مجله علمی-پژوهشی
رایانش نرم و فن آوری اطلاعات
جلد ۴، شماره ۴، سال ۱۳۹۴

# ارائه الگوریتم جدید جهت کشف داده‌های پرت محلی در جریان داده‌ها

آرش مزیدی[۱]، محمدهادی صدرالدینی[۲] و هومان تحیری[۳]
[۱]دانشگاه شیراز، mazidi@cse.shirazu.ac.ir
[۲]دانشگاه شیراز، sadredin@shirazu.ac.ir
[۳]دانشگاه شیراز، tahayori@shirazu.ac.ir

**چکیده - افزایش روزافزون داده‌ها در پایگاه داده‌ها، نیاز به روش‌های بهینه برای آنالیز داده‌ها را افزایش داده است. بیشتر مطالعات، بر روی پیداکردن الگوهای کاربردی در پایگاه داده‌ها متمرکز شده‌اند. این مطالعات برای کاربردهای تشخیص فعالیت مجرمین در تجارت الکترونیک و تشخیص انحرافات نسبت به کاربردهای دیگر مفیدتر واقع شده است. الگوریتم‌های زیادی برای تشخیص داده‌های پرت ارائه شده است، اما اکثر این الگوریتم‌ها بر روی داده‌های ایستا کارایی دارند. داده‌های جریانی، داده‌های پیوسته و نامحدودی هستند که در طول زمان، تغییر توزیع خواهند داشت. این تغییر توزیع، باعث افزایش نرخ مثبت- کاذب و عدم کارایی الگوریتم‌های موجود می‌شود. در این مقاله، الگوریتمی جهت شناسایی داده های پرت، با استفاده از روش تقسیم جریان داده‌ها به قطعه‌های مساوی و محاسبه ضریب ناهنجاری محلی برای داده‌ها و استفاده از لیستی برای داده‌های پرت کاندید ارائه داده ایم تا علاوه بر شناسایی داده‌های پرت، نرخ مثبت-کاذب پایینی داشته باشد. نتایج به‌دست آمده بر روی مجموعه داده‌های مصنوعی و حقیقی، نشان می‌دهد که الگوریتم ارائه شده، باعث کاهش قابل توجه نرخ مثبت-کاذب و افزایش دقت شده و نسبت به الگوریتم‌های دیگر کارایی بهتری دارد.**

**کلید واژه‌ها**-داده‌های پرت، جریان داده‌ها، تشخیص داده‌های پرت، ضریب ناهنجاری محلی.

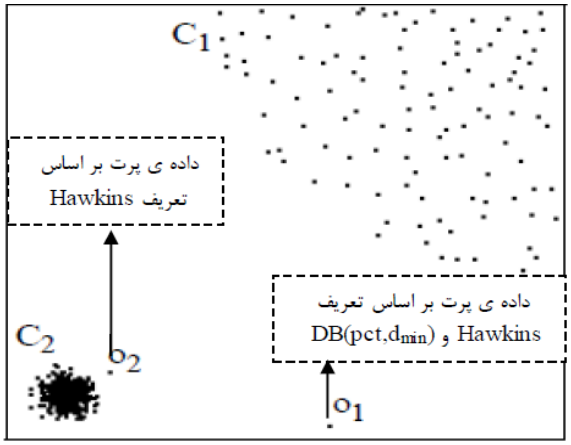
## ۱. مقدمه

امروزه، با پیشرفت روزافزون فناوری اطلاعات، تعداد پایگاه داده‌ها و همچنین خودکارسازی سیستم‌ها افزایش یافته است. داده‌های زیادی در حوزه‌های کاری مختلف، در کامپیوترها ذخیره می‌شوند. اما این داده‌های خام، به تنهایی مفهوم خاصی را نخواهند داشت و ما باید از این داده‌ها دانش و اطلاعات مفید را استخراج کنیم. به استخراج دانش از داده‌های خام، داده‌کاوی گفته می‌شود. داده‌کاوی کاربردهای زیادی مانند خوشه‌بندی‌ها، طبقه‌بندی داده‌ها و کشف وابستگی بین داده‌ها در حوزه‌های مختلف دارد. همچنین یکی دیگر از کاربردهای داده‌کاوی، کشف داده‌های پرت و الگوهای نامتعارف در مجموعه داده کاربرد مورد نظر است.

بر اساس تعریف Hawkins (1980)، داده‌ای که انحراف زیادی نسبت به داده‌های دیگر درون پایگاه داده دارد، داده پرت است. این انحراف به قدری است که به نظر می‌رسد که این داده با مکانیزم جدیدی تولید شده است [۱].

همچنین برطبق تعریف دیگری که به عنوان $DB(pct,d_{min})$ معروف است، زمانی که حداقل pct درصد از شی‌های پایگاه داده، دارای فاصله‌ای بیشتر از $d_{min}$ از داده p (داده p یک داده از مجموعه داده است که پرت بودن آن را بررسی می‌کنیم) باشند، داده p، پرت خواهد بود [۲]. در شکل ۱، نمونه‌هایی از داده‌های نرمال و پرت نشان داده شده است.

شکل ۱، پایگاه داده دو بعدی با دو خوشه $C_1$ و $C_2$ که به ترتیب شامل ۴۰۰ و ۱۰۰ شی هستند، را نشان می‌دهد. همچنین دو شی $o_1$ و $o_2$ در پایگاه داده وجود دارند. در این مثال خوشه $C_2$ متراکم‌تر از خوشه $C_1$ است. بر اساس تعریف Hawkins داده‌های $o_1$ و $o_2$ هر دو شی‌های پرت شناخته می‌شوند، درحالی که شی‌هایی که در خوشه‌های $C_1$ و $C_2$ قرار دارند، نمی‌توانند شی‌های پرت به‌شمار آیند.




شکل ۱: مثالی ساده از داده‌های متعارف و پرت در مجموعه داده دو بعدی [۳]

از طرفی انتظار می‌رود که شی‌های $o_1$ و $o_2$ به عنوان شی‌های پرت محلی شناخته شوند. اما بر اساس الگوریتم‌های مبتنی بر فاصله مانند DB(pct,$d_{min}$)، تنها شی $o_1$ یک داده پرت می‌باشد. به‌طور خلاصه، پدیده نامتعارف، به الگو یا الگوهایی در داده‌ها گفته می‌شود که با رفتار مورد انتظار از داده‌ها مطابقت نداشته باشند. بنابراین، جهت ارائه یک راهکار مشخص و ساده برای تشخیص این پدیده‌ها، می‌بایست ناحیه‌ای از فضای اطلاعاتی را ناحیه نرمال تعریف کرده و هر داده ای که در این محدوده قرار نگیرد را غیر نرمال یا پرت بنامیم [۴]. اما مشکلات زیادی این تعریف به ظاهر ساده را بسیار پیچیده و مشکل‌زا می‌کند که این مشکلات عبارتند از:

- تعریف یک مرز مشخص همیشگی و ثابت میان رفتار نرمال و غیر نرمال کاری بسیار سخت و در بسیاری از شرایط غیرممکن است [۴]. به‌عبارتی دیگر تعریف رفتار نرمال، بر اساس حوزه کاربرد آن، متغیر است [۵].
- در بسیاری از کاربردها مانند سیستم‌های تشخیص نفوذ، رفتار نرمال داده‌ها در حال تغییر است، بنابراین ممکن است این تغییرات، به اشتباه به‌عنوان داده‌های غیرنرمال شناسایی شوند [۴].
- از آنجا که پدیده های نامتعارف نتیجه رفتارهای بداندیشانه هستند، افراد بداندیش رفتار مخرب خود را تا حد ممکن به رفتار نرمال و معمول سیستم‌ها نزدیک می‌کنند. از این‌رو تشخیص چنین رفتارهایی بسیار دشوار می‌شود [۴].
- برچسب داده‌ها در همه کاربردها در دسترس نبوده و تشخیص را دشوار می کند [۵].
- تشخیص نویز از داده های پرت بسیار مهم و دشوار است. نویزها باید شناسایی شده و از مجموعه داده حذف شوند تا باعث کاهش دقت الگوریتم نشوند [۴].

از آنجایی که سازمان‌ها به‌صورت مداوم در حال رشد هستند، پایگاه داده‌های آنها نیز به همان نسبت در حال رشد هستند. درنتیجه، تکنیک‌های داده‌کاوی قدیمی قادر به پوشش پایگاه داده‌های در حال رشد آنها نیستند و با شکست روبرو می‌شوند. بزرگ و پویا بودن، طبیعت پایگاه داده‌ها است و تغییر در آنها به‌طور معمول اتفاق می‌افتد. داده‌های جدید به پایگاه داده اضافه می‌شوند و داده‌های قدیمی حذف یا اصلاح می‌شوند. با توجه به حجم زیاد داده‌ها، سیستم قادر به ذخیره سازی همه داده‌ها نخواهد بود و از طرفی دیگر، داده‌های قدیمی پس از مدتی اعتبار و ارزش خود را از دست داده و برای افزایش کارایی و همچنین افزایش سرعت در فرایندهای سیستم، داده های قدیمی حذف می‌شوند. داده‌ها دارای برچسب زمانی می‌شوند و برحسب این برچسب، زمان ورود داده‌ها به سیستم مشخص می‌شود. افزایش حجم داده‌ها و اضافه شدن دائمی داده‌ها، عدم توانایی در مدیریت و ذخیره داده‌ها در حافظه و سیستم‌های محدود و همچنین جریانی و متوالی بودن داده‌های ورودی، باعث به‌وجود آمدن نوع داده جدیدی به نام داده جریانی شده است.

