علوم رایانش و فناوری اطلاعات
نشریه علمی انجمن کامپیوتر ایران
مجلد ۲۲، شماره ۱، ۱۴۰۳
صفحات ۹-۱۹
مقاله پژوهشی

بهبود شبکه عصبی تابع پایه شعاعی مبتنی بر خوشه‌بندی‌های K-Means++ و DBSCAN برای مجموعه داده‌های بزرگ

فرشته حاج قاضی استرآبادی[۱]، رضا قائمی[۲*]، یعقوب آراد[۳]

*نویسنده مسئول، دریافت: ۱۴۰۱/۰۵/۱۸، بازنگری: ۱۴۰۲/۰۸/۱۵، پذیرش: ۱۴۰۳/۰۲/۱۶

[۱] دانشجوی دکتری مهندسی کامپیوتر، واحد نیشابور، دانشگاه آزاد اسلامی، نیشابور، ایران
[۲] استادیار گروه مهندسی کامپیوتر، واحد قوچان، دانشگاه آزاد اسلامی، قوچان، ایران
[۳] دانشجوی دکتری مهندسی کامپیوتر، واحد نیشابور، دانشگاه آزاد اسلامی، نیشابور، ایران

چکیده

شبکه‌های عصبی تابع پایه شعاعی در مسائل تقریبی مفید هستند، اما برای آموزش در داده‌های بزرگ، نیاز به زمان طولانی دارند و درصورتی‌که داده‌ها نامعتبر یا پرت باشند، آموزش آن با خطا همراه است. خوشه‌بندی یکی از فن‌های داده‌کاوی است که در شبکه عصبی تابع پایه شعاعی به کار می‌رود. استفاده از شبکه عصبی تابع پایه شعاعی در بسیاری از حوزه‌ها مانند پردازش تصویر، طبقه‌بندی متن، بیومتریک و ریزآرایه به دلیل اندازه بزرگ مجموعه داده‌ها، نیاز به زمان و حافظه زیادی برای پردازش دارد. بااین‌حال، خوشه‌بندی می‌تواند این مشکل را بهبود بخشیده و مراکز توابع گوسی را به‌درستی در بین داده‌ها تعیین کند. در این مقاله، روش خوشه‌بندی جدیدی با استفاده از الگوریتم‌های K-Means++ و DBSCAN ارائه‌شده که می‌تواند سرعت و دقت شبکه عصبی تابع پایه شعاعی را بهبود دهد. درروش پیشنهادی، ابتدا داده‌ها توسط الگوریتم K-Means++ گروه‌بندی‌شده و سپس، الگوریتم DBSCAN به‌طور جداگانه برای هر گروه اعمال می‌شود و داده‌ها خوشه‌بندی می‌شوند. ایده این مقاله، در بهبود سرعت اجرای الگوریتم DBSCAN از طریق کاهش محاسبات با تقسیم فضای کاری به تعدادی ناحیه جداگانه و استفاده از آن در شبکه عصبی تابع پایه شعاعی است. نتایج مدل پیشنهادی در مقایسه با شبکه عصبی تابع پایه شعاعی استاندارد و مدل بهبودیافته RBF-KMC بر روی مجموعه داده‌های Image segmentation، Pendigit، Letters و Shuttle control بر اساس زمان آموزش و دقت نتایج بهتری داشته است.

کلمات کلیدی: خوشه‌بندی، داده‌های حجیم، سرعت، دقت، شبکه عصبی تابع پایه شعاعی، K-Means++، DBSCAN.

۱- مقدمه

عصر داده‌های بزرگ منجر به توسعه و کاربرد فناوری‌ها و روش‌هایی باهدف استفاده از مقادیر زیادی داده برای حمایت از تصمیم‌گیری و فعالیت‌های کشف دانش شده است [۱]. در بین فن‌های داده‌کاوی، شبکه‌های عصبی ازجمله شبکه عصبی تابع پایه شعاعی[۱] (RBF) از روش‌های پرکاربرد در مواجه با مجموعه داده‌های مختلف ازجمله داده‌های بزرگ است. حجم زیادی از داده‌ها باعث شده است تا محققان و صنایع درراه حل‌های محاسباتی برای تجزیه‌وتحلیل داده‌های بزرگ تجدیدنظر کنند. به‌عنوان نمونه، تأکید زیادی بر طراحی الگوریتم‌های جدید که در محاسبات کارآمدتر هستند [۲]. بااین‌حال، عملیات تجزیه‌وتحلیل و بازیابی به دلیل هزینه‌های محاسباتی بالای پردازش داده‌ها می‌تواند بسیار زمان‌بر شود. دسته‌ای از روش‌های رایج برای تجزیه‌وتحلیل داده‌ها به‌عنوان داده‌کاوی شناخته می‌شود که به معنای شناسایی مفید، قابل‌اعتماد، الگوهای داده ساده و قابل‌درک که داده‌های خام را به داده یا اطلاعات مفید تبدیل می‌کند [۳]. خوشه‌بندی یکی از فن‌های داده‌کاوی است که در شبکه عصبی تابع پایه شعاعی نیز به کار گرفته می‌شود. خوشه‌بندی داده‌ها حوزه مهمی از یادگیری بدون نظارت در نظر گرفته می‌شود که در آن داده‌ها را می‌توان بر اساس شباهت‌هایشان از دیدگاهی آگاهانه در کل مجموعه داده به گروه‌های مختلف تقسیم نمود [۴]. این خوشه‌بندی به شبکه عصبی تابع پایه شعاعی این امکان را می‌دهد تا نورون‌های خود را در مرکز داده‌ها قرار داده و تخمین بهتری از الگو

داده‌ها انجام دهد. خوشه‌بندی در طیف وسیعی از زمینه‌ها مانند شناسایی مجدد وسیله نقلیه [۵]، حذف نویز تصویر [۶]، پردازش سری زمانی [۷] و تجزیه‌وتحلیل شبکه‌های اجتماعی مبتنی بر وب [۸] استفاده می‌شود. امروزه، روش‌های فوقی مرسوم در برخورد با داده‌های بزرگ به‌اندازه کافی کارآمد نیستند و به دلیل پیچیدگی محاسباتی بالا قابل‌اجرا نمی‌باشند [۹].

بنابراین یکی از بهترین ویژگی‌های شبکه‌های عصبی، توانایی آن در تعمیم و تقریب داده‌های نمونه بدون نیاز به تعیین معادله و ضرایب است، به‌ویژه زمانی که یک مدل ناشناخته، یک رابطه پیچیده ناشناخته را توصیف می‌کند و داده‌های آموزشی فراوانی دارد. به دلیل توانایی آن‌ها در تعمیم، شبکه‌های عصبی تابع پایه شعاعی معمولاً برای این منظور انتخاب می‌شوند [۱۰-۱۱]؛ اما در کلان داده‌ها، شبکه عصبی تابع پایه شعاعی در بسیاری از حوزه‌ها مانند پردازش تصویر، طبقه‌بندی متن، بیومتریک، ریزآرایه و غیره دارای اندازه مجموعه داده‌هایی به‌قدری بزرگ هستند که سیستم بلادرنگ نیاز به زمان و حافظه طولانی برای پردازش آن‌ها دارد. شبکه عصبی باید در کلیه نمونه‌های آموزشی موجود جستجو کند که به حافظه زیادی نیاز دارد و محاسبه فاصله تا مرکز را انجام دهد که در طول آموزش شبکه عصبی برای اهداف تقریبی، فرآیند کندی است. علاوه بر این، به دلیل ذخیره تمامی فواصل نمونه برای مجموعه داده‌های آموزشی در شبکه عصبی تابع پایه شعاعی، ممکن است فواصل نویز نیز ذخیره شوند که این می‌تواند باعث کاهش دقت تقریبی گردد.

الگوریتم DBSCAN [12]، یکی از محبوب‌ترین و کاربردی‌ترین الگوریتم‌های خوشه‌بندی است. بااین‌حال، این الگوریتم در مواجهه باکلان داده‌ها به‌درستی عمل نمی‌کند. یکی از مهم‌ترین معایب آن، سرعت اجرای پایین است. با توجه به‌سرعت پیشرفت در دنیای امروز، زمان یکی از مهم‌ترین پارامترها است و باید موردتوجه ویژه قرار گیرد. برای غلبه بر مشکل پردازش کلان داده، روش‌های مختلفی ازجمله بهینه‌سازی الگوریتم‌های مرسوم ازنظر محاسبات و حتی مصرف حافظه، کاهش داده‌ها از طریق کاهش ابعاد یا نمونه‌های داده، پردازش جریان داده‌ها و استفاده از ابزارهای سنگین پیشنهادشده است.

ایده این مقاله، در بهبود سرعت اجرای الگوریتم DBSCAN از طریق ایده کاهش محاسبات با تقسیم فضای کاری به تعدادی ناحیه جداگانه و استفاده از این خوشه‌بندی در شبکه عصبی تابع پایه شعاعی است. برای تقسیم داده‌ها به گروه‌های کوچک‌تر، از الگوریتم خوشه‌بندی معروف دیگری بنام K-Means++ [13] استفاده می‌گردد. این الگوریتم سرعت همگرایی مناسبی را نشان می‌دهد و می‌تواند تقسیم داده‌های خوبی را فراهم کند. لازم به ذکر است که اجرای K به معنی چند تکرار، این امکان را می‌دهد که بدون بار محاسباتی زیاد به نتایج دلخواه دست‌یابیم، چراکه K-Means++ نه برای به دست آوردن خوشه‌بندی نهایی داده‌ها، بلکه صرفاً برای تقسیم آن‌ها استفاده می‌شود.

به‌طورکلی، الگوریتم پیشنهادی به این صورت عمل می‌کند که ابتدا، داده‌ها توسط الگوریتم K-Means++ با تعداد کمی از تکرارها گروه‌بندی می‌شوند. سپس، الگوریتم DBSCAN به‌طور جداگانه برای هر قسمت اعمال می‌شود و داده‌ها خوشه‌بندی می‌شوند. درنهایت، بخش‌های مختلف یکپارچه‌شده‌اند. برای به دست آوردن نقاط اصلی در الگوریتم DBSCAN، اگر قطعات MinPts در شعاع ε وجود داشته باشد، باید برای هر قطعه داده تحلیل و آنالیز شود. ازاین‌رو، فاصله بین هر قطعه از داده‌ها و سایر داده‌های موجود در مجموعه داده باید محاسبه گردد؛ بنابراین، مقدار زیادی از محاسبات برای یک مجموعه داده بزرگ انجام می‌شود. بااین‌حال درروش پیشنهادی، فضای داده‌ای که باید در نظر گرفته شود، کوچک‌تر است، زیرا DBSCAN به‌طور جداگانه در هر گروه اعمال می‌شود و درنتیجه، تعداد نقاط کمتری وجود دارد که فواصل آن‌ها باید محاسبه شود. این امر، زمان اجرا را به میزان قابل‌توجهی کاهش می‌دهد. استفاده از این خوشه‌بندی در شبکه عصبی تابع پایه شعاعی باعث افزایش دقت در انتخاب مراکز و همچنین، سرعت آموزش تا حد زیادی شده است.

