی از کاربردهای واقعی DenStream مانند بررسی الگوهای مختلف هواشناسی یا مسائل مرتبط با اقتصاد و تجارت در این بخش قرار می‌گیرد.

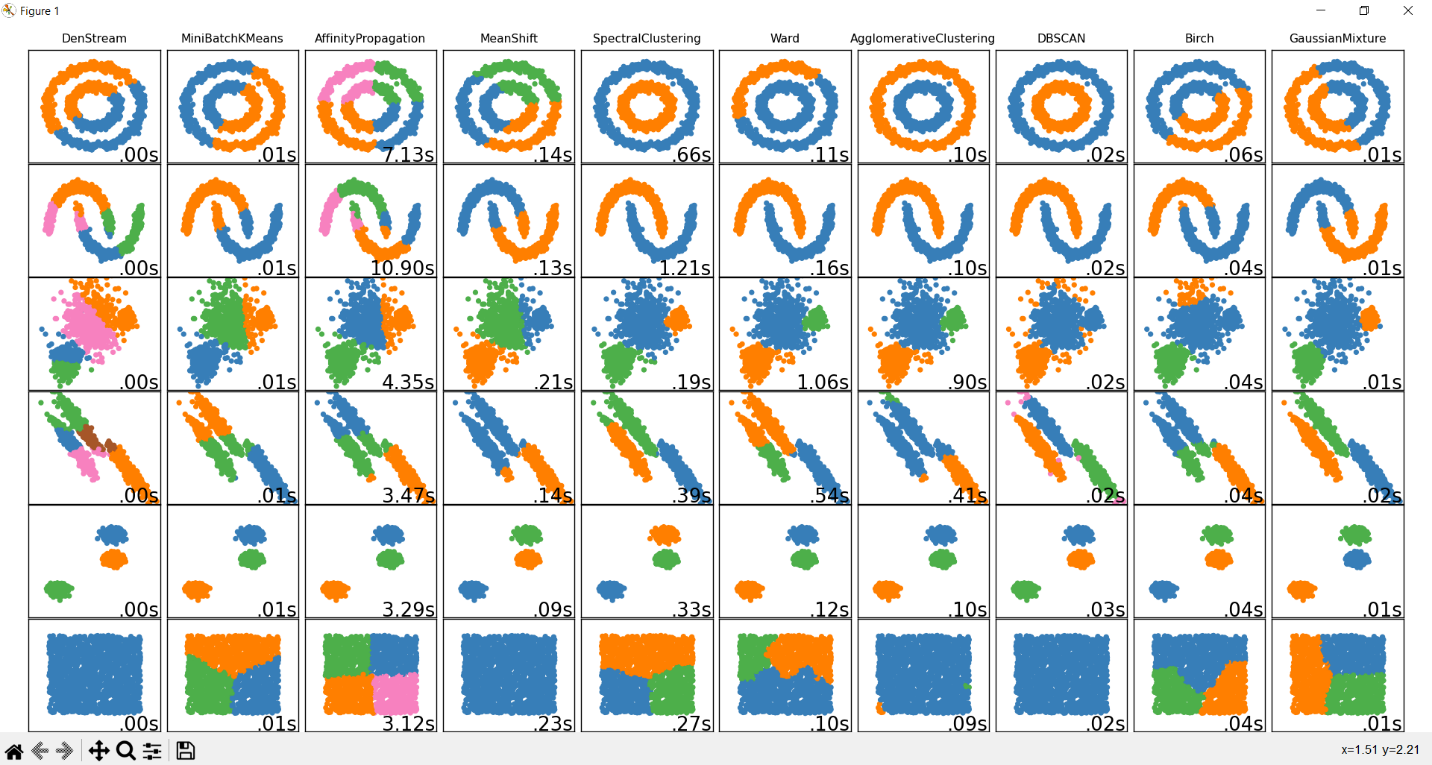


Figure 3 مقایسه DenStream با روش‌های دیگر با Cpu CoreI9-12900k

## بحث

تحلیل نتایج به دست آمده، بحث در مورد مزایا و معایب DenStream، ارتباط آن با کاربردهای مختلف و چگونگی تأثیر آن بر این کاربردها. همچنین، پیشنهادات برای بهبود یا توسعه‌ی آینده‌ی این الگوریتم. همچنین برای بهتر بررسی کردن و دیباگ کردن قسمت های مختلف این روش یک روش واقعی تر روی نقشه واقعی پیاده سازی شده که کامل سرویس و کلاینت در زبان های python و C++ به ترتیب پیاده‌سازی شده است که کدهای منابع و تمامی این کارها به صورت متن باز قابل مشاهده می‌باشد:

<https://github.com/SC-One/ClusteringDataStream>

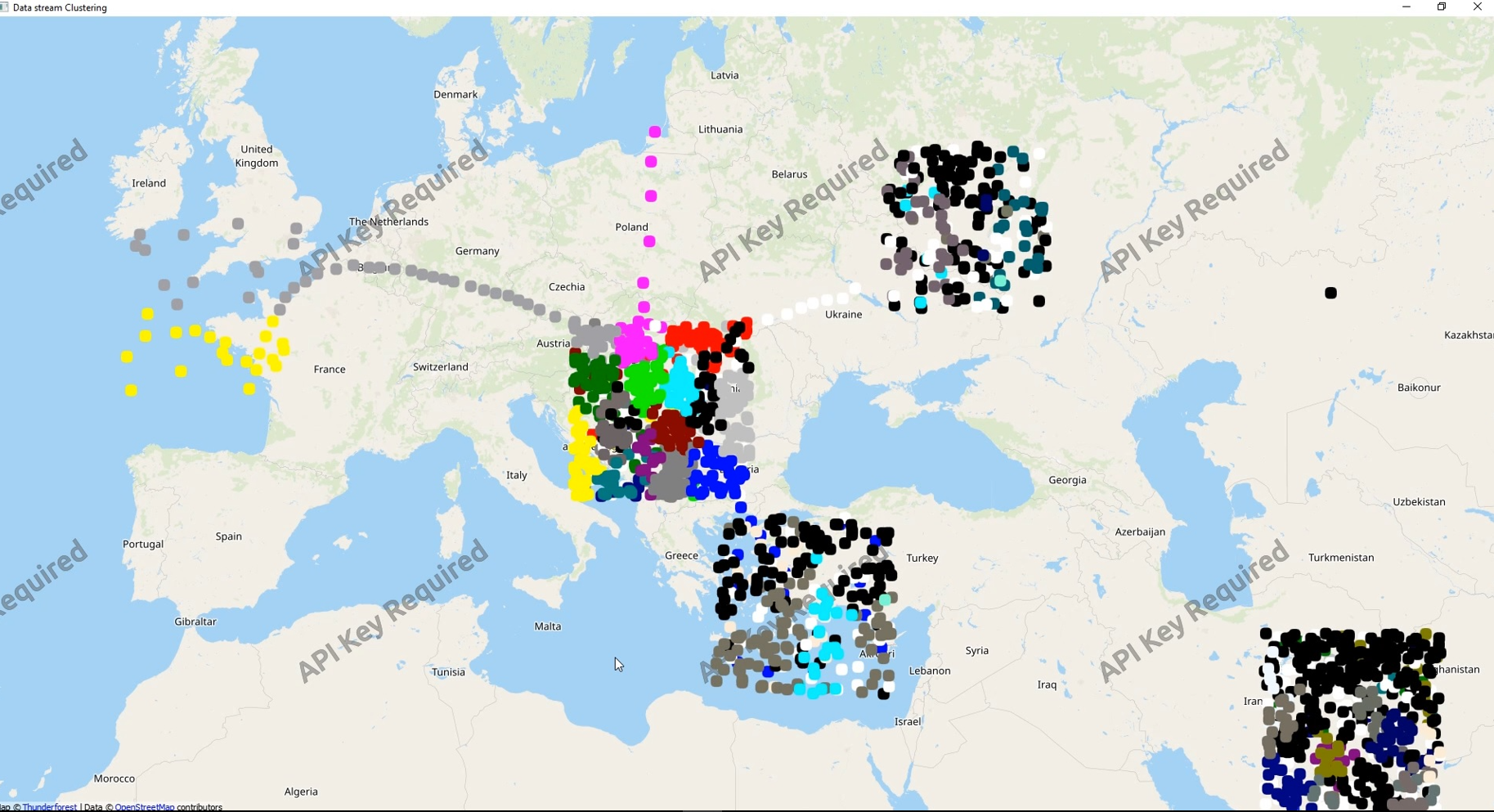


Figure 4 پیاده سازی یک تست کامل برای این روش

## نتیجه‌گیری

در این مقاله، روش DenStream را به عنوان یک روش موثر و کارآمد برای خوشه‌بندی جریان‌های داده در حال تغییر، معرفی کردیم. این روش قادر است خوشه‌هایی با اشکال دلخواه در جریان‌های داده را شناسایی کند و به نویز حساس نیست. ساختارهای p-میکرو-خوشه‌ها و o-میکرو-خوشه‌ها اطلاعات کافی برای خوشه‌بندی را حفظ می‌کنند و یک استراتژی قطعه‌بندی نوآورانه طراحی شده است تا مصرف حافظه را با ضمانت دقت محدود کند. ارزیابی عملکرد تجربی ما بر روی تعدادی از مجموعه‌های داده واقعی و مصنوعی، کارایی و کارآیی DenStream را در کشف خوشه‌های با اشکال دلخواه در جریان‌های داده نشان می‌دهد. کارهای آینده شامل شناسایی خوشه‌ها با اشکال دلخواه در چند سطح دقت، تطبیق پویا پارامترها در جریان‌های داده و بررسی چارچوب ما برای تشخیص پرت و خوشه‌بندی بر مبنای چگالی در مدل‌های دیگر جریان‌ها، به ویژه در مدل پنجره لغزان، می‌باشد..

## منابع

[1] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In Proc. of VLDB, 2003.

[2] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. Incremental clustering for mining in a data warehousing environment. In Proc. of VLDB, pages 323–333, 1998.

[3] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for projected clustering of high dimensional data streams. In Proc. of VLDB, 2004.