مشخصه‌های اصلی داده‌های جریانی که باعث تمایز بین داده‌های جریانی از داده‌های ایستا می‌شوند، عبارتند از:

- داده‌ها در جریان، غیر ثابت و نامحدود هستند.
- تغییر در توزیع داده‌ها به‌طور دائم ممکن است رخ دهد.
- معمولاً فقط یک بار می‌توان آنها را خواند.
- هر سیستمی که بر روی آن‌ها کار می‌کند، در ترتیب و نحوه ورود آن‌ها به سیستم کنترلی ندارد.

در ادامه این مقاله، در بخش ۲، کارهای مرتبط انجام شده در حوزه تشخیص داده‌های پرت، در بخش ۳ مفاهیم الگوریتم پایه جهت کشف داده‌های پرت، در بخش ۴ الگوریتم ارائه شده در مقاله و در بخش ۵ پیاده سازی و ارزیابی انجام شده بر روی مجموعه داده‌های مختلف آورده شده است و در نهایت، در بخش۶، به نتیجه گیری و پیشنهادات خود برای کارهای آینده خواهیم پرداخت.

## ۲. کارهای مرتبط

در مقاله [۶]، مرور و بررسی الگوریتم‌های موجود جهت کشف انحرافات و داده‌های پرت در داده‌های وابسته به زمان و مکان

انجام شده است. کاربردهای مختلف داده‌های وابسته به زمان و الگوریتم‌های مختص هر کاربرد در این مقاله دسته بندی شده اند. در مقاله [۷] نوع جدیدی از الگوریتم خوشه بندی K-Means ارائه شده است. این الگوریتم، به داده‌ها وزنی را تخصیص می‌دهد و برای داده‌های جریانی پویا مناسب است. وزن‌های تخصیص داده شده، اهمیت هر داده در میان مجموعه داده‌ها را نشان می‌دهد. از این الگوریتم برای تشخیص داده‌های پرت در جریان داده‌ها استفاده می‌شود.

در مقاله [۲]، الگوریتمی جهت شناسایی داده‌های پرت ارائه شده است. در این الگوریتم، زمانی که حداکثر π درصد از همه داده‌های مجموعه داده، فاصله ای کمتر از ε از یک داده داشته باشند، آن داده، داده پرت شناخته می‌شود.

الگوریتم‌هایی جهت تشخیص داده‌های پرت مبتنی بر فاصله ارائه شده است. در مقاله [۸] یکی از این الگوریتم‌های مبتنی بر فاصله ارائه شده است. این الگوریتم، فاصله از همسایه k ام داده مورد نظر را به‌عنوان امتیاز برای داده در نظر می‌گیرد و همچنین در مقاله [۹]، با در نظر گرفتن مجموع تمامی فاصله‌های همسایه اول تا همسایه k ام از داده مورد نظر، به داده امتیازی داده می‌شود و الگوریتم‌ها بر اساس امتیازهای تخصیص داده شده به داده‌ها، داده های پرت را شناسایی می‌کنند. یکی از الگوریتم‌های معتبر که مبتنی بر فاصله بوده و در جریان داده‌ها مورد استفاده قرار می‌گیرد، الگوریتم DBOD-DS می‌باشد. این الگوریتم بر اساس یک تابع احتمال چگالی برای توزیع داده‌ها، تشخیص داده‌های پرت را انجام می‌دهد [۱۰].

در مقاله [۱۱]، روش جدیدی جهت کشف داده‌های پرت ارائه شده است که مبتنی بر معکوس نزدیک‌ترین همسایه‌های داده می‌باشد. این روش بر روی جریان داده‌ها ارائه شده است و از مدل پنجره‌ی لغزان برای پردازش جریان داده‌ها استفاده می‌کند.

در مقاله [۱۲]، روشی جهت کشف داده‌های پرت با توجه به کل مجموعه داده (داده‌های پرت عمومی) و همسایه‌های داده‌ها (داده‌های پرت محلی) ارائه شده است. این روش مبتنی بر محتوی عمومیGDF و محتوی محلیLDF عمل می‌کند. GDF، انحراف داده را نسبت به کل مجموعه داده و LDF، انحراف داده را نسبت به داده‌های تازه وارد شده، نشان می‌دهد. هر دو فاکتور بر اساس چگالی همسایه‌ها محاسبه می‌شوند. در این الگوریتم، داده‌ای که مقدار GDF یا LDF آن، بیشتر از سه برابر انحراف معیار از میانگین باشد، داده پرت خواهد بود. با استفاده از سه برابر انحراف معیار، نیازی به تعیین مقداری توسط کاربر به‌عنوان مقدار حد آستانه برای تشخیص داده پرت، نمی‌باشد.

در مقاله [۱۳]، الگوریتمی جهت تشخیص داده‌های پرت در داده‌های جریانی تراکنش‌ها ارائه شده است. در این مقاله، مدلی افزایشی به‌منظور شناسایی داده های پرت به صورت خودکار ارائه شده است. نتایج حاصل از الگوریتم ارائه شده، نشان داده است که این الگوریتم به‌صورت بهینه و خودکار داده‌های پرت را با دقت قابل قبولی شناسایی کرده است.

مقاله [۱۴]، تشخیص داده‌های پرت در کاربرد کشف اسپم‌های توییتر را بررسی می‌کند. الگوریتم ارائه شده یک الگوریتم خوشه‌بندی است، در حالی که روش‌های پیشین در این زمینه، الگوریتم‌های طبقه‌بندی بوده‌اند. در الگوریتم ارائه شده از دو الگوریتم خوشه‌بندی DenStream و StreamKM++ که در مقاله‌های [۱۵، ۱۶] ارائه شده‌اند، استفاده شده است و نتایج آزمایشات، دقت ۱۰۰٪ را نشان داده است.

در مقاله [۱۷]، الگوریتمی مبتنی بر پیش‌بینی چگالی هسته (KDE) بر روی داده‌های توزیع شده جریانی جهت کشف داده‌های پرت در شبکه‌های حسگر بیسیم ارائه شده است. الگوریتم ارائه شده از پنجره لغزان جهت پردازش داده‌های جریانی استفاده می‌کند و به داده‌ها وزنی را تخصیص می‌دهد. نتایج الگوریتم، کارایی مناسب الگوریتم ارائه شده در محیط‌های بلادرنگ و داده‌های جریانی را نشان می‌دهد.

## ۳. تعاریف و مفاهیم الگوریتم پایه

الگوریتم ضریب ناهنجاری محلی، الگوریتم مناسبی برای تشخیص داده‌های پرت محلی در داده‌های ایستا می‌باشد [۳]. دراین الگوریتم، امتیازی (ضریب ناهنجاری محلی) به‌عنوان درجه پرتی داده‌ها در پایگاه داده محاسبه می‌شود که نشان دهنده پرتی یا نرمالی داده است. این فاکتور به‌صورت محلی محاسبه شده و بر اساس همسایه‌های هر شی به‌دست می‌آید. مقدار ضریب ناهنجاری محلی داده‌های نرمال برابر یک و داده‌های پرت، فراتر از یک خواهد بود. مقدار ضریب ناهنجاری محلی فقط به یک مقدار MinPts (تعداد همسایه های نزدیک) وابسته است. ضریب ناهنجاری محلی (فاکتور میزان پرتی یا امتیازی که به هر شی تخصیص داده می‌شود)، با استفاده از معادله ۱ (برای شی p)، محاسبه می‌شود [۳].

$$LOF_{MinPts}(p) = \frac{\sum_{o \epsilon N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|} \qquad (۱)$$

Lrd میزان دسترسی چگالی محلی را نشان می‌دهد [۳]. این مقدار برابر میانگین نسبت دسترسی چگالی محلی شی p و شی-های همسایه نزدیک p می‌باشد. همچنین MinPts تعداد همسایه‌های نزدیک به شی را نشان می‌دهد. $N_{MinPts}$ مجموعه شی‌های همسایه نزدیک به شی مورد نظر را که تعداد آنها با MinPts نشان داده می‌شد، مشخص می‌کند. زمانی که مقدار به-دست آمده از معادله ۱، برابر یک باشد، داده در محلی با چگالی و تراکم بالا قرار دارد و به‌عنوان داده نرمال شناخته می‌شود و در صورتی که مقداری فراتر از یک داشته باشد، چگالی اطراف داده کم بوده و احتمال پرتی داده، افزایش می‌یابد.