در ادامه مقاله، بخش دوم ادبیات تحقیق را مرور و بررسی می‌کند. در بخش سوم، الگوریتم پیشنهادی تشریح می‌شود. بخش چهارم، نتایج آزمایش‌های الگوریتم پیشنهادی را تحلیل و ارزیابی می‌کند. درنهایت، بخش پنجم نتیجه‌گیری و پیشنهادهایی را برای تحقیقات آینده ارائه می‌دهد.

۲- کارهای مرتبط

امروزه سازمان‌ها به دنبال ثبت و تجزیه‌وتحلیل داده‌ها از طریق روش‌های کارآمد هستند تا بتوانند تصمیم‌گیری مبتنی برداده را تحقق بخشند. کلان داده‌ها را می‌توان به داده‌های ساختاری و منظم برای پردازش آسان‌تر تبدیل نمود. در این مقاله، به داده‌های ساختاریافته و بزرگ پرداخته می‌شود. اصطلاح داده‌های بزرگ به داده‌هایی اطلاق می‌شود که برای استخراج دانش نیاز به پردازش و تجزیه‌وتحلیل روش‌ها و معماری‌های جدید دارند [۱۴]. با توجه به‌اندازه بزرگ این داده‌ها، تجزیه‌وتحلیل کارآمد آن‌ها از طریق فن‌های مرسوم موجود بسیار دشوار است. با توجه به مطالعات بررسی‌شده در تحقیق [۱۵]، منظور از داده‌های بزرگ مجموعه داده‌هایی است که نمی‌توانند توسط ابزارهای نرم‌افزاری معمولی روی یک ماشین در زمان قابل قبولی مدیریت و پردازش شوند.

در سال‌های اخیر، روش خوشه‌بندی مختلفی برای کلان داده‌ها معرفی‌شده است. روش‌های فوق برای اجرای الگوریتم‌های خوشه‌بندی و کار با داده‌های بزرگ از طریق بهبود مقیاس‌پذیری و سرعت آن‌ها استفاده می‌شوند. اخیراً میر جلیلی [۱۶] نشان داد که ترکیبی از الگوریتم‌های تکاملی مانند بهینه‌سازی ازدحام ذرات با شبکه عصبی تابع پایه شعاعی عملکرد خوبی در مسائل طبقه‌بندی و مسائل تقریب نشان می‌دهد. استفاده از الگوریتم تکاملی به‌عنوان ابزاری برای انتخاب مراکز دقیق‌تر نیز در بسیاری از متون اخیر گزارش‌شده است [۱۷-۱۸]. لذا الگوریتم تکاملی برای آموزش شبکه عصبی تابع پایه شعاعی روش مناسبی است، اگر سرعت آموزش شبکه و هزینه محاسبات نگرانی اصلی نباشد. در کنار افزایش محبوبیت شبکه‌های عصبی تابع پایه شعاعی، کاربرد این شبکه عصبی در زمینه‌هایی مانند طبقه‌بندی [۱۹]، تشخیص الگو [۲۰] و پیش‌بینی [۲۱] نیز افزایش داشته و استفاده و قابلیت اطمینان شبکه عصبی تابع پایه شعاعی در بسیاری از زمینه‌ها را نشان می‌دهد.

بااین‌حال، دقت شبکه عصبی تابع پایه شعاعی به‌طور عمده به مراکز اولیه انتخاب‌شده از مجموعه داده، قبل از شروع آموزش شبکه بستگی دارد [۲۲-۲۳]. علاوه بر این، اندازه مجموعه داده‌های آموزشی و داده‌های نامعتبر موجود در مجموعه داده‌ها نیز نقش مهمی در تعیین سرعت و دقت آموزش شبکه ایفا می‌کنند [۲۴-۲۵]. همچنین، گزارش‌شده است که الگوریتم یادگیری برای آموزش شبکه‌ها ممکن است با افزایش مجموعه داده‌ها بدتر عمل کند [۲۶]. ازاین‌رو برای حل این مشکلات، محققان استفاده از الگوریتم‌های خوشه‌بندی را در انتخاب مراکز [۲۷-۳۰] برای شبکه عصبی تابع پایه شعاعی برای دستیابی به‌دقت و سرعت آموزش بهتر و جلوگیری از گنجاندن مجموعه داده‌های نامعتبر احتمالی در آموزش شبکه‌ها پیشنهاد کردند.

پرکاربردترین الگوریتم خوشه‌بندی در انتخاب مراکز، الگوریتم K-Means است، زیرا سریع‌ترین و کم پیچیده‌ترین الگوریتم است و در مقایسه با سایر الگوریتم‌های خوشه‌بندی موجود، دقت خوبی برای انتخاب مراکز دارد. بااین‌حال، الگوریتم‌های خوشه‌بندی مرسوم موجود، الگوریتم‌های بی‌عیب و نقصی نیستند که دقت بالایی را در انتخاب مراکز تضمین نمایند. به‌این‌ترتیب، محققان راه‌هایی برای بهبود ضعف الگوریتم‌های خوشه‌بندی موجود را بررسی می‌کنند.

در ادبیات اخیر، در تحقیق [۳۱]، یک الگوریتم K-Means بهبودیافته با رویکردهای چند خوشه‌ای تکراری برای انتخاب مراکز و درعین‌حال به حداقل

رساندن تابع هدف پیشنهادشده است. این روش خوشه‌بندی سطح اول را انجام داده و مراکز نسل اول را ذخیره می‌کند و سپس، خوشه‌بندی سطح دوم را برای کوتاه کردن فواصل مراکز اجرا می‌کند. این روش دقت خوبی دارد، اما هزینه محاسباتی سنگینی است. در تحقیق [۳۲]، از یک K-Means سه سطحی برای تعیین مراکز استفاده‌شده است. این الگوریتم می‌تواند بر نویز غلبه کند و داده‌های نامعتبر را فیلتر کند، اما در بار محاسباتی پرهزینه است. کار در بهبود الگوریتم K-Means با استفاده از مفاهیم چگالی ادامه یافته تا جایی‌که در تحقیق [۲۷]، استفاده از چگالی را برای بهبود الگوریتم K-Means و به دست آوردن دقت مناسب معرفی می‌گردد. این روش همچنین نشان می‌دهد که الگوریتم K-Means نسبت به داده‌های نویز حساسیت کمتری دارد. این روش بر محاسبه تراکم هر شیء یا نقاط داده در مجموعه داده متمرکز است و تنها مراکزی را انتخاب می‌کند که نقاط تراکم بالاتری در نزدیکی آن دارند. تحقیقات مشابهی از سال‌های اولیه در ادبیات [۳۳]، با استفاده از مفاهیم چگالی که بر محاسبه چگالی نقاط داده در اطراف مراکز تمرکز دارد، یافت شده است.

در تحقیق [۳۴]، یک روش انتخاب مراکز اولیه برای الگوریتم K-Means با استفاده از شاخص‌های اتکینسون[۲] معرفی‌شده است. به‌جای انتخاب تصادفی مراکز اولیه برای ادامه الگوریتم K-Means برای تنظیم بهتر مراکز، نویسندگان برای استفاده از این رویکردها، خطای احتمالی داده‌های نامعتبر را از رویکردهای انتخاب تصادفی کاهش می‌دهند. شاخص‌های اتکینسون از مرز و نابرابری برای تعیین محدوده انتخاب مراکز برای الگوریتم K-Means استفاده می‌کنند. در تحقیق [۳۵]، از مفاهیم مشابه در الگوریتم K-Means پیشنهادی استفاده می‌کنند. این الگوریتم از کران‌های بالا و پایین بین یک نقطه و مراکز اولیه تصادفی برای انتخاب مراکز مناسب استفاده می‌کند. این الگوریتم یک فرآیند پیچیده را برای انجام محاسبات اعمال کرده که برای محققان به‌راحتی قابل‌درک نیست. علاوه بر این، محققانی نیز از توابع هسته برای محاسبه فاصله در الگوریتم K-Means به‌منظور به دست آوردن نتایج طبقه‌بندی بهتر استفاده کردند [۳۶]. با استفاده از تابع هسته، مجموعه داده‌های پیچیده به ابعاد بالاتر نگاشت می‌شوند که مجموعه داده را به‌راحتی قابل‌تفکیک و طبقه‌بندی مجموعه داده‌ها تبدیل می‌کند.

در تحقیق [۳۷]، روشی برای تعیین وزن به خوشه‌ها نسبت به واریانس معرفی‌شده است. این کار بینش جدیدی در مورداستفاده از رویکردهای آماری در الگوریتم یادگیری ماشین ایجاد می‌کند، بااین‌حال در محاسبات بسیار پیچیده و پرهزینه است. در تحقیق [۳۸]، استفاده از فاصله ماهالانوبیس[۳] برای جایگزینی فاصله اقلیدسی در الگوریتم K-Means پیشنهادشده است که البته بهبود قابل‌توجهی در دقت ایجاد نکرده است.

به‌طورکلی، خوشه‌بندی را می‌توان به دو گروه اصلی [۳۹] طبقه‌بندی نمود، شامل روش‌های مبتنی بر یک ماشین واحد و روش‌های مبتنی بر ماشین‌های متعدد. الگوریتم‌های خوشه‌بندی بر اساس یک ماشین واحد روی یک ماشین اجرا می‌شوند و فقط می‌توانند از منابع آن ماشین استفاده کنند. الگوریتم‌های خوشه‌بندی مبتنی بر ماشین‌های متعدد بر روی دو یا چند ماشین اجرا می‌شوند و می‌توانند از منابع چندین ماشین به‌طور هم‌زمان استفاده کنند.

در روش‌های خوشه‌بندی بر اساس یک ماشین واحد، روش‌های پیشنهادی در تحقیق‌های قبلی برای حل مشکل پردازش داده‌های بزرگ اشاره می‌شود. روش‌های فوق را می‌توان به سه گروه تقسیم نمود. اولین فن، کاهش اندازه داده است که می‌تواند از طریق کاهش نمونه‌ها، ابعاد یا هر دو انجام شود. فن دوم، پردازش جریان داده است که پردازش حجم زیادی از داده‌ها را در یک ماشین ممکن می‌کند، اما از مشکلات خاصی رنج می‌برد. فن سوم، الگوریتم‌های بهینه‌سازی پردازش داده است که در آن، داده‌ها را می‌توان با بار محاسباتی کمتری پردازش کرد. ایده این مقاله، در گروه سوم روش‌ها بر اساس ماشین نهفته است.