دو مزیت اصلی که در الگوریتم ضریب ناهنجاری محلی وجود دارد، عبارتند از:

۱) الگوریتم داده‌های پرت را با توجه به چگالی همسایه‌های شی شناسایی کرده و مدلی به‌صورت عمومی ندارد.
۲) الگوریتم قادر به شناسایی داده‌های پرت در هر توزیع داده-ای است و هیچ پیش فرضی را برای توزیع داده‌ها درنظر ندارد.

با استفاده از ایده الگوریتم ضریب ناهنجاری محلی برای داده‌ها، الگوریتم ضریب ناهنجاری محلی افزایشی ارائه شده است [۱۸]. این الگوریتم، جهت کشف داده‌های پرت در جریان داده‌ها مناسب می‌باشد. دراین الگوریتم، برای پردازش داده‌های جریانی، از پنجره لغزان استفاده شده است. همچنین، مقدار ضریب ناهنجاری برای هر داده، در زمان ورود آن داده به پنجره لغزان، محاسبه شده و تشخیص پرتی یا نرمالی داده ورودی انجام می-شود. ورود داده جدید به پنجره لغزان، باعث ایجاد تغییر در همسایه‌های داده‌های موجود در پنجره شده که نیاز به به‌روز رسانی همسایه‌ها، فاصله‌ها و در نهایت ضریب ناهنجاری محلی داده‌ها می‌باشد. این الگوریتم در دو گام ورود داده و خروج داده از پنجره لغزان انجام می‌شود. به‌طور کلی، در زمان ورود و خروج هر داده، پارامترهای داده‌های همسایه که متاثر از ورود یا خروج داده‌ای هستند، باید به‌روز رسانی شوند.

مهمترین مزایای استفاده از الگوریتم ضریب ناهنجاری محلی افزایشی درمقابل سایر روش‌های تشخیص الگوهای نامتعارف را می‌توان به‌ترتیب زیر نام برد:

- الگوریتم این توانایی را دارد که الگوهای نامتعارف را نسبت به چگالی همسایه‌های هر نقطه تشخیص دهد. بنابراین هر موجودیت را با تمام نقاط درون مجموعه داده مقایسه نمی-کند، که این کار زمانبر و غیر مفید خواهد بود.
- الگوریتم توانایی تشخیص الگوهای نامتعارف را بدون در نظر گرفتن توزیع داده‌ها دارد، چرا که این روش هیچ پیش فرضی را در مورد توزیع داده‌های ورودی در نظر نمی‌گیرد.
- الگوریتم ضریب ناهنجاری محلی افزایشی، هر نوع تغییر در توزیع داده‌ها را بلافاصله و به‌صورت بر خط تشخیص می-دهد.
- الگوریتم برای داده‌های جریانی مناسب است و هر داده را تنها در یک گذر تحلیل و آنالیز می‌کند، در نهایت به هر نمونه، یک درجه از میزان نامتعارفی، انتساب می‌کند.

تمام الگوریتم‌ها، در کنار مزایایی که ارائه می‌دهند، دارای معایبی هستند که محققان در تلاش برای رفع معایب الگوریتم‌ها هستند. الگوریتم ضریب ناهنجاری محلی افزایشی نیز دارای معایبی است که به آنها اشاره می‌شود.

- الگوریتم ضریب ناهنجاری محلی افزایشی یک الگوریتم کارا و مؤثر جهت تشخیص تغییرات در رفتار و توزیع داده‌ها است. اما این الگوریتم در تشخیص الگوهای نامتعارف، دچار تشخیص‌های مثبت-کاذب فراوانی می‌شود و هر نوع تغییر جدید در رفتار داده‌ها را به‌عنوان الگوی نامتعارف شناسایی می‌کند. در حالی که بسیاری از این تغییرات، مربوط به ایجاد رفتار جدیدی در داده‌ها خواهند بود و الگوی پرت و نامتعارفی را نشان نمی‌دهند.
- الگوریتم ضریب ناهنجاری محلی افزایشی، دارای دو بخش ورود داده‌های جدید و حذف داده‌های قدیمی است. نگه داشتن و یا حذف پدیده‌های نامتعارف در حافظه، بر تحلیل داده‌هایی که در طول زمان وارد می‌شوند بسیار تأثیرگذار است.
- داده‌های جریانی، حجم بسیار بالایی از داده‌ها هستند که به صورت پیوسته وارد سیستم می‌شوند. از آنجا که در الگوریتم افزایشی، چگالی هر نمونه داده نسبت به سایر همسایه‌های خود محاسبه می‌شود، همه داده‌ها باید در حافظه قرار بگیرند، در حالی که حافظه بسیار محدود است. بنابراین حذف داده‌های قدیمی برای آزادسازی فضای حافظه و نیز برای افزایش دقت سیستم بسیار ضروری است.

الگوریتم دیگری که در مقاله [۱۹] ارائه شده است، الگوریتم بهبود یافته ضریب ناهنجاری محلی افزایشی است. در این الگوریتم، تشخیص الگوهای نامتعارف را از تشخیص توزیع جدید جریان داده متمایز می‌سازد. این رویکرد سبب کاهش نرخ تشخیص‌های مثبت-کاذب می‌شود. استراتژی حذف الگوهای

نامتعارف نیز به این الگوریتم افزوده شده است که دقت آن را در تشخیص پدیده‌های نامتعارف افزایش می‌دهد. همچنین استراتژی حذف داده‌های قدیمی در این الگوریتم به گونه‌ای تغییر یافته است که سرعت الگوریتم در حذف این الگوها بسیار افزایش یافته است.

برای رفع مشکلات موجود در الگوریتم ضریب ناهنجاری محلی افزایشی و بهبود یافته و با هدف کاهش نرخ مثبت-کاذب و افزایش دقت، الگوریتم جدیدی جهت کشف داده‌های پرت در جریان داده‌ها ارائه داده‌ایم که در ادامه، الگوریتم پیشنهادی تشریح شده است.

## ۴. الگوریتم پیشنهادی

الگوریتم ضریب ناهنجاری محلی افزایشی، به هر نمونه داده‌ای که وارد سیستم می‌شود، ضریب یا امتیازی از درجه پرتی، تخصیص می‌دهد. با ورود هر داده جدید، پارامترهای یک سری از نقاط تأثیر می‌پذیرند. از این رو، ضریب ناهنجاری محلی داده‌ها، در هر مرحله ورود داده جدید به سیستم، به صورت پویا امکان به‌روزرسانی دارند. نمونه داده‌هایی که در ابتدای ورود به سیستم در نواحی با تراکم پایین قرار می‌گیرند، دارای ضریب ناهنجاری محلی بالاتری خواهند بود. بنابراین، این نقاط در ابتدا با احتمال بالایی الگوی ناهنجار و نامتعارف تشخیص داده خواهند شد. اما با مرور زمان و با قرار گرفتن داده‌های جدید در نزدیکی این نمونه داده، داده‌ها یک خوشه متمرکز و با چگالی بالا تشکیل خواهند داد. بنابراین، با ورود داده‌های جدید و با افزایش چگالی داده‌ها، ضریب ناهنجاری محلی داده‌ها به‌روز شده و کاهش می‌یابد. با کمک مثالی که در ادامه بیان می‌شود، می‌توان این موضوع را به سادگی نشان داد. در این مثال از یک مجموعه داده مصنوعی با توزیع نرمال، دارای ۶۰۰ نمونه داده که با نرم افزار متلب تولید شده است، استفاده کرده‌ایم. این داده‌ها، از ترکیب دو سری داده دو بعدی، با توزیع‌های نرمال برابر اما میانگین‌های متفاوت ایجاد شده‌اند. هریک از این دو سری داده، دارای ۳۰۰ نمونه داده هستند. مجموعه داده‌ها با توزیع نرمال، دارای ویژگی‌های زیر می‌باشند.

سری داده اول: $N_1(\mu_1,\Sigma_1)$ ; $\mu_1= [+1,+1]$ ; $\Sigma_1= \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix}$

سری داده دوم: $N_2(\mu_2,\Sigma_2)$ ; $\mu_2= [+8,+8]$ ; $\Sigma_2= \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix}$

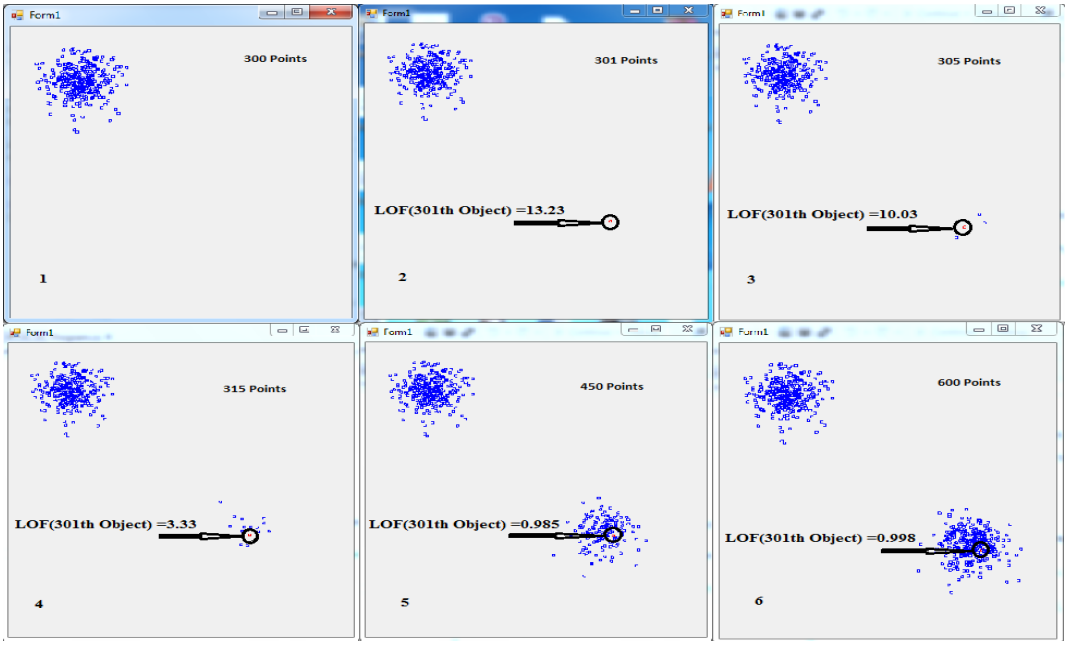
در شکل ۲، روند تکاملی ورود جریان داده‌های مجموعه داده تولید شده، نشان داده شده است. در این شکل، ابتدا ۳۰۰ داده متعلق به خوشه (سری) داده اول ( بالا سمت چپ شکل ۲) وارد شده‌اند و پس از آن، داده شماره ۳۰۱ ( داده ۳۰۱ ام، با رنگ قرمز و دایره مشکی نشان داده شده و مقدار LOF برای آن در شکل ۲ نوشته شده است) وارد می‌شود و مقدار ضریب ناهنجاری محلی آن همان‌طور که در شکل ۲ نشان داده شده است، برابر ۱۳,۲۳ محاسبه شده و همین روند را پس از ورود ۳۰۵، ۳۱۵، ۴۵۰ و ۶۰۰ نمونه داده، ادامه داده‌ایم و مقدار ضریب ناهنجاری محلی برای نمونه داده ۳۰۱ ام، به ترتیب برابر ۱۰,۰۳، ۳,۳۳، ۰,۹۸۵ و ۰,۹۹۸ محاسبه شده است. الگوریتم ضریب ناهنجاری محلی برای این مثال با مقدار k=20 (k تعداد همسایه‌های نزدیک به داده می‌باشد) اجرا شده است. تاثیر اندازه k بر روی الگوریتم و کشف داده‌های پرت، در زیر بخش ۴,۱ ارائه شده است.

همان‌طور که در شکل ۲ مشاهده شد، پس از ورود نمونه داده شماره ۳۰۱، مقدار ضریب ناهنجاری محلی آن ۱۳,۲۳ بوده و با توجه به الگوریتم ضریب ناهنجاری محلی افزایشی، این نمونه داده به‌عنوان داده پرت شناسایی می‌شود. اما همان‌طور که در سیر تکاملی جریان داده‌ها نشان داده شده است، مقدار ضریب ناهنجاری محلی برای نمونه داده شماره ۳۰۱، با ورود داده‌های بعدی و افزایش تراکم در اطراف این داده، کاهش یافته و به مقدار یک (مقدار داده نرمال)، رسیده است.

یکی از مشکلات الگوریتم ضریب ناهنجاری محلی افزایشی، عدم شناخت الگوی جدید بوده و هر الگوی جدید را الگویی نامتعارف می‌شناسد که این مشکل در شکل ۲ نشان داده شده است.

رویکرد الگوریتم ضریب ناهنجاری محلی افزایشی، بدین صورت است که پس از ورود نمونه داده، مقدار ضریب ناهنجاری محلی محاسبه شده و در صورتی که مقدار محاسبه شده، از حد آستانه بیشتر باشد، به‌عنوان داده پرت شناسایی می‌شود.

با توجه به مشکلات مشاهده شده در الگوریتم‌های پیشین، الگوریتمی جهت کشف داده‌های پرت محلی در جریان داده‌ها در این مقاله ارائه شده است.




شکل ۲: روند ورود داده‌ها و محاسبه مقدار ضریب ناهنجاری محلی

با توجه به اینکه داده‌ها به‌صورت جریانی هستند، الگوریتم نیاز به ویژگی‌های خاصی همچون سرعت بالا و برخورد مناسب با تغییر توزیع در جریان داده‌ها دارد. در شکل ۳، شبه کد الگوریتم ارائه شده، نشان داده شده است.

```
1.  Procedure Chunk LOF
2.  Begin
3.      Set size of chunk
4.      Set threshold for outlier
5.      Set threshold_count for counter
6.      Repeat
7.          Fill chunk by objects
8.          Calculate LOF for objects in chunk
9.          For each object that LOF(object) > threshold
10.             Add object to outliers
11.         Candidate_outliers = outliers
12.         Candidate_outliers_count ++
13.         Add candidate_outliers to Next Chunk
14.         If(candidate_outliers_count=threshold_counter)
15.          Then
16.             Real_outlier = candidate_outlier
17.             Delete real_outlier from candidate_outlier
18.             Set candidate_outliers_count to zero
19.         End IF
20.         Counter += size of chunk
21.     Until(counter < dataset_size)
22.     Show Real_outlier
23. End
```

شکل ۳: شبه کد الگوریتم پیشنهادی تشخیص داده‌های پرت

در الگوریتم ارائه شده، از ساختمان داده صف استفاده شده و داده‌ها وارد سیستم می‌شوند. با توجه به اینکه جریان داده‌ها نامحدود هستند، حافظه کافی برای ذخیره همه داده‌ها در سیستم وجود ندارد و برای رفع این مشکل، جریان داده‌ها را قطعه‌بندی کرده و داده ها به صورت قطعه قطعه وارد سیستم می‌شوند. اندازه قطعه و حدهای آستانه در ابتدای الگوریتم تعیین می شوند (خطوط ۳، ۴ و ۵ در شکل ۳). پس از تکمیل قطعه‌ای از داده‌ها ( خط ۷ در شکل ۳)، الگوریتم ضریب ناهنجاری محلی بر روی داده‌های درون قطعه اجرا شده و با توجه به حد آستانه در نظر گرفته شده برای الگوریتم ضریب ناهنجاری محلی، داده-های پرت شناسایی می‌شوند (خطوط ۸ و ۹ در شکل ۳). اما این داده‌های پرت شناسایی شده را به‌عنوان داده پرت معرفی نکرده و آنها را در لیستی با عنوان لیست داده‌های پرت کاندید قرار می-دهیم ( خط ۱۱ در شکل ۳). لیست کاندید شامل داده‌های پرتی است که در قطعه‌های مختلف جریان داده‌ها شناسایی شده‌اند. همچنین برای هر یک از نمونه داده‌ها یک شمارنده با مقدار اولیه صفر در نظر می‌گیریم و هر بار که داده‌ای وارد لیست کاندید شد، به شمارنده آن داده، یک واحد اضافه می‌کنیم (خط ۱۲ در شکل ۳).

درگام بعدی، داده‌هایی که در لیست کاندید پرت، قرار دارند را ابتدا به قطعه وارد کرده (خط ۱۳ در شکل ۳) و سپس، داده‌های جدید را وارد قطعه می‌کنیم و مراحل محاسبه ضریب ناهنجاری

محلی را برای داده‌های درون قطعه تکرار می‌کنیم. اما قبل از تکرار مراحل الگوریتم، شمارنده‌ای که برای داده ها جهت شمارش تکرار پرتی داده‌ها در قطعه های مختلف در نظر گرفته بودیم را با حد آستانه آن مقایسه کرده و اگر اندازه شمارنده به اندازه حد آستانه رسیده باشد، آن داده را به‌عنوان داده پرت واقعی شناسایی کرده و از لیست کاندید حذف شده و شمارنده آن صفر می‌شود (خطوط ۱۸-۱۳ شکل ۳).

زمانی که داده‌های پرت کاندید، در کنار داده‌های بعدی که از جریان داده‌ها وارد قطعه می‌شوند، قرار می‌گیرند و ضریب ناهنجاری محلی آنها محاسبه می‌شود، بدین معنی است که قصد داریم با وارد شدن داده‌های جدید، بررسی کنیم که آیا الگوی جدیدی تولید می‌شود یا اینکه آن داده‌ها هنوز هم پرت هستند و الگوی نامتعارفی را تعریف می‌کنند.