در تحقیق [۴۰]، یک الگوریتم خوشه‌بندی مبتنی بر چگالی سریع بر اساس الگوریتم DBSCAN و با استفاده از ایده کاهش محاسبات پیشنهادشده است. الگوریتم ابتدا کلیه داده‌ها را بر اساس مختصات بعد مرتب می‌کند و سپس، با کاهش زمان اجرای هر پرس‌وجو منطقه یا فرکانس پرس‌وجوهای ناحیه، سرعت کل اجرای DBSCAN را افزایش می‌دهد. در تحقیق دیگری، یک الگوریتم خطی DBSCAN بر اساس هشینگ حساس به محل ارائه‌شده است تا سرعت یافتن نزدیک‌ترین گره‌های همسایه را افزایش دهد [۴۱]. مزیت استفاده از هشینگ حساس به محل این است که پیچیدگی زمانی و مقیاس داده را کاهش می‌دهد. الگوریتم دو بخش را در نظر گرفته است. شاخص هشینگ حساس به محل در بخش اول ساخته‌شده، درحالی‌که خوشه‌بندی در بخش دوم، توسط الگوریتم DBSCAN بر اساس شاخص بازیابی هشینگ حساس به محل[۴] انجام می‌شود که یک ضعف الگوریتم مشکل در تعیین مقادیر برای پارامترهای ورودی است.

یک الگوریتم خوشه‌بندی جدید در تحقیق [۴۲] ارائه‌شده است. این الگوریتم برای بهبود دقت خوشه‌بندی با ادغام K-Means++، با یک نقشه خودسازمان‌دهی[۵] پیشنهادشده است. درواقع، K-Means++ وزن اولیه و مرکز اولیه را تعیین کرده و سپس، از نقشه خودسازمان‌دهی برای یافتن راه‌حل مناسب برای خوشه‌بندی نهایی داده‌ها استفاده‌شده است. الگوریتم نقشه خودسازمان‌دهی باهدف پیشنهاد خوشه‌بندی بهینه و با سرعت‌بالا، ازنظر صرفه‌جویی در زمان آموزش و میزان خطا به‌درستی عمل نموده است. بااین‌وجود، مشکل اصلی K-Means++ این است که ذاتاً ترتیبی است. درنتیجه، کارایی آن برای کلان داده محدود است.

درراهکار پیشنهادی در این مقاله، نیازی به تکرار K-Means++ تا انتها نیست، زیرا تکرارها محدود هستند. در مقاله [۴۳]، روشی از نوع الگوریتم‌های مبتنی بر چگالی افزایشی برای مجموعه داده‌های بزرگ پیشنهادشده است که هدف این روش، بهبود الگوریتم افزایشی DBSCAN با محدود کردن فضای جستجو به چند قسمت به‌جای استفاده از کل مجموعه داده بود. این امر فرآیند خوشه‌بندی را تسریع می‌کند.

الگوریتم خوشه‌بندی سلسله مراتبی K-Means برای یافتن مراکز اولیه باکیفیت بالا برای مجموعه داده‌های بزرگ در تحقیق [۴۴] پیشنهاد شد. در این الگوریتم، داده‌ها در یک ساختار سلسله مراتبی کاهش می‌یابند؛ بنابراین، در گروه فن‌های کاهش داده‌ها قرار می‌گیرند. مرکزها از طریق ساختار K-Means سلسله مراتبی متشکل از چندین لایه به دست می‌آیند که اولین لایه شامل مجموعه داده اصلی است. لایه بعدی شامل مجموعه داده‌های کوچک‌تری است که از طریق مدل-هایی مشابه مجموعه داده اصلی ایجادشده‌اند. به‌عبارت‌دیگر، مراکز هر خوشه نشان‌دهنده داده‌های آن خوشه است و به سطح بعدی منتقل می‌شود. ازنظر محاسباتی فشرده، الگوریتم‌های سلسله مراتبی به‌طورمعمول به‌عنوان خط مبنا یا برای بهینه‌سازی الگوریتم اصلی خوشه‌بندی استفاده می‌شوند و نمی‌توانند نتایج بسیار خوبی برای محاسبات در مقیاس بزرگ به دست آورند. روش دیگری در بهبود K-Means++ با رویکردی تکراری برای افزایش کیفیت و سرعت خوشه‌بندی حاصل از K-Means و K-Means++ از طریق حذف یک خوشه و تقسیم خوشه دیگر در تحقیق [۴۵] ارائه‌شده است. خوشه‌ها بر اساس مقایسه توابع سود و هزینه حذف و اضافه می‌شوند و مشکلی که در این الگوریتم وجود دارد، این است که محاسبات بیش‌ازحد دارد که باعث کاهش سرعت اجرا می‌شود.

در تحقیق دیگری، الگوریتمی برای افزایش سرعت DBSCAN در هنگام اجرای داده‌های بزرگ طراحی‌شده است [۴۶]. الگوریتم، ابتدا فضای داده را به یک ساختار شبکه تقسیم می‌کند و سپس، یک مقدار چگالی را به هر سلول شبکه اختصاص می‌دهد. سپس فضاهای شبکه را بررسی می‌کند تا فضاهای همسایه را پیداکرده و متراکم‌ترین همسایه را در هر فضا مشخص کند. در مرحله بعدی، یعنی در مرحله تشکیل خوشه، الگوریتم زنجیره‌ای از متراکم‌ترین همسایگان را تشکیل می‌دهد و این کار را ادامه می‌دهد تا خوشه‌ها توسعه یابند. برای این منظور، بخش‌های شبکه

را در نظر گرفته و متراکم‌ترین همسایه‌هایشان را به خوشه اضافه می‌کند. بخش بزرگی از زمان اجرای الگوریتم، صرف یافتن متراکم‌ترین همسایگان می‌شود که به‌عنوان یک ضعف الگوریتم در نظر گرفته می‌شود. در ادامه و در تحقیق [۴۷]، الگوریتم HCA-DBSCAN ارائه شد که در آن، ابرمکعب‌هایی را مانند شبکه‌هایی با قطر ε در فضای داده در نظر می‌گیرد و داده‌های هر ابرمکعب در یک خوشه قرار دارند. در هر ابرمکعب، چند نقطه که نزدیک به مرزها هستند، به‌عنوان عامل انتخاب می‌شوند. فقط عوامل هر جفت شبکه به‌جای کلیه نقاط موردبررسی و مقایسه قرار می‌گیرند. اگر فاصله بین دو عامل از دو شبکه کوچک‌تر از ε باشد، دو شبکه ادغام می‌شوند و داده‌های آن‌ها در یک خوشه قرار می‌گیرند. با استفاده از این عوامل، تعداد مقایسه‌هایی که برای محاسبه فواصل بین داده‌ها باید انجام شود، کاهش می‌یابد و درنهایت، الگوریتم با سرعت‌بالاتری اجرا می‌شود. به دلیل چندین تلاش برای جستجوی شاخص‌ها برای به دست آوردن شاخص‌های مجاور، این الگوریتم نیاز به‌صرف زمان زیادی دارد، موردی که به‌عنوان یک ضعف محسوب می‌شود.

ایده‌ای که در سال ۲۰۱۹ برای پیاده‌سازی DBSCAN بر روی‌داده‌های بزرگ ارائه شد، یک روش نمونه‌گیری برای کاهش اندازه داده‌ها است [۴۸]. این مقاله دو روش را برای نمونه‌گیری بهتر در DBSCAN پیشنهاد می‌کند. درنتیجه، زمان اجرای DBSCAN به دلیل کاهش تعداد نمونه‌ها کاهش می‌یابد. یکی از روش‌ها، نسخه اصلاح‌شده Rough DBSCAN است، درحالی‌که روش دیگر یک فن اکتشافی است. در تحقیق دیگری، یک روش تقریبی بنام KNN-BLOCK DBSCAN برای خوشه‌بندی کلان داده در سال ۲۰۱۹ پیشنهاد شد [۴۹]. برای این منظور، آن‌ها از یک الگوریتم نزدیم‌ترین همسایه[۶] برای شناسایی بلوک‌های اصلی (CB)، بلوک‌های غیرهسته‌ای (NCB) و بلوک‌های نویز (NOB) که به‌ترتیب شامل نقاط اصلی، نقاط مرزی و نقاط نویز هستند، استفاده شد. سپس CBهای قابل‌دسترسی با چگالی ادغام شدند، درحالی‌که NCBها به یک خوشه مناسب اختصاص داده شدند. این محققین در مقاله [۵۰]، یک روش خوشه‌بندی مبتنی بر شبکه بنام K-Means++ DBSCAN BLOCK را در دو نسخه برای داده‌های بزرگ با ابعاد بالا در سال ۲۰۲۰ پیشنهاد کردند. با استفاده از روش کاهش محاسباتی، این ایده دو فن را پیشنهاد می‌کند. فن اول، برای شناسایی بلوک‌های هسته داخلی که کلیه نقاط آن نقاط اصلی هستند، استفاده می‌شود. فن دوم، یک الگوریتم تقریبی با سرعت‌بالا است که پیش‌بینی می‌کند که آیا دو بلوک هسته داخلی قابل‌دسترسی به چگالی هستند یا خیر. علاوه بر این، یک درخت پوشش برای تسریع محاسبات چگالی برای نقاط بازدیدنشده استفاده می‌شود. مزیت این روش این است که می‌تواند داده‌هایی با ابعاد بالا را نسبتاً سریع خوشه‌بندی کند.

روش‌های خوشه‌بندی بر اساس چند ماشین نیز دارای تحقیقات متفاوتی است. در تحقیق [۵۱]، برای بهبود خوشه‌بندی کلان داده از طریق الگوریتم K-Means تلاش‌هایی انجام‌شده و برای این منظور از چارچوب MapReduce استفاده‌شده است. این مقاله بر روی بهینه‌سازی انتخاب مراکز خوشه‌بندی اولیه برای افزایش دقت و سرعت متمرکزشده است. به این منظور، تعدادی زیرمجموعه از کلان داده‌ها از طریق نمونه‌گیری انتخاب شدند. زیرمجموعه‌ها به‌صورت موازی به‌منظور به دست آوردن مراکزی که می‌توانند برای خوشه‌بندی مجموعه داده اصلی استفاده شوند، پردازش شدند. ادغام پس‌ازانجام نمونه‌برداری‌های موردنیاز صورت گرفت و در این روش، از K-Means++ برای انتخاب مراکز اولیه استفاده شد.