در واقع با این رویکرد، اعلام برچسب داده‌ها را برای چند قطعه به تعویق انداخته تا بتوانیم تمایز بین الگوی جدید و الگوی نامتعارف را تشخیص دهیم. تعیین حد برای شمارنده می‌تواند تاثیر زیادی در شناسایی داده‌های پرت داشته باشد و با توجه به مجموعه داده و کاربرد مورد نظر، میزان تاخیر در اعلام پرت بودن واقعی را به عنوان شمارنده تعیین می‌کنیم.

در نهایت، داده‌هایی که در لیست داده‌های پرت واقعی (Real_outlier) قرار دارند را به عنوان داده‌های پرت شناسایی شده توسط الگوریتم معرفی می‌کنیم (خط ۲۲ در شکل ۳).

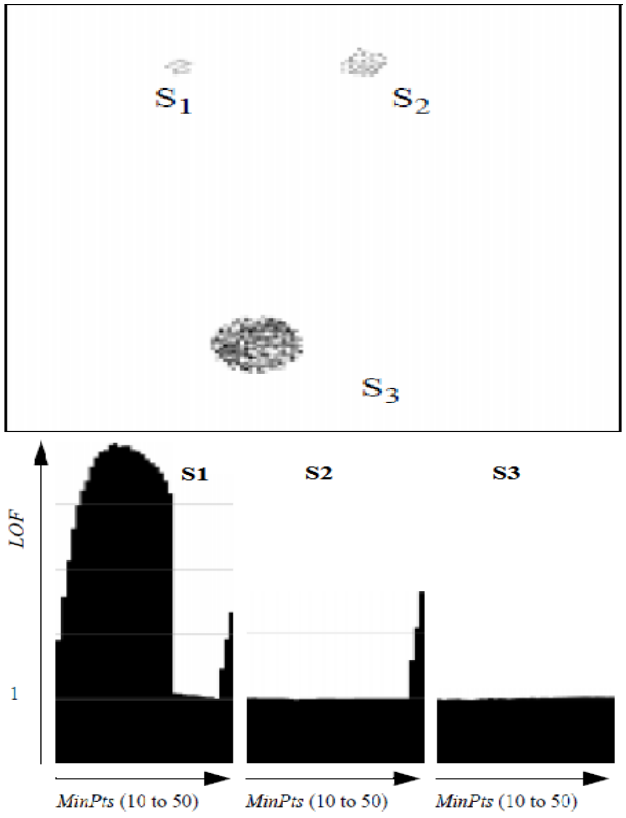
الگوریتم ارائه شده، از مزایای الگوریتم ضریب ناهنجاری محلی افزایشی استفاده می‌کند و سعی در پوشش معایب این الگوریتم دارد. همان‌طور که اشاره شد، نرخ بالای تشخیص مثبت-کاذب، بزرگترین عیب الگوریتم ضریب ناهنجاری محلی افزایشی محسوب می‌شود و دلیل این نرخ بالا، تعیین داده پرت، پس از مشاهده مقدار ضریب ناهنجاری محلی بالا در زمان ورود داده به پنجره لغزان می‌باشد. الگوریتم ارائه شده، با به تعویق انداختن اعلام پرتی داده‌ها و استفاده از روش جدید برای پردازش جریان داده‌ها، نرخ مثبت-کاذب را کاهش داده و باعث بهبود الگوریتم ضریب ناهنجاری محلی افزایشی و الگوریتم ضریب ناهنجاری محلی افزایشی بهبود یافته می‌شود.

برای پردازش داده‌های جریانی، معمولا از روش پنجره لغزان استفاده می‌شود. اما روش پنجره لغزان نیاز به به‌روزرسانی همسایه‌های داده در زمان ورود و خروج داده دارد. در الگوریتم پیشنهادی، روشی جدید جهت پردازش جریان داده ها ارائه شده است که نیاز به به‌روزرسانی‌های روش پنجره لغزان که زمانبر هستند، نمی‌باشد. قطعه کردن جریان داده‌ها، روش مناسبی جهت پردازش این نوع داده‌ها محسوب می‌شود. زمانی که جریان داده‌ها را به قطعه‌های مساوی تقسیم می‌کنیم، تاثیرات هر قطعه را به قطعه‌های بعدی منتقل کرده و تکرار تاثیرات را در قطعه-های مختلف مشاهده و ثبت می‌کنیم. با استفاده از این روش، داده‌های پرتی که در چندین قطعه به‌طور مکرر شناسایی شده‌اند را به‌عنوان داده پرت واقعی شناسایی می‌کنیم.

### ۱،۴. تاثیر پارامتر k (تعداد همسایه‌های نزدیک) در الگوریتم پیشنهادی

در این بخش تاثیر پارامتر k در محاسبه مقدار LOF هر شی را بررسی خواهیم کرد. در بخش‌های دیگری از مقاله، k با عنوان MinPts اعلام شده است که هر دو معنای تعداد همسایه‌های نزدیک یک شی را می‌دهند. اندازه LOF برای شی‌های داده موجود در مجموعه داده، وابسته به فاصله آن شی با همسایه‌های اطرافش می‌باشد. در شکل ۴، تاثیر پارامتر MinPts بر روی اندازه LOF نشان داده شده است.

همان‌طور که در شکل ۴، مشاهده می‌شود، مجموعه داده دو بعدی با سه خوشه $S_1$ با ۱۰ داده، $S_2$ با ۳۵ داده و $S_3$ با ۵۰۰ داده وجود دارد. میانگین LOF برای اشیا داخل سه خوشه برای اندازه‌های مختلف MinPts از ۱۰ تا ۵۰ محاسبه و نمایش داده شده است. بسیار روشن است که داده های خوشه $S_3$ در تمام اندازه‌های MinPts دارای LOF یکنواخت و برابر یک هستند و نمی‌توانند داده‌های پرت باشند. اما در خوشه $S_1$ داده‌ها برای اندازه MinPts از ۱۰ تا ۳۵ دارای LOF بسیار زیاد بوده و داده-های پرت محسوب می‌شوند. داده‌های خوشه $S_2$ نیز با اندازه MinPts = 46 شروع به پرت بودن می‌کنند، زیرا در این حالت داده‌های موجود در خوشه $S_1$ و $S_3$ به عنوان همسایه‌های داده-های خوشه $S_2$ در نظر گرفته می‌شوند و بر اساس فرمول LOF، این مقدار زیاد خواهد شد. زمانی که اندازه MinPts برابر ۴۶ می-شود، همسایه داده‌های هر شی در خوشه‌های $S_1$ و $S_2$، شامل داده‌هایی از هر سه خوشه می‌شوند و مقدار LOF زیاد خواهد شد و به سمت پرتی داده‌ها میل می‌کند.




شکل ۴: تاثیر پارامتر MinPts بر روی مقدار LOF داده‌های مجموعه داده [۳]

بنابراین، با توجه به کاربرد مورد نظر، می‌توان حد پایین (MinPtsLB) و حد بالا (MinPtsUB) برای اندازه MinPts در نظر گرفت و برای این حدود مقدار LOF را محاسبه کرد. در مثال شکل ۴، حد پایین (MinPtsLB) را ۱۰ در نظر گرفته‌ایم زیرا برای مقادیر کمتر از ۱۰ نتایج مشابهی به‌دست خواهد آمد و همچنین حد بالا (MinPtsUB) را ۵۰ انتخاب کرده‌ایم زیرا برای مقادیر بیشتر از ۵۰ نتایج مشابه می‌شود و اندازه LOF به حالت پایدار می‌رسد. سه روش اکتشافی برای محاسبه LOF پیشنهاد می‌شود. برای کاربردهای حساس مانند پزشکی و یا بانکی، به طور بد بینانه، بیشترین مقدار LOF را برای هر داده بر اساس رابطه ۲ در نظر گرفته تا بتوانیم داده‌های مشکوک به پرتی را شناسایی کنیم. برای کاربردهایی با حساسیت کمتر از میانگین LOF با MinPts های مختلف، بر اساس رابطه ۳ استفاده شود و همچنین برای کاربردهایی که نیاز به سخت گیری کمتری دارند، از کمترین مقدار LOF محاسبه شده با اندازه‌های MinPts در محدوده تعیین شده، بر اساس رابطه ۴ استفاده شود و پرتی داده‌ها تعیین شوند.

$$\text{Max}\{\text{LOF}_{MinPts}(p) \mid MinPtsLB \le MinPts \le MinPtsUB\} \quad (۲)$$

$$\text{Mean}\{\text{LOF}_{MinPts}(p) \mid MinPtsLB \le MinPts \le MinPtsUB\} \quad (۳)$$

$$\text{Min}\{\text{LOF}_{MinPts}(p) \mid MinPtsLB \le MinPts \le MinPtsUB\} \quad (۴)$$

بنابراین، در آزمایش‌های انجام شده در این مقاله، مقدار k یا همان MinPts را برابر ۲۰ در نظر گرفته‌ایم تا بهترین نتیجه را داشته باشیم.

## ۵. پیاده‌سازی و ارزیابی

برای ارزیابی و مقایسه الگوریتم‌های مختلف، نیاز به معیارهای استاندارد داریم. معیارهایی همچون دقت، نرخ تشخیص و ... برای ارزیابی الگوریتم‌های تشخیص داده پرت مورد استفاده قرار می‌گیرند. معیارها در جدول ۱ نشان داده شده‌اند.

تمام آزمایشات، توسط سیستمی با پردازنده اینتل core2Duo T9300 با فرکانس 2.5 GHz و ۴ گیگابایت حافظه رم انجام شده است که نسخه نهایی سیستم عامل ویندوز ۷ را اجرا می‌کند. همچنین الگوریتم‌ها با زبان برنامه نویسی C# در ویژوال استادیو ۲۰۱۲ پیاده سازی شده‌اند. مجموعه داده با توزیع نرمال با استفاده از نرم افزار متلب نسخه R2010a تولید شده است.

جدول۱: جدول تصادم برای معیارهای استاندارد

| | شناسایی به عنوان داده پرت (O) | شناسایی به عنوان داده نرمال (N) |
|---|---|---|
| دسته O (داده پرت) | مثبت-صحیح (TP) | منفی-کاذب (FN) |
| دسته N (داده نرمال) | مثبت-کاذب (FP) | منفی-صحیح (TN) |

با توجه به جدول ۱، می‌توان معیارهای نرخ تشخیص، نرخ هشدار خطا و دقت را به‌ترتیب از رابطه‌های ۵، ۶ و ۷ محاسبه کرد.

$$Detection\ Rate = \frac{TP}{(TP+FN)} \quad (۵)$$

$$False\ Alarm\ Rate = \frac{FP}{(FP+TN)} \quad (۶)$$

$$Accuracy = \frac{(TP+TN)}{(P+N)} \quad (۷)$$

در مقاله‌های [۱۰، ۲۰]، یکی دیگر از معیارهای اندازه‌گیری عملکرد و کارایی الگوریتم‌ها، معیار Jaccard Coefficient (JC) ارائه شده است. این معیار با توجه به معیارهای استاندارد جدول ۱ به‌دست می‌آید و در رابطه ۸ نشان داده شده است.

$$JC = \frac{TP}{FP+FN+TP} \quad (۸)$$

نرخ تشخیص، اطلاعاتی از تعداد داده‌هایی که به‌درستی الگوی نامتعارف شناسایی شده‌اند، به ما می‌دهد. در حالی که نرخ تشخیص مثبت-کاذب، در مورد تعداد داده‌هایی که به اشتباه الگوی نامتعارف شناسایی شده‌اند، گزارش می‌دهد. همچنین دقت، میزان دقت الگوریتم در شناسایی درست الگوهای نامتعارف و پرت و همچنین شناسایی درست داده‌های نرمال را تعیین می-کند. معیار JC برای یک الگوریتم در صورتی که به مقدار یک نزدیک باشد، آن الگوریتم دارای عملکرد بهتری است و دقت بالاتری در تشخیص داده های پرت دارد.

برای ارزیابی الگوریتم‌ها، نیاز به مجموعه داده‌ها داریم که در این مقاله از مجموعه داده‌های حقیقی DARPA 98 و CIMIS و همچنین مجموعه داده مصنوعی تولید شده استفاده شده است. مجموعه داده حقیقی DARPA 98، مجموعه داده‌ای پرکاربرد در زمینه ارزیابی سیستم‌های تشخیص نفوذ در شبکه‌های کامپیوتری است [۲۱]. مجموعه داده CIMIS مجموعه داده‌ای از داده‌های آب و هوای بیش از ۱۲۰ جایگاه ثبت آب و هوا در کالیفرنیا می‌باشد. ویژگی‌های موجود در این مجموعه داده شامل دمای هوا، تابش خورشید، سرعت باد و دمای خاک می‌باشد [۲۲]. در آزمایش‌های این مقاله فقط از داده‌های دمای خاک که به‌صورت روزانه از سال ۱۹۹۸ تا ۲۰۰۹ جمع آوری شده است، استفاده می‌شود [۱۰].

مجموعه داده مصنوعی با توزیع نرمال به‌صورت دو بعدی و از ۱۶۰۰ نمونه داده با هفت خوشه توزیع نرمال و میانگین‌های مختلف تشکیل شده است. مجموعه داده با توزیع نرمال، با استفاده از نرم افزار متلب تولید شده است. مجموعه داده مذکور، شامل ۵۰۰ نمونه داده با توزیع نرمال $N_1(\mu_1,\Sigma_1)$، ۵۰۰ نمونه داده با توزیع نرمال $N_2(\mu_2,\Sigma_2)$ و ۵۰۰ نمونه داده با توزیع نرمال $N_3(\mu_3,\Sigma_3)$ می‌باشد. همچنین چهار دسته داده دیگر با نرخ پایین و تعداد ۲۵ نمونه داده و توزیع‌های نرمال $N_4(\mu_4,\Sigma_4)$، $N_5(\mu_5,\Sigma_5)$، $N_6(\mu_6,\Sigma_6)$ و $N_7(\mu_7,\Sigma_7)$ در این مجموعه داده وجود دارند. پارامترهای توزیع‌های دسته داده‌ها به‌صورت زیر تعریف شده‌اند.

دسته اول : $N_1(\mu_1,\Sigma_1)$ ; $\mu_1= [+1,+1]$ ; $\Sigma_1= \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$

دسته دوم : $N_2(\mu_2,\Sigma_2)$ ; $\mu_2= [-1,-1]$ ; $\Sigma_2= \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$

دسته سوم: $N_3(\mu_3,\Sigma_3)$ ; $\mu_3= [+1,-1]$ ; $\Sigma_3= \begin{bmatrix} 0.03 & 0 \\ 0 & 0.03 \end{bmatrix}$

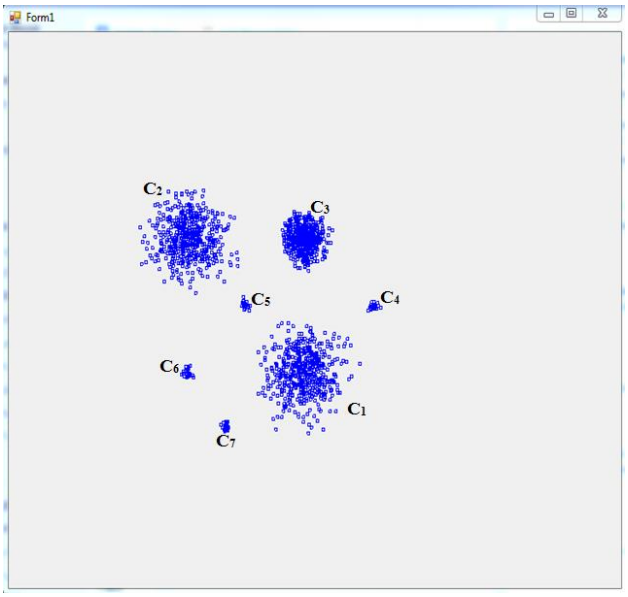
دسته چهارم: $N_4(\mu_4,\Sigma_4)$ ; $\mu_4= [0,+1.5]$; $\Sigma_4= \begin{bmatrix} 0.002 & 0 \\ 0 & 0.002 \end{bmatrix}$

دسته پنجم: $N_5(\mu_5,\Sigma_5)$ ; $\mu_5= [+1.5,0]$; $\Sigma_5= \begin{bmatrix} 0.002 & 0 \\ 0 & 0.002 \end{bmatrix}$

دسته ششم: $N_6(\mu_6,\Sigma_6)$ ; $\mu_6= [0,0]$ ; $\Sigma_6= \begin{bmatrix} 0.002 & 0 \\ 0 & 0.002 \end{bmatrix}$

دسته هفتم: $N_7(\mu_7,\Sigma_7)$; $\mu_7= [-1,+1]$ ; $\Sigma_7= \begin{bmatrix} 0.002 & 0 \\ 0 & 0.002 \end{bmatrix}$

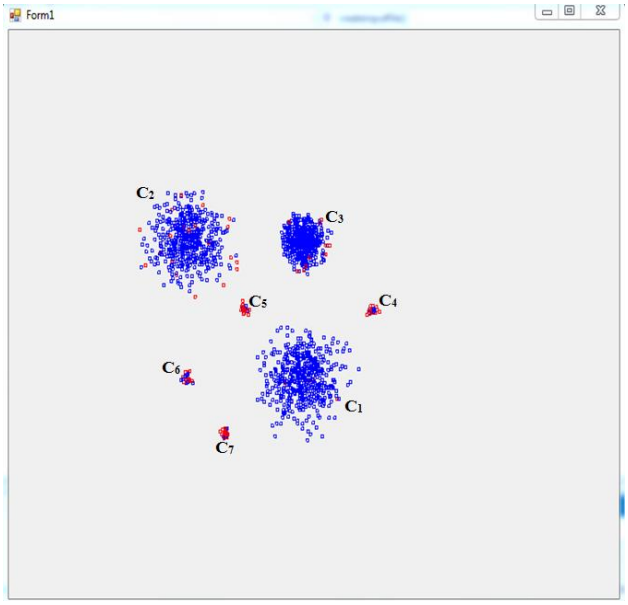
هدف در مجموعه داده مصنوعی تولید شده، شناسایی دسته داده‌ها با تراکم پایین می‌باشد. شکل ۵، نمایی از مجموعه داده دو بعدی تولید شده، را نشان می‌دهد.