در پژوهش [۵۲]، مجموعه داده به بخش‌های کوچک‌تر تقسیم‌شده و به چندین گره در یک خوشه از ماشین‌ها توزیع‌شده است. برای این منظور Apache Hadoop به‌عنوان یک چارچوب مقیاس‌پذیر و قدرتمند استفاده شد. K-Means در دو فاز بر روی ماشین‌ها پیاده‌سازی می‌شود که در مرحله اول، تعداد زیادی خوشه‌های اولیه ایجاد می‌شوند. برای این منظور، از خوشه‌بندی احتمالی برای انتخاب مرکزهای اولیه بهتر و کاهش تعداد دوره‌های همگرایی استفاده می‌شود. در فاز دوم، یک آستانه برای ادغام خوشه‌های به‌دست‌آمده از فاز اول استفاده می‌شود. زمان اجرای این روش طولانی‌تر از روش‌های جدید ارائه‌شده در ادبیات تحقیق مطرح‌شده است. درواقع، تقریباً برابر با K-Means است که یک نقطه‌ضعف محسوب می‌شود.

در تحقیق [۵۳]، طرحی برای اجرای موازی DBSCAN در سال ۲۰۱۸ ارائه شد که سلول‌های کوچک داده را به‌طور تصادفی به موازی تقسیم می‌کند. در این تحقیق، یک الگوریتم موازی جدید DBSCAN معروف به RP-DBSCAN ارائه شد که در آن، از طرح تقسیم تصادفی داده‌ها و یک فرهنگ لغت سلولی دوسطحی استفاده شد که خلاصه‌ای فشرده از کل مجموعه داده بود. الگوریتم به‌طور هم‌زمان برای هر بخش از داده‌ها خوشه‌های محلی را پیداکرده و سپس، آن‌ها را به‌منظور به دست آوردن خوشه‌بندی نهایی ادغام می‌کند. در این روش، هزینه‌های تقسیم فضایی به دلیل استفاده از توزیع داده‌های تصادفی کاهش می‌یابد. بااین‌حال، زمان اجرا را طولانی می‌کند.

در آخرین روش‌های مطرح‌شده در سال ۲۰۲۰، بر اساس شباهت همسایگی، یک الگوریتم DBSCAN اصلاح‌شده ارائه‌شده است که با استفاده از روش درخت پوشش، همسایه‌های موازی هر نقطه را بازیابی می‌کند و بسیاری از محاسبات غیرضروری را برای اندازه‌گیری فاصله از طریق اصل نابرابری مثلثی بررسی می‌کند. مزیت قابل‌توجه این الگوریتم سرعت‌بالای آن است. درواقع، مناطق با تراکم بالا را به‌عنوان یک دسته طبقه‌بندی می‌کند و خوشه‌بندی فضایی موردنظر را به‌سرعت پیدا می‌کند. ازآنجایی‌که این روش موازی است، اگر از فن‌هایی مانند کاهش نقشه استفاده شود، الگوریتم پیشنهادی بسیار سریع‌تر اجرا می‌شود [۵۴]. جدول ۱ خلاصه‌ای از روش‌های متفاوت مرورشده که مبتنی بر خوشه‌بندی DBSCAN است را نشان می‌دهد.

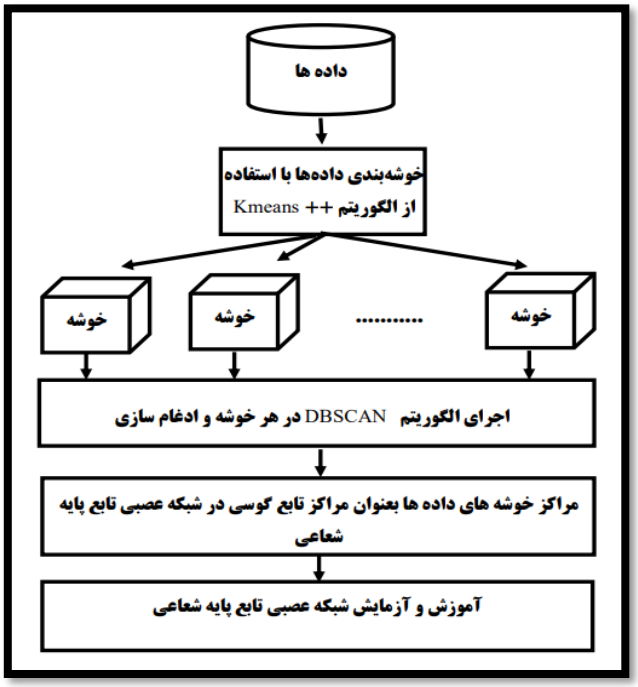
جدول ۱- روش‌های خوشه‌بندی مبتنی بر DBSCAN

مرجع	الگوریتم پیشنهادی	نقاط قوت	نقاط ضعف
[۴۰]	چگالی سریع در DBSCAN	دقت مناسب	سرعت‌پایین و زمان اجرای بالا
[۴۱]	هشینگ حساس به محل در DBSCAN	دقت و سرعت مناسب	دقت نامناسب
[۴۲]	K-Means++ با نقشه خودسازمان‌ده	سرعت مناسب	زمان اجرای بالا
[۴۳]	محدود کردن فضای جستجو در DBSCAN	زمان اجرای مناسب	دقت پایین
[۴۴]	K-Means سلسله مراتبی	سرعت مناسب	دقت پایین
[۴۵]	حذف و تقسیم خوشه در K-Means++	دقت و سرعت مناسب	دقت نامناسب
[۴۶]	تقسیم شبکه‌ای در DBSCAN	زمان اجرای مناسب	دقت پایین
[۴۷]	تقسیم ابرمکعب در DBSCAN	زمان اجرای مناسب	دقت نامناسب
[۴۸]	کاهش نمونه‌ها در DBSCAN	زمان اجرای مناسب	دقت پایین
[۴۹]	KNN-BLOCK-DBSCAN	زمان اجرای مناسب	دقت پایین
[۵۰]	Block K-Means++ DBSCAN	دقت و سرعت مناسب	دقت نامناسب
[۵۱]	K-Means++ MapReduce	سرعت مناسب	دقت پایین
[53]	اجرای موازی RP-DBSCAN	سرعت مناسب	دقت پایین
[54]	شباهت همسایگی در DBSCAN	زمان اجرای مناسب	دقت پایین

۳- الگوریتم پیشنهادی

به‌طورکلی، ساختار روش پیشنهادی از سه مرحله تشکیل‌شده است، شامل گروه‌بندی داده‌ها با استفاده از الگوریتم K-Means++، اجرای الگوریتم DBSCAN بر روی‌داده‌های هر گروه در مرحله قبل و ادغام خوشه‌های تولیدشده در گروه‌های مختلف. الگوریتم پیشنهادی Means Scan++ نام‌گذاری شده است، چراکه داده‌ها در ابتدا با استفاده از الگوریتم K-Means++ به K گروه تقسیم می‌شوند و سپس،

الگوریتم DBSCAN در هر گروه استفاده می‌شود. هر مرحله از الگوریتم به‌طور جداگانه در ادامه توضیح داده‌شده است. در شکل ۱، فلوچارت کلی از روش پیشنهادی نشان داده‌شده است.



شکل ۱- مراحل روش پیشنهادی

۳-۱- گروه‌بندی داده‌ها از طریق الگوریتم K-Means++

در این مرحله، الگوریتم K-Means++ روی کل مجموعه داده اجرا می‌شود تا داده‌ها به چند قسمت کوچک‌تر تقسیم شوند. ازآنجایی‌که الگوریتم DBSCAN بر اساس محاسبه چگالی هر قطعه داده عمل می‌کند و هدف آن قراردادن داده‌های نزدیک به یکدیگر در یک خوشه است، مانند سایر روش‌های خوشه‌بندی، داده‌ها را نمی‌توان به‌صورت دلخواه یا تصادفی تقسیم کرد تا کارایی مناسب به دست آید. درروش پیشنهادی از K-Means++ که یک الگوریتم خوشه‌بندی است، استفاده می‌شود. نتایج نشان می‌دهد که اجرای K-Means++ در تعداد کمی از تکرارها نیز می‌تواند تقسیم مناسبی را بر روی‌داده‌ها ارائه دهد، یافته‌ای که تعادل قابل قبولی را نیز نشان می‌دهد. تنها تفاوت بین K-Means++ و K-Means کلاسیک این است که مراکز اولیه خوشه در K-Means++ هوشمندانه‌تر از K-Means انتخاب می‌شوند. ازاین‌رو می‌تواند تقسیم‌بندی مناسب‌تری را پیشنهاد کند.

همان‌طور که قبلاً ذکر شد، تنها هدف از به‌کارگیری K-Means++ تقسیم داده‌ها است، کاری که می‌تواند بدون بار محاسباتی زیاد بر روی سیستم انجام شود، حتی زمانی‌که الگوریتم در تعداد کمی از تکرارها اجرا شود؛ مانند سایر روش‌های خوشه‌بندی، الگوریتم K-Means++ داده‌ها را خوشه‌بندی می‌کند. بااین‌حال در این مقاله، خوشه‌های K-Means++ همراه با عناوین گروه نام‌گذاری شده‌اند که با خوشه‌های به‌دست‌آمده از DBSCAN متفاوت باشند.

۳-۲- اجرای الگوریتم DBSCAN در هر گروه

در این مرحله، الگوریتم DBSCAN به‌طور جداگانه روی‌داده‌های هر یک از گروه‌های K-Means++ اجرا می‌شود. برای تعیین این‌که آیا یک نقطه یک هسته در DBSCAN است، باید فاصله آن تا کلیه نقاط دیگر مجموعه داده اندازه‌گیری شود. مقدار زیادی از محاسبات صرفاً برای به دست آوردن تعداد نقاط واقع در فاصله ε انجام می‌شود؛ بنابراین، تقسیم داده‌ها به گروه‌های کوچک‌تر و استفاده از DBSCAN به‌طور جداگانه در هر گروه می‌تواند به کاهش محاسبات موردنیاز برای اندازه‌گیری فاصله تا نقاط دیگر کمک نماید. فواصل یک نقطه تا کلیه نقاط دیگر در مجموعه داده باید اندازه‌گیری شود تا مشخص شود که آیا یک هسته است یا خیر. گروه‌بندی ارائه‌شده توسط K-Means++ باعث می‌شود که محاسبات سنگین به‌ویژه در مجموعه‌های داده بزرگ، به‌جای این‌که در کل مجموعه داده انجام شود، به گروه‌ها محدود شود.