شکل ۵: مجموعه داده مصنوعی ساخته شده با توزیع نرمال

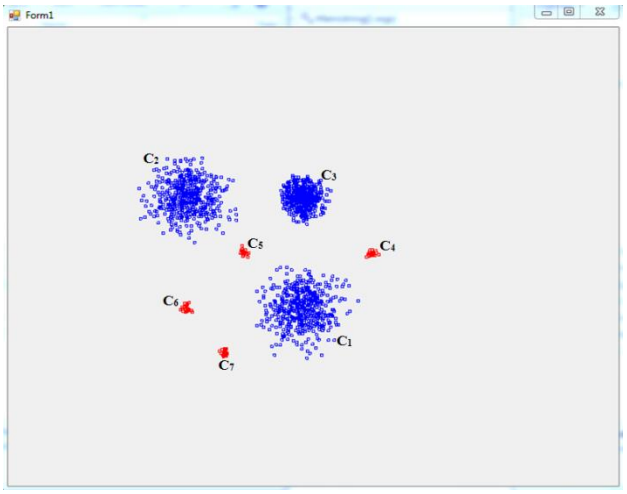
در مجموعه داده مصنوعی، ابتدا داده های دسته $N_1$ وارد سیستم شده و پس از آن داده‌های دسته $N_4$ و $N_5$ در داده‌های دسته $N_2$ پخش شده و وارد سیستم می‌شوند و در نهایت، داده‌های دسته $N_6$ و $N_7$ در داده‌های دسته $N_3$ پخش شده و وارد سیستم می-شوند. نتیجه اجرای الگوریتم ضریب ناهنجاری محلی افزایشی، برای k=20 (k تعداد همسایه‌های نزدیک به داده می‌باشد) در شکل ۶ نشان داده شده است. داده‌های نرمال با رنگ آبی و داده-های پرت با رنگ قرمز نشان داده شده است.

همان‌طور که در شکل ۶ نشان داده شده است، نرخ مثبت-کاذب بالاست، یعنی داده‌های نرمال بسیاری به‌عنوان داده پرت شناسایی شده‌اند (داده‌ها در خوشه‌های $C_1$، $C_2$ و $C_3$ با رنگ قرمز نشان داده شده‌اند که به معنی تشخیص نادرست است) و داده‌های پرتی که در دسته‌های چهارم تا هفتم قرار گرفته‌اند، به-طور کامل شناسایی نشده‌اند. علت نرخ بالای مثبت-کاذب، عدم توانایی الگوریتم در تشخیص الگوی جدید و الگوی نامتعارف می-باشد.




شکل ۶: نتایج اجرای الگوریتم ضریب ناهنجاری محلی افزایشی بر روی مجموعه داده توزیع نرمال

حال، الگوریتم ارائه شده در این این مقاله را بر روی مجموعه داده با توزیع نرمال اجرا می‌کنیم. داده‌ها به قطعه‌های ۵۰ تایی تقسیم می‌شوند. نتیجه به‌دست آمده از اجرای الگوریتم، در شکل ۷ نشان داده شده است.

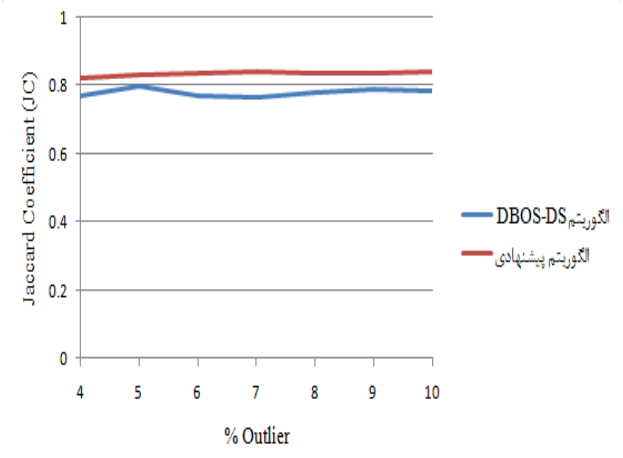



شکل ۷: نتایج اجرای الگوریتم پیشنهادی بر روی مجموعه داده توزیع نرمال

همان‌طور که در شکل ۷ نشان داده شد، الگوریتم ارائه شده در مقاله، موفق به شناسایی همه داده‌های پرت شده و نرخ مثبت-کاذب را برای این مجموعه داده به صفر رسانده است. دلیل به صفر رساندن نرخ مثبت-کاذب در الگوریتم، تاخیری است که برای شناسایی داده‌ها به‌عنوان الگویی جدید یا نامتعارف به سیستم داده می‌شود. در جدول ۲، نتایج عددی اجرای الگوریتم-های توضیح داده شده، آمده است.

در آزمایشی دیگر، الگوریتم ارائه شده در مقاله با الگوریتم DBOD-DS که در بخش ۲ معرفی شد، مقایسه شده است. دو الگوریتم بر روی مجموعه داده CIMIS اجرا شده و نتایج به‌دست آمده در شکل ۸ آورده شده است. در این آزمایش از معیار ضریب جکارد (JC) برای ارزیابی الگوریتم‌ها استفاده شده است. در مجموعه داده CIMIS درصدی داده پرت به‌صورت تصادفی اضافه شده است و نمودار به‌دست آمده در شکل ۸، اندازه ضریب جکارد در برابر درصدهای مختلفی از داده‌های پرت اضافه شده به مجموعه داده می‌باشد. همان‌طور که توضیح داده شد، مقدار ضریب جکارد در صورتی که الگوریتم عملکرد بهتری داشته باشد و نرخ تشخیص بالاتری داشته باشد، نزدیک به یک خواهد شد. در شکل ۸ عملکرد دو الگوریتم نشان داده شده و مشاهده می-شود که الگوریتم ارائه شده در همه حالات بهتر از الگوریتم DBOD-DS عمل می‌کند و ضریب جکارد آن بیشتر است.

آزمایش دیگری که در این مقاله به‌منظور ارزیابی الگوریتم ارائه شده انجام شده است، اجرا و ارزیابی الگوریتم ارائه شده، الگوریتم ضریب ناهنجاری محلی افزایشی و همچنین الگوریتم ضریب ناهنجاری محلی افزایشی بهبود یافته بر روی مجموعه داده DARPA 98 می باشد.




شکل ۸: مقایسه الگوریتم پیشنهادی با الگوریتم DBOD-DS روی مجموعه داده CIMIS

حجم داده‌ها در این مجموعه داده بسیار زیاد است، به‌همین دلیل، در این آزمایش، یک توالی ۱۶۰۰ رکوردی از رشته داده‌ها انتخاب شده است. در الگوریتم ضریب ناهنجاری محلی افزایشی، زمانی که داده‌ای وارد پنجره لغزان می‌شود، بلافاصله مقدار

ضریب ناهنجاری برای آن داده محاسبه شده و پرتی یا نرمالی داده تعیین می‌شود. بنابراین، الگوریتم ضریب ناهنجاری محلی افزایشی بسیاری از داده‌ها را به‌عنوان الگوی نامتعارف شناسایی می‌کند. الگوریتم دیگری که در بخش ۴ بررسی شد، الگوریتم بهبود یافته ضریب ناهنجاری محلی افزایشی می‌باشد. این الگوریتم نرخ تشخیص را به صد در صد رسانده است، اما هنوز نرخ مثبت-کاذب به صفر نرسیده و نیاز به بهبود دارد. در نهایت، الگوریتم ارائه شده در این مقاله، جهت بهبود دقت و شناسایی بهتر داده‌های نرمال و پرت بر روی مجموعه داده اجرا می‌شود و نتایج حاصل، در جدول ۳ آورده شده است.