به‌عنوان نمونه، اگر مجموعه داده‌ای متشکل از یک‌میلیون قطعه داده را از طریق K-Means++ به ده گروه تقسیم نماییم، تنها صدهزار محاسبه به‌طور متوسط به‌جای محاسبه یک‌میلیون فاصله زمانی که DBSCAN اجرا می‌شود، انجام می‌شود. اکنون باید مشخص شود که آیا خوشه‌های تولیدشده توسط DBSCAN در گروه‌های K-Means++ مجاور وجود دارد که باید ادغام شوند. از آن‌جایی‌که الگوریتم DBSCAN هر گروه را جداگانه در نظر می‌گیرد و خوشه‌بندی را تنها با در نظرگرفتن داده‌های همان گروه انجام می‌دهد، خوشه‌های یک گروه باید همراه با گروه‌های مجاور بررسی شوند؛ بنابراین، اگر هیچ مرزی بین گروه‌ها وجود نداشته باشد، DBSCAN می‌تواند داده‌ها را در گروه‌های مجاور در هنگام انجام خوشه‌بندی مشاهده نماید.

۳-۳- ادغام خوشه‌های مرزی انتخاب‌شده

در این مرحله باید بررسی شود که آیا خوشه‌های به‌دست‌آمده از الگوریتم DBSCAN درصورتی‌که مرزهای گروهی بین داده‌ها وجود نداشته باشد، یا این‌که خوشه‌های مرزی به یکی ادغام می‌شوند. برای کاهش محاسبات، در این مرحله نیز دو دور هرس برای مواردی که نیاز به بررسی دارند، پیشنهاد می‌شود. دور اول، شامل تجزیه‌وتحلیل فواصل بین گروه‌های K-Means++ و اعمال هرس برای گروه‌های فواصل طولانی (بیش‌از ε) است که ادغام خوشه‌های داخلی را غیرممکن می‌کند. دور دوم، شامل هرسی می‌شود که وقتی خوشه‌های داخلی دو گروه برای ادغام و حذف خوشه‌هایی که ادغام آن‌ها غیرممکن است، بررسی می‌شود. پس‌ازآنجام این دو دور هرس، گروه‌های انتخاب‌شده توسط الگوریتم DBSCAN ادغام می‌شوند و دوباره بر روی‌داده‌های این خوشه‌ها اجرا می‌شود.

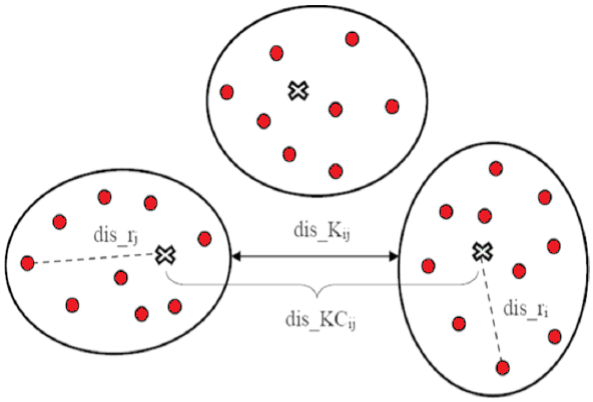
بنابراین، اگر فاصله بین دو مجموعه بیشتر از ε باشد، قطعاً هیچ نقطه‌ای از مجموعه دوم در همسایگی ε نقاط مجموعه اول قرار نخواهد گرفت و بالعکس (مقدار ε ثابت و برابر ۰.۰۱ درنظر گرفته‌شده است). بدیهی است که در صورت پیاده‌سازی DBSCAN روی‌داده‌های این دو مجموعه، ایجاد یک خوشه غیرممکن خواهد بود. سپس روش ادغام خوشه‌های مرزی تعیین‌شده در سه مرحله توضیح داده می‌شود. فواصل بین مجموعه‌ها در مرحله اول تعیین می‌شود و اگر فاصله آن‌ها از ε بیشتر باشد، مجموعه‌ها هرس می‌شوند. این اولین فرآیند هرس در این مرحله است. پس‌ازآن، فواصل بین خوشه‌های موجود در مجموعه‌های باقی‌مانده از فرآیند هرس اول، دوبه‌دو محاسبه می‌شوند. اگر فاصله بین دو خوشه از ε بیشتر شود، فرآیند هرس دوم اتفاق می‌افتد. درنهایت، DBSCAN دوباره از داده‌های خوشه‌های تعیین‌شده در مرحله سوم اجرا می‌شود.

مرحله اول: بررسی فواصل بین گروه‌ها است و این هرس برای گروه‌های K-Means++ قبل از بررسی ادغام خوشه‌ها اعمال می‌شود. بر اساس ایده ارائه‌شده در این مقاله، امکان ادغام باید برای خوشه‌های واقع در گروه‌های مجاور بررسی شود. بدیهی است که ادغام خوشه‌های جفت گروه‌هایی که به‌جای همسایگی از یکدیگر دور هستند، غیرممکن است و این گروه‌ها را می‌توان از فرآیند محاسبات برای بررسی امکان ادغام خوشه‌های گروهی حذف نمود. لذا، گروه‌های به‌دست‌آمده از الگوریتم K-Means++ دوبه‌دو در نظر گرفته‌شده است و معادله (۱) برای آن‌ها محاسبه می‌شود.

$$dis\_K_{ij} = dis\_KC_{ij} - (dis\_r_i + dis\_r_j) \quad (۱)$$

که در آن، dis_K_{ij} نشان‌دهنده فاصله بین مراکز دو گروه در K-Means++ و dis_r_i و dis_r_j نشان‌دهنده فواصل بین مراکز دو گروه و دورترین نقاط روی آن‌ها

است. معادله فاصله بین دو گروه یعنی dis_K_{ij} را با در نظرگرفتن بدترین حالت به دست می‌آورد. هنگامی‌که الگوریتم K-Means++ اعمال می‌شود، می‌توان این پارامترها را برای هر گروه بدون محاسبات قابل‌توجهی محاسبه و ذخیره نمود. فواصل به‌صورت شماتیک در شکل ۲ نشان داده‌شده‌اند. اگر مقدار dis_K_{ij} بزرگ‌تر از ε باشد، قطعاً بعید است که خوشه‌های مرزی دو گروه ادغام شوند. بنابراین، نیازی به بررسی خوشه‌های داخلی کلیه گروه‌ها برای ادغام نیست.



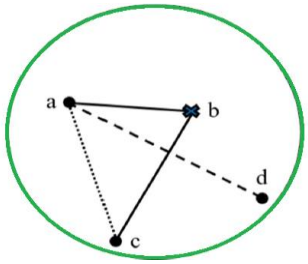
شکل ۲: شماتیکی از نحوه محاسبه فاصله بین دو گروه [۵۵]

فقط گروه‌هایی که در فواصل کمتر یا مساوی ε از یکدیگر قرار دارند، موردبررسی قرار می‌گیرند. این شرایط باعث می‌شود که بسیاری از محاسبات هرس شوند. به‌طورکلی، این هرس دومین موردی است که برای الگوریتم DBSCAN در الگوریتم پیشنهادی پس از تقسیم داده‌ها به گروه‌های کوچک‌تر اعمال می‌شود.

مرحله دوم: تجزیه‌وتحلیل احتمال ادغام خوشه‌های DBSCAN است. در این مرحله، امکان ادغام خوشه‌های داخلی برای گروه‌هایی که در مرحله قبل هرس نشده‌اند، تحلیل می‌شود. فواصل بین خوشه‌ها باید برای یافتن آن‌هایی که می‌توانند ادغام شوند، به دست آید. اگر فاصله کوچک‌تر یا مساوی ε باشد، ادغام می‌تواند اتفاق بیافتد، در غیر این‌صورت، احتمال صفر است و داده‌های مربوط به آن خوشه‌ها در مرحله ادغام در نظر گرفته نمی‌شوند. برای اعمال این هرس، باید یک مرکز برای هر خوشه DBSCAN در نظر گرفته شود که بر اساس داده‌های میانگین در خوشه محاسبه می‌شود. لازم به ذکر است که مرکز برای کلیه خوشه‌ها محاسبه نمی‌شود و در مقایسه با محاسباتی که بعداً هرس می‌شود، پیچیدگی محاسباتی بالایی را تحمیل نمی‌کند.

پارامتر دیگری که برای اعمال هرس موردنیاز است، فاصله بین مرکز خوشه و دورترین قطعه داده در خوشه است. پارامتر را می‌توان با تغییر جزئی در الگوریتم DBSCAN و بدون بار محاسباتی قابل‌توجه‌ای محاسبه نمود. همان‌طور که ذکر شد، DBSCAN از یک نقطه تصادفی شروع می‌شود و بررسی می‌کند که آیا آن نقطه یک هسته است یا خیر. اگر نقطه به‌عنوان یک هسته شناسایی شود، الگوریتم این روش را در نقاطی در فاصله ε از هسته تکرار می‌کند. به این معنا که ابتدا داده‌هایی که در فاصله حداکثر ε قرار دارند و سپس، کلیه داده‌هایی که در فاصله ε قرار دارند، بررسی می‌شوند. این روند تا زمانی ادامه می‌یابد که کل خوشه مشخص شود. بااین‌حال، هنگام اجرای الگوریتم می‌توان فاصله داده‌هایی را که در آن محاسبه می‌شود، ذخیره نمود و بنابراین این می‌تواند حد بالایی برای فاصله بین نقطه شروع و دورترین فاصله از آن باشد.

شکل ۳ یک خوشه DBSCAN را نشان می‌دهد. فرض کنید که تشکیل خوشه به‌صورت تصادفی در نقطه a آغازشده است. دورترین نقطه از a، d نام دارد که قطعاً یک نقطه مرزی نیز است و بنابراین، این می‌تواند حد بالایی برای فاصله بین نقطه شروع و دورترین نقطه از آن باشد.



شکل ۳: شماتیکی از نحوه محاسبه فاصله بین مرکز خوشه و دورترین نقطه (bc)

بر اساس فرآیند ذکرشده، کران بالای d را می‌توان با استفاده از DBSCAN با بار محاسباتی غیرقابل ملاحظه‌ای محاسبه کرد (ad با خط‌چین نشان داده‌شده است). برای محاسبه فاصله بین مرکز خوشه b و دورترین نقطه از آن‌که به‌صورت c نشان داده‌شده است، یک خط از a به c ترسیم کرده و مثلث abc تشکیل می‌شود. با توجه به مثلث فوق، قضیه نابرابری مثلث به‌صورت معادله (۲) نشان داده می‌شود [۵۵]:

$$bc < ab + ac \qquad (۲)$$

اگر مرکز خوشه را b فرض کنیم، می‌توان فاصله بین نقطه a و مرکز b را محاسبه نمود. علاوه بر این، ازآنجایی‌که d دورترین نقطه از a است، معادله (۳) صادق است [۵۵].

$$ac \leq ad \qquad (۳)$$

با توجه به مقدار نامعلوم ac و برای جلوگیری از محاسبات اضافی، معادله (۲) را می‌توان به‌صورت معادله (۴) و بر اساس معادله (۳) بازنویسی کرد [۵۵].

$$bc < ab + ad \qquad (۴)$$

از معادله (۴) مشخص است که مقدار bc قطعاً کمتر از مجموع ab و ad است. بنابراین، کران بالایی برای مقدار bc، یعنی فاصله بین مرکز خوشه و دورترین قطعه داده در خوشه را می‌توان بر اساس معادله (۴) محاسبه کرد [۵۵]. با درنظر گرفتن دو خوشه از دو گروه، مانند شکل (۴)، از معادله (۵) برای به دست آوردن فاصله بین دو خوشه استفاده می‌شود [۵۵].

$$dis\_D_{ij} = dis\_DC_{ij} - (dis\_r_i + dis\_r_j) \qquad (۵)$$

در معادله (۵)، dis_r_i و dis_r_j نشان‌دهنده فواصل بین مراکز خوشه و دورترین نقاط از آن‌ها در همان خوشه‌ها، dis_D_{ij} نشان‌دهنده فاصله بین دو خوشه DBSCAN و dis_DC_{ij} نشان‌دهنده فاصله بین مراکز دو خوشه DBSCAN در دو گروه K-Means++ است. اگر مقدار dis_DC_{ij} کمتر یا مساوی با مقدار ε باشد، این دو خوشه را می‌توان ادغام نمود و در غیر این‌صورت هرس می‌شوند. فقط گروه‌هایی که در فواصل کمتر یا مساوی ε از یکدیگر قرار دارند موردبررسی قرار می‌گیرند. این شرایط باعث می‌شود که بسیاری از محاسبات هرس شوند.

مرحله سوم: خوشه‌های مرزی با اجرای مجدد الگوریتم DBSCAN ادغام می‌شوند. پس از مشخص شدن امکان ادغام خوشه‌ها و انجام هرس پیشنهادی، داده‌های مربوط به آن‌هایی که می‌توانند ادغام شوند، با یکدیگر در نظر گرفته می‌شوند و الگوریتم DBSCAN بدون توجه به هر مرزی برای آن‌ها دوباره اعمال می‌شود. سپس، یا خوشه‌ها به همان ترتیب ادامه می‌دهند یا دو یا چند تا از آن‌ها در یک خوشه ادغام می‌شوند. بنابراین، کل فضای داده درنهایت توسط الگوریتم DBSCAN خوشه‌بندی می‌شود.

پیچیدگی زمانی روش پیشنهادی با در نظر گرفتن پیچیدگی هر سه مرحله آن محاسبه‌شده است. معادله (۶)، پیچیدگی زمانی رویکرد پیشنهادی را به‌صورت ترکیبی از پیچیدگی بیان می‌کند، با این تفاوت که محاسباتی که انجام می‌شوند بسیار پیچیده‌تر از محاسبات سنگین در DBSCAN استاندارد می‌باشند.

$$O((iter * k * n) + \left(\frac{n}{k}\right)^2 + \frac{k(k-1)}{2}m^2) \qquad (۶)$$

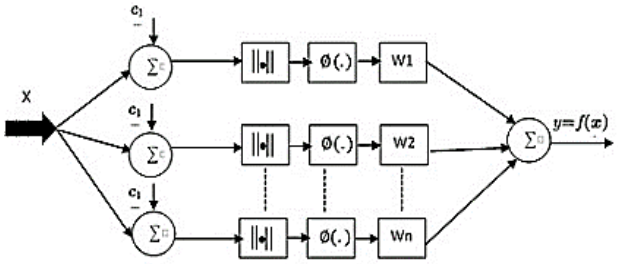
در این مراحل، ازآنجایی‌که فواصل از k نقطه مرکزی تا n نقطه باید در الگوریتم K-Means++ به دست آید و محاسبات پارها تکرار می‌شوند، پیچیدگی زمانی

الگوریتم، $O((iter * k * n)$ است. تعداد تکرارها، تعداد مراکز اولیه و اندازه داده بر زمان اجرای الگوریتم تأثیر می‌گذارد. پیچیدگی زمانی الگوریتم DBSCAN نیز $O(n^2)$ است، زیرا باید فواصل بین نقاط در n جفت به دست آید. با توجه به مراحل ذکرشده و الگوریتم توسعه‌یافته برای روش پیشنهادی، پیچیدگی زمانی الگوریتم پیشنهادی، همان‌طور که در معادله (۶) بیان‌شده است، از سه بخش تشکیل‌شده است.

ازآنجایی‌که الگوریتم K-Means++ برای اولین بار اجرا می‌شود و داده‌ها را گروه‌بندی می‌کند، اولین بخش پیچیدگی زمانی مربوط به K-Means++ است. بخش دوم مربوط به الگوریتم DBSCAN است که به‌طور جداگانه برای داده‌های موجود در k گروه اعمال می‌شود. بخش سوم پیچیدگی زمانی به مرحله نهایی درروش پیشنهادی می‌پردازد. در این مرحله از الگوریتم پیشنهادی، خوشه‌های موجود در گروه‌ها دوبه‌دو برای انتخاب خوشه‌های قابل ادغام جستجو می‌شوند. سپس، DBSCAN به داده‌های این گروه‌های انتخاب‌شده، اعمال می‌شود. بنابراین، واضح است که پیچیدگی زمانی الگوریتم ارائه‌شده کمتر از DBSCAN است و زمانی‌که الگوریتم پیشنهادی روی مجموعه داده‌های مختلف اجرا می‌شود، به‌ویژه در داده‌های بزرگ، کاهش می‌یابد، چراکه ابتدا گروه‌بندی صورت پذیرفته و سپس، DBSCAN بر روی هر گره انجام می‌شود.

۴-۳- مدل پیشنهادی

شبکه عصبی تابع پایه شعاعی به دلیل سادگی ساختار و همچنین روش کارآمد آموزش آن، موردتوجه بسیاری قرارگرفته است. شبکه عصبی RBF قابلیت تخمین همه‌منظوره را دارد، بنابراین شبکه عصبی RBF می‌تواند برای مسائل درون‌یابی استفاده شود. یک شبکه عصبی تابع پایه شعاعی گوسی، فرم غیر نرمالیزه‌شده تابع توزیع گوسی و بسیار غیرخطی است و ویژگی‌های مناسبی برای یادگیری افزایشی دارد. شبکه‌های عصبی گوسی برای یادگیری نگاشت‌های پیچیده تعریف‌شده‌اند، اما در شبکه عصبی تابع پایه شعاعی، انتخاب مراکز تابع گوسی اهمیت دارد و این کار با استفاده از الگوریتم K-Means انجام می‌شود. در مواجه با داده‌های حجیم، الگوریتم خوشه‌بندی باید بتواند مراکز خوشه‌ها را با دقت و سرعت‌بالاتری به‌جای استفاده از الگوریتم K-Means به همراه داشته باشد. در مدل پیشنهادی بنام Means Scan++ در تعیین مراکز خوشه‌های گوسی در شبکه عصبی تابع پایه شعاعی استفاده می‌شود. یک شبکه RBF از دو لایه کاملاً متصل به یکدیگر بانام‌های لایه پنهان یا لایه RBF و لایه خروجی مطابق شکل ۴ تشکیل می‌شود.



شکل ۴: معماری شبکه RBF [56]

مطابق شکل ۴، لایه ورودی شبکه عصبی RBF به‌طور مستقیم به لایه پنهان متصل می‌شود و خروجی نورون j ام لایه پنهان به‌صورت معادله (۷) به دست می‌آید [۵۶]:

$$h_j = \frac{\varphi(\|X - C_j\|)}{\sigma_j} \quad (۷)$$

که در آن، h_j خروجی نورون j ام، φ تابع عملکرد غیرخطی RBF، X بردار ورودی، C_j مرکز نورون و σ_j میزان گستردگی مرکز نورون است. در مدل پیشنهادی، C_j توسط روش خوشه‌بندی پیشنهادی انجام می‌شود. عملکرد غیرخطی شبکه عصبی تابع پایه شعاعی به دلیل تابع عملکرد φ است. نورون‌ها در لایه خروجی، دارای تابع عملکرد خطی هستند و خروجی‌های شبکه عصبی تابع پایه شعاعی برابر مجموع وزن‌دار شده خروجی‌های نرون‌های لایه پنهان متصل به لایه خروجی هستند. درواقع، خروجی y_k نورون k ام در لایه خروجی به‌صورت معادله (۸) محاسبه می‌شود [۵۶]:

$$y_k = \sum_{j=1}^{m} W_{kj}\, h_j \quad (۸)$$

که در آن، W_{kj} وزن اتصال نورون j ام لایه پنهان و نورون k ام لایه خروجی و m تعداد نرون‌های لایه پنهان است. پارامترهای قابل تنظیم در فرآیند آموزش شبکه عصبی تابع پایه شعاعی متعدد هستند و کارایی شبکه در بعضی از موارد به‌شدت وابسته به این پارامترها و روش تطبیق آن‌ها است. این پارامترها مراکز توابع گوسی، میزان گستردگی آن‌ها و وزن‌های اتصالات طبقه خروجی می‌باشند. یک مسئله مهم نیز تعداد نرون‌های لایه پنهان است که باید به روش مناسبی آن را به دست آورد. نکته مهم این است که لایه‌های متفاوت یک شبکه RBF، وظایف متفاوتی انجام می‌دهند و منطقی است که روش‌های جداگانه‌ای برای تطبیق و آموزش هر یک در نظر گرفته شود.

بنابراین راه‌کار پیشنهادی این مقاله، متمرکز بر روی تعیین خوشه‌ها و مراکز خوشه با دقت و سرعت مناسب است تا شبکه عصبی تابع پایه شعاعی بتواند در مواجه با داده‌های حجیم به‌خوبی عمل نماید.

۴- ارزیابی نتایج آزمایش

در این بخش، ابتدا تنظیمات و محیط سخت‌افزاری و شبیه‌سازی آزمایش و سپس، مجموعه داده‌های آزمون در آزمایش تشریح می‌شوند و درنهایت، نتایج حاصل‌شده در آزمایش ارزیابی می‌گردند.

۴-۱- تنظیمات و محیط سخت‌افزاری و شبیه‌سازی آزمایش

به‌منظور شبیه‌سازی آزمایش، محیط سخت‌افزار شامل رایانه‌ای با حافظه ۸ گیگابایت و پردازنده سه هسته‌ای اینتل و سیستم‌عامل ویندوز ۱۰ استفاده‌شده است. به‌علاوه، آزمایش‌ها در محیط نرم‌افزاری MATLAB-R2019b پیاده‌سازی شده است. با توجه به محاسبه زمان اجرا، کلیه روش‌ها بر روی این سیستم سخت‌افزاری و نرم‌افزاری پیاده‌سازی شده و نتایج آن ارزیابی‌شده است.

۴-۲- مجموعه داده‌های آزمون در آزمایش

در آزمایش‌ها، از چهار مجموعه داده آزمون استفاده‌شده است که شامل مجموعه داده Image segmentation دارای ۲۳۱۰ نمونه، Pendigit دارای ۱۰۹۹۲ نمونه، Letters دارای ۲۰۰۰۰ نمونه و Shuttle control دارای ۵۸۰۰۰ نمونه می باشند [۵۶]. این چهار مجموعه داده از پایگاه داده‌های UCI برداشته‌شده است [۵۶]. خصیصه‌های این چهار مجموعه داده آزمون در جدول ۲ نشان داده است.

جدول ۲: مشخصات مجموعه داده‌های آزمون

مجموعه داده	تعداد نمونه	تعداد کلاس	تعداد ویژگی‌ها	نوع ویژگی‌ها
Image segmentation	2310	7	19	عدد حقیقی
Pendigit	10992	44	16	عدد صحیح
Letters	20000	۴۰	16	عدد صحیح
Shuttle control	۵۸۰۰۰	۲	۱۵	اسمی

۴-۳- معیارهای ارزیابی آزمایش

در کلیه آزمایش‌ها، از روش اعتبارسنجی K-fold با K=10 استفاده‌شده است و نتایج به‌صورت میانگین آمده است. برای ارزیابی سرعت، از ثانیه و برای دقت دسته‌بندی،

از معیار خطای MSE استفاده‌شده است که در معادله (۹) آمده است و که در آن، پارامتر Y_i خروجی واقعی، $\hat{Y}_i$ خروجی مدل است و n تعداد نمونه‌ها است.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \quad (۹)$$

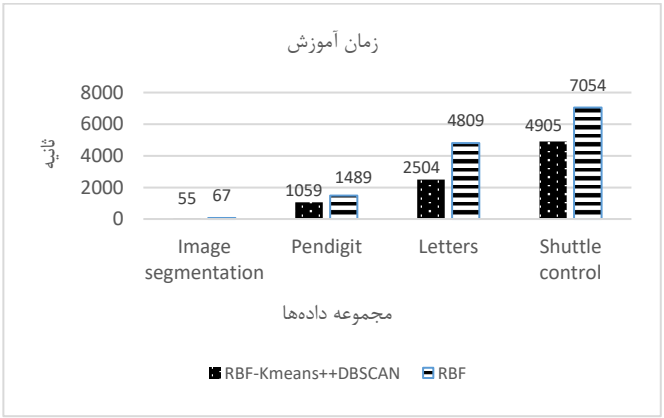
۴-۴- نتایج آزمایش

نتایج عددی برای بررسی عملکرد مدل خوشه‌بندی پیشنهادی با شبکه عصبی تابع پایه شعاعی استاندارد که با الگوریتم K-Means خوشه‌بندی می‌کند، مقایسه شده است. در جدول ۳، مقایسه زمان آموزش و میزان خطا در آزمایش مدل‌های شبکه عصبی بهبودیافته و مدل استاندارد آمده است.

جدول ۳: مقایسه نتایج مدل پیشنهادی

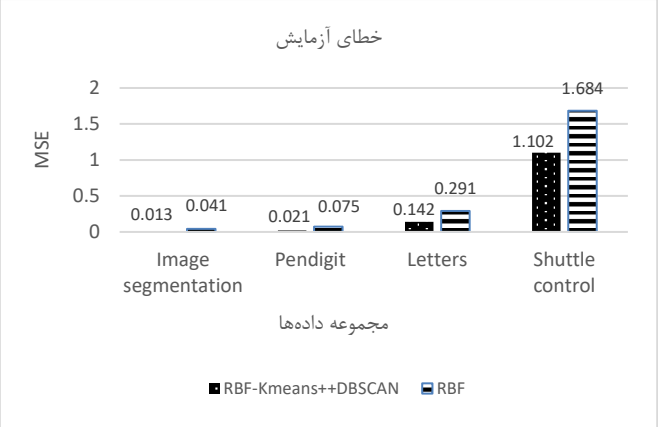
مدل	روش پیشنهادی RBF K-Means++ DBScan		روش استاندارد RBF	
مجموعه داده	زمان (s) در آموزش	خطا (MSE) در آزمایش	زمان (s) در آموزش	خطا (MSE) در آزمایش
Image segmentation	45	٪0.013	67	٪0.041
Pendigit	1059	٪0.021	1489	٪0.075
Letters	2504	٪0.142	4809	٪0.291
Shuttle control	4905	٪1.102	7054	٪1.684

در شکل ۵، مدت زمان آموزش برای مجموعه داده‌های آزمایش‌شده، آمده است که نشان می‌دهد با مدل خوشه‌بندی پیشنهادی، زمان آموزش کاهش‌یافته است و هرچه مجموعه داده حجیم‌تر باشد، کاهش زمان آموزش محسوس‌تر است. مدل خوشه‌بندی پیشنهادی توانسته است که با خوشه‌بندی سریع و دقیق به شبکه عصبی تابع پایه شعاعی کمک کند تا آموزشی در زمان کمتر و دقت دسته‌بندی بالاتری در آزمایش داشته باشد. می‌توان بیان نمود که در مجموعه داده Image segmentation، الگوریتم شبکه عصبی تابع پایه شعاعی پیشنهادی ۴۵ ثانیه زمان آموزش داشته است، درصورتی‌که نسبت به شبکه عصبی تابع پایه شعاعی استاندارد میزان ۱۲ ثانیه کمتر است. همچنین، خطای مدل پیشنهادی برای فرآیند آزمایش به‌اندازه ۰.۰۲۸٪ نسبت به شبکه عصبی تابع پایه شعاعی استاندارد کمتر بوده است. برای مجموعه دادهPendigit، مدل خوشه‌بندی پیشنهادی نیاز به ۱۰۵۹ ثانیه آموزش داشته است که به میزان ۴۳۰ ثانیه از مدل شبکه عصبی تابع پایه شعاعی استاندارد کمتر است. به‌علاوه، خطای مدل خوشه‌بندی پیشنهادی در آزمایش به میزان ۰.۰۵۴٪ کمتر بوده است.



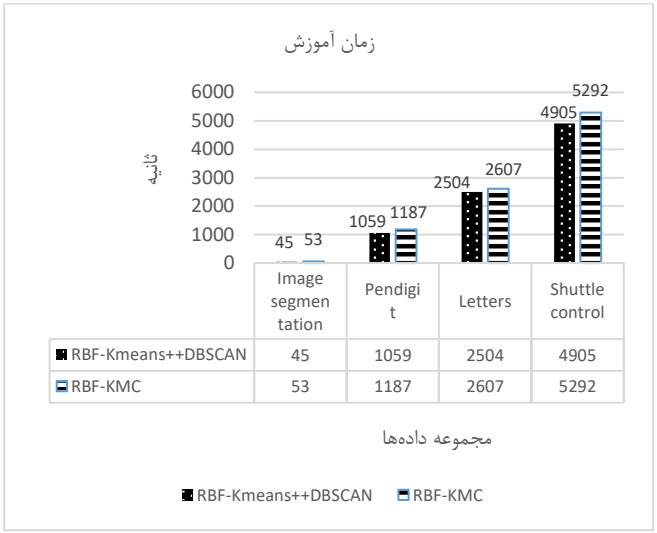
شکل ۵: مقایسه زمان مدل پیشنهادی با الگوریتم RBF استاندارد

در شکل ۶، میزان خطای آزمایش برای مجموعه داده‌های آزمون آمده است که نشان می‌دهد با مدل خوشه‌بندی پیشنهادی، میزان خطا در شبکه عصبی تابع پایه شعاعی کاهش‌یافته است، زیرا مراکز تابع گوسی در این شبکه عصبی به‌درستی تنظیم‌شده است. در مجموعه داده‌های Letters و Shuttle control که تعداد نمونه‌های بیشتری را دارند، نتایج اختلاف بیشتری را نشان می‌دهد. در مجموعه داده Letters، برای مدل خوشه‌بندی پیشنهادی زمان لازم برای آموزش ۲۵۰۴ ثانیه بوده است، اما در شبکه عصبی استاندارد در حدود دو برابر و به میزان ۴۸۰۹ ثانیه است. میزان خطا در این مجموعه داده برای مدل خوشه‌بندی پیشنهادی در حدود ۰.۱۴۹ ٪ خطای کمتری از روش استاندارد است. در مجموعه داده Shuttle control، برای مدل خوشه‌بندی پیشنهادی زمان لازم برای آموزش ۴۹۰۵ ثانیه بوده است، اما در شبکه عصبی استاندارد ۷۰۵۴ ثانیه است. میزان خطا در این مجموعه داده برای مدل خوشه‌بندی پیشنهادی در حدود ۰.۵۸۲٪ خطای کمتری از مدل شبکه عصبی استاندارد بوده است.



شکل ۶: مقایسه خطای مدل پیشنهادی با روش RBF استاندارد

مقایسه مدل خوشه‌بندی پیشنهادی با مدل بهبودیافته شبکه عصبی تابع پایه شعاعی، با بهبود خوشه‌بندی الگوریتم K-Means سراسری تغییریافته سریع [۵۶] بنام RBF-KMC انجام‌شده است. در شکل‌های ۷ و ۸ نشان می‌دهد که مدل خوشه-بندی پیشنهادی در مجموعه داده‌های مشترک توانسته است که زمان آموزش کمتر و دقت بیشتری را در آزمایش داشته باشد.

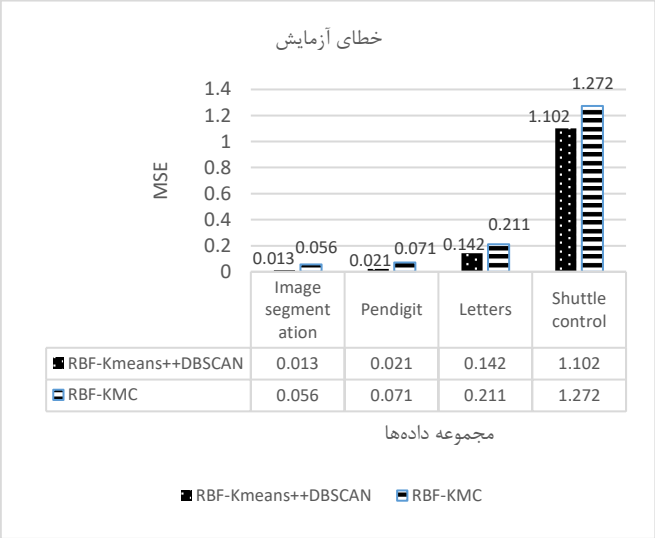


شکل ۷: مقایسه زمان آموزش مدل پیشنهادی با الگوریتم RBF-KMC

در شکل ۷، میزان خطای آزمایش مدل خوشه‌بندی پیشنهادی با الگوریتم خوشه‌بندی K-Means سراسری تغییریافته مقایسه شده است که نشان می‌دهد که خوشه‌بندی، میزان خطا در شبکه عصبی تابع پایه شعاعی را در هر دو روش کاهش داده است، زیرا مراکز تابع گوسی با دقت بهتری مشخص‌شده است، اما در مدل خوشه‌بندی پیشنهادی میزان خطای پایین‌تری به‌دست‌آمده است. همان‌طور که در شبیه‌سازی مدل پیشنهادی در شکل‌های ۷ و ۸ مشاهده می‌شود، می‌توان بیان نمود که در مجموعه داده Image segmentation، الگوریتم شبکه عصبی تابع پایه

شعاعی پیشنهادی ۴۵ ثانیه زمان آموزش داشته است، درصورتی‌که نسبت به الگوریتم RBF-KMC میزان ۸ ثانیه کمتر است. به‌علاوه، خطای مدل

خوشه‌بندی پیشنهادی برای فرآیند آزمایش به‌اندازه ۰.۰۴۳٪ نسبت به الگوریتم RBF-KMC کمتر بوده است. برای مجموعه داده Pendigit، مدل خوشه-بندی پیشنهادی نیاز به ۱۰۵۹ ثانیه آموزش داشته است که به میزان ۱۱۸ ثانیه از الگوریتم RBF-KMC کمتر است. همچنین، خطای مدل خوشه‌بندی پیشنهادی به میزان ۰.۰۵٪ از الگوریتم RBF-KMC کمتر بوده است.



شکل ۸: مقایسه خطای مدل پیشنهادی با الگوریتم RBF-KMC

در مجموعه داده Letters، برای مدل خوشه‌بندی پیشنهادی زمان لازم برای آموزش ۲۵۰۴ ثانیه بوده است، اما در الگوریتم RBF-KMC در حدود ۲۶۰۷ ثانیه است. میزان خطا در این مجموعه داده برای مدل خوشه‌بندی پیشنهادی در حدود ۰.۰۷۱٪ خطای کمتری نسبت به الگوریتم RBF-KMC است. در مجموعه داده Shuttle control، برای مدل خوشه‌بندی پیشنهادی زمان لازم برای آموزش ۴۹۰۵ ثانیه بوده است، اما در شبکه عصبی استاندارد ۵۲۹۲ ثانیه است. میزان خطا در این مجموعه داده برای مدل خوشه‌بندی پیشنهادی در حدود ۰.۱۵٪ خطای کمتری نسبت به الگوریتم RBF-KMC بوده است.

در جداول ۴ و ۵، الگوریتم‌های پیشنهادشده در مرجع [۵۷] شامل K-DBSCAN و HCA-DBSCAN با مدل خوشه‌بندی پیشنهادی مقایسه شده است. همان‌طور که در جدول ۴ مشاهده می‌شود، مدل خوشه‌بندی K-DBSCAN زمان آموزش بیشتری نسبت به مدل خوشه‌بندی پیشنهادی دارد و دقت آن نیز اختلاف زیادی با نتایج مدل خوشه‌بندی پیشنهادی ندارد.

جدول ۴: مقایسه نتایج مدل پیشنهادی با الگوریتم RBF K-DBSCAN

مدل	روش پیشنهادی RBF K-Means++ DBScan		روش استاندارد RBF K-DBSCAN	
مجموعه داده	زمان (s) در آموزش	خطا (MSE) در آزمایش	زمان (s) در آموزش	خطا (MSE) در آزمایش
Image segmentation	45	٪0.013	52	٪0.013
Pendigit	1059	٪0.021	1163	٪0.022
Letters	2504	٪0.142	2564	٪0.145
Shuttle control	4905	٪1.102	4926	٪1.112

در جدول ۵ نیز مدل خوشه‌بندی HCA-DBSCAN دارای دقت خوبی است که نزدیک به مدل خوشه‌بندی پیشنهادی عمل کرده است، اما زمان آموزش بالاتری داشته است. میزان بهبود نتایج در مدل خوشه‌بندی پیشنهادی نسبت به مدل بهبودیافته شبکه عصبی تابع پایه شعاعی با بهبود خوشه‌بندی الگوریتم K-Means سراسری تغییریافته سریع [۵۶]، نشان می‌دهد که مدل خوشه‌بندی پیشنهادی در مجموعه داده‌های مشترک توانسته است که زمان آموزش کمتر و همچنین دقت بیشتری را به همراه داشته باشد.

مدل	روش پیشنهادی RBF K-Means++ DBScan		روش استاندارد RBF	
مجموعه داده	زمان (s) در آموزش	خطا (MSE) در آزمایش	زمان (s) در آموزش	خطا (MSE) در آزمایش
Image segmentation	45	٪0.013	63	٪0.016
Pendigit	1059	٪0.021	1352	٪0.026
Letters	2504	٪0.142	3805	٪0.149
Shuttle control	4905	٪1.102	6552	٪1.114

جدول ۵: مقایسه نتایج مدل پیشنهادی با روش RBF HCA-DBSCAN

۵- نتیجه‌گیری و کارهای آینده

در این مقاله، یک مدل شبکه عصبی تابع پایه شعاعی جدید با بهبود خوشه‌بندی در مرحله بدون نظارت این شبکه ارائه شد. در خوشه‌بندی مطرح‌شده که مناسب برای داده‌هایی با تعداد بالا می‌باشند، K-Means++ در کنار DBSCAN قرار گرفت که توانسته است در مواجهه با داده‌های بزرگ، سرعت در خوشه‌بندی و دقت در دسته‌بندی را برای شبکه عصبی تابع پایه شعاعی به همراه داشته باشد. نتایج بر روی چهار مجموعه داده با ویژگی‌های متنوع و تعداد نمونه‌های مختلفی آزمایش شد و ارزیابی نتایج نشان داد که الگوریتم پیشنهادی توانسته است نسبت به روش استاندارد شبکه عصبی تابع پایه شعاعی، سرعت آموزش و دقت دسته‌بندی را بهبود دهد.

به‌عنوان کارهای آینده، می‌توان به دو مورداشاره نمود. در بهبود مدل پیشنهادی، استفاده از خوشه‌بندی بدون نظارت مبتنی بر انبوه داده‌ها می‌تواند با روش‌های دیگری بررسی شود، زیرا در مدل پیشنهادی تأثیر زیادی بر عملکرد مدل داشته است. به‌علاوه، استفاده از مدل پیشنهادی در کاربردهای دنیای واقعی نیز می‌تواند آزمایش شود.

۶- مآخذ

[1] V. Storey and I. Song, "Big data technologies and management: What conceptual modeling can do," *Data Knowledge Engineering*, no. 108, pp. 50–67, 2017.

[2] M. Ianni, E. Masciari, G. Mazzeo, and C. Zaniolo, "Eficient big data clustering," *22nd International Database Engineering and Applications Symposium*, ACM, pp. 103–109, 2018.

[3] P. Arora, D. Deepali, and S. Varshney, "Analysis of K-Means and K-Medoids algorithm for big data," *Proceding Computer Science*, no. 78, pp. 507–512, 2016.

[4] A. Jain, M. Murty, and P. Flynn, "Data clustering: a review," *ACM Computer Survey*, vol. 31, no. 3, pp. 264–323, 1999.

[5] J. Zhu, M. Zeng, J. Huang, S. Liao, C. Cai, and L. Zheng, "Vehicle re-identifcation using quadruple directional deep learning features," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 410–420, 2020.

[6] S. Liu, M. Liu, P. Li, J. Zhao, Z. Zhu, and X. Wang, "SAR image denoising via sparse representation in Shearlet domain based on continuous cycle spinning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 5, pp. 2985–2992, 2017.

[7] S. Pei, T. Shen, X. Wang, C. Gu, Z. Ning, X. Ye, and N. Xiong, "3DACN: 3D augmented convolutional network for time series data," *Information Scieince*, no. 513, pp. 17–29, 2020.

[8] S. Qiao, T. Li, H. Li, J. Peng, and H. Chen, "A new blockmodeling based hierarchical clustering algorithm for web social networks," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 3, pp. 640–647, 2012.

[9] D. Che, M. Safran, and Z. Peng, "From big data to big data mining: challenges, issues, and opportunities," *International Conference on Database Systems for Advanced Applications*, Springer, pp. 1–15, 2013.

[10] S. Mirjalili, "Evolutionary Radial Basis Function Networks," *Studies in Computational Intelligence*, pp. 105-139, 2019.

[11] A. Osmanović, S. Halilović, L. A. Ilah, A. Fojnica, and Z. Gromilić, "Machine learning techniques for classification of breast cancer," *IFMBE Proceedings*, 2019.
[12] M. Ester, H. Kriegel, J. Sander, X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *2nd International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.
[13] A. David, and V. Sergei, "K-Means++: the advantages of careful seeding," *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, ACM, pp. 1027–1035, 2007.
[14] A. Katal, M. Wazid, and R. Goudar, "Big data: Issues, challenges, tools and good practices," *6th International Conference on Contemporary Computing*, IEEE, pp. 404–409, 2013.
[15] S. Shahrivari, "Beyond batch processing: Towards real-time and streaming big data," *Computers*, vol. 3, no. 4, pp. 117–129, 2014.
[16] S. Mirjalili, "Evolutionary Radial Basis Function Networks," *Studies in Computational Intelligence*, pp. 105-139, 2019.
[17] C. C. Liao, "Genetic K-Means algorithm-based RBF network for photovoltaic MPP prediction," *Energy*, 2010.
[18] M. Diez, S. Volpi, A. Serani, F. Stern, and E. F. Campana, "Simulation-Based Design Optimization by Sequential Multi-criterion Adaptive Sampling and Dynamic Radial Basis Functions," *Computational Methods in Applied Sciences*, 2019.
[19] M. Smolik, V. Skala, and Z. Majdisova, "Advances in Engineering Software Vector field radial basis function approximation," *Advances in Engineering Software*, vol. 123, no. 17, pp. 117-129, 2018.
[20] M. Diez, S. Volpi, A. Serani, F. Stern, and E. F. Campana, "Simulation-Based Design Optimization by Sequential Multi-criterion Adaptive Sampling and Dynamic Radial Basis Functions", *Computational Methods in Applied Sciences*, 2019.
[21] T. Wangchamhan, S. Chiewchanwattana, and K. Sunat, "Efficient algorithms based on the k-means and Chaotic League Championship Algorithm for numeric, categorical, and mixed-type data clustering," *Expert Systems with Applications