همان‌طور که در شکل ۸ و جداول ۲ و ۳ مشاهده شد، الگوریتم ارائه شده دراین مقاله، به درستی داده‌های پرت را تشخیص داده و در مقایسه با الگوریتم های DBOD-DS، ضریب ناهنجاری محلی افزایشی و بهبودیافته عملکرد بهتری داشته و در ارزیابی بر روی مجموعه داده DARPA 98 دارای نرخ تشخیص۱۰۰٪ و دقت ۹۹,۲۵٪ بوده است. دقت بدست آمده توسط الگوریتم ارائه شده، برای الگوریتم‌های تشخیص داده پرت مناسب و قابل قبول است.

## ۶. نتیجه‌گیری

در مقالات مختلف، الگوریتم‌های زیادی جهت تشخیص داده‌های پرت و الگوهای نامتعارف ارائه شده است. اما برای تشخیص داده-های پرت محلی، استفاده از الگوریتم‌های مبتنی بر چگالی توصیه می‌شود. الگوریتم ضریب ناهنجاری محلی، الگوریتم مناسبی جهت تشخیص داده‌های پرت محلی برای داده‌های ایستا می-باشد. نوع دیگری از داده‌ها که امروزه در کاربردهای مختلفی دیده می‌شود، داده‌های جریانی است. راهکار مناسبی که برای غلبه بر اکثر مشکلات در هنگام برخورد با جریان داده‌ها ارائه شده است، الگوریتم ضریب ناهنجاری محلی افزایشی می‌باشد. اما با توجه به قابلیت‌های این الگوریتم، همچنان مشکلات و نرخ مثبت-کاذب بالایی در این الگوریتم وجود دارد. در این مقاله، راهکاری به منظور افزایش دقت و نرخ تشخیص و کاهش نرخ تشخیص‌های مثبت-کاذب الگوریتم‌های تشخیص داده‌های پرت ارائه کردیم. الگوریتم ارائه شده در مقاله، جریان داده‌ها را به قطعه‌های مساوی تقسیم می‌کند و ضریب ناهنجاری محلی را برای داده‌های هر قطعه محاسبه کرده و داده‌ها با ضریب ناهنجاری بالا را به لیست کاندیدای پرت اضافه می‌کند و اعلام پرتی داده را برای چند قطعه از داده‌ها به تعویق می‌اندازد. این رویکرد باعث افزایش دقت و کاهش نرخ مثبت-کاذب یا همان تشخیص اشتباه الگوهای نامتعارف می‌شود. الگوریتم ارائه شده بر روی مجموعه داده‌های حقیقی DARPA 98 و CIMIS و هم چنین مجموعه داده مصنوعی تولید شده در مقاله اجرا شده و نتایج بدست آمده، افزایش دقت و نرخ تشخیص و کاهش نرخ مثبت-کاذب را نشان داده است. دقت الگوریتم ارائه شده در مقاله بر روی مجموعه داده حقیقی DARPA 98 برابر ۹۹,۲۵٪ و همچنین بر روی مجموعه داده مصنوعی با توزیع نرمال، برابر ۱۰۰٪ می‌باشد.

جدول ۲: نتایج اجرای الگوریتم‌ها بر روی مجموعه داده مصنوعی با توزیع نرمال

| الگوریتم | TP | FN | TN | FP | نرخ تشخیص | نرخ هشدار خطا | دقت |
|---|---|---|---|---|---|---|---|
| ضریب ناهنجاری محلی افزایشی | ۶۲ | ۳۸ | ۱۴۶۳ | ۳۷ | ٪۶۲ | ٪۲,۴۶ | ٪۹۵,۳ |
| **ارائه شده در مقاله** | **۱۰۰** | **۰** | **۱۵۰۰** | **۰** | **٪۱۰۰** | **٪۰** | **٪۱۰۰** |

جدول ۳: نتایج اجرای الگوریتم‌ها بر روی مجموعه داده حقیقی DARPA98

| الگوریتم | TP | FN | TN | FP | نرخ تشخیص | نرخ هشدار خطا | دقت |
|---|---|---|---|---|---|---|---|
| ضریب ناهنجاری محلی افزایشی | ۱ | ۱ | ۱۵۴۴ | ۵۴ | ٪۵۰ | ٪۳,۳۷ | ٪۹۶,۵۶ |
| ضریب ناهنجاری محلی افزایشی بهبود یافته | ۲ | ۰ | ۱۵۷۵ | ۲۳ | ٪۱۰۰ | ٪۱,۴۴ | ٪۹۸,۵۶ |
| **ارائه شده در مقاله** | **۲** | **۰** | **۱۵۸۶** | **۱۲** | **٪۱۰۰** | **٪۰,۷۵** | **٪۹۹,۲۵** |

## مراجع


[1]. D. M. Hawkins, Identification of Outliers, London: Chapman and Hall, 1980.

[2]. E. M. Knorr and R. T. Ng, "A unified approach for mining outliers", *Centre for Advanced Studies on Collaborative research*, Toronto, 1997.

[3]. M. M. Breunig, H.-P. Kriegel, R. T. Ng and J. Sander, "LOF: Identifying Density-Based Local Outliers", *ACM Sigmod Record*, vol. 29, no. 2, pp. 93-104, 2000.

[4]. V. Chandola, A. Banerjee and V. Kumar, "Anomaly Detection : A Survey", *ACM Computing Surveys*, vol. 41, no. 3, pp. 15-26, 2009.

[5]. K. Das, Detecting patterns of anomalies, ProQuest, 2009.

[6]. M. Gupta, J. Gao, Ch. Aggarwal and Jiawei Han, "Outlier detection for temporal data", *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 5, no. 1, pp. 2250-2267, 2014.

[7]. D. R. Rani, N. Dhulipala, T. Pinniboyina and P. Chattu, "Outlier Detection for Dynamic Data Stream Using Weighted K-Means", *International Journal of Engineering Science and Technology*, vol. 3, no. 10, pp. 7484-7490, 2011.

[8]. S. Ramaswamy, R. Rastogi and K. Shim, "Efficient Algorithms for Mining Outliers from Large Data Sets", *ACM SIGMOD Record*, vol. 29, no. 2, pp. 427-438, 2000.

[9]. F. Angiulli and C. Pizzuti, "Fast Outlier Detection in High Dimensional Spaces", *Principles of Data Mining and Knowledge Discovery*, Helsinki, Finland, 2002.

[10]. M. S. Sadik and L. Gruenwald, "DBOD-DS: Distance based outlier detection for data streams", *Database and Expert Systems Applications*, Springer Berlin Heidelberg, 2010.

[11]. C. Lijun, L. Xiyin, Z. Tiejun, Z. Zhongping and L. Aiyong, "A Data Stream Outlier Delection Algorithm Based On Reverse K Nearest Neighbors", *International Symposium on Computational Intelligence and Design*, vol. 2, pp. 236-239, 2010.

[12]. S. Shiblee and L. Gruenwald, "An Adaptive Outlier Detection Technique for Data Streams", *Scientific and Statistical Database Management*, pp. 596-597, 2011.

[13]. Ch. Jua and Y. Lia, "An incremental outlier detection model for transaction data streams", *J. Inf. Comput. Sci*, vol. 10, no. 1, pp. 49-59, 2013.

[14]. Z. Miller, B. Dickinson, W. Deitrick, W. Hu and A. H. Wang, "Twitter spammer detection using data stream clustering", *Information Sciences*, vol. 260, pp. 64-73, 2014.

[15]. F. Cao, M. Ester, W. Qian and A. Zhou, "Density-Based Clustering over an Evolving Data Stream with Noise", *SDM*, vol. 6, pp. 328-339, 2006.

[16]. M. R. Ackermann, M. Märtens, Ch. Raupach, K. Swierkot, Ch. Lammersen and Ch. Sohler, "StreamKM++: A clustering algorithm for data streams", *Journal of Experimental Algorithmics (JEA)*, vol. 17, pp. 2-4, 2012.

[17]. Zh. Zheng, H. Y. Jeong, T. Huang and J. Shu, "KDE Based Outlier Detection on Distributed Data Streams in Sensor Network", *Journal of Sensors*, vol. 7, no.2, 2015.

[18]. D. Pakrajac, A. Lazarevic and L. J. Latecki, "Incremental Local Outlier Detection for Data Streams", *IEEE Symposium on Computational Intelligence and Data Mining*, 2007.

[19]. S. H. Karimian, M. Kelarestaghi and S. Hashemi, "I-IncLOF: Improved incremental local outlier detection for data streams", *16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)*, pp. 23-28, 2012.

[20]. S. Basu and M. Meckesheimer, "Automatic outlier detection for time series: an application to sensor data", *Knowledge Information System*, vol. 11, no. 2, pp. 137–154, 2007.

[21]. Lazarevic, L. Ertöz, V. Kumar, A. Ozgur and j. Srivastava, "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection", *SDM*, pp. 25-36, 2003.

[22]. California Irrigation Management Information System, web-link: ttp://wwwcimis.water.ca.gov/cimis/welcome.jsp, (accessed January, 2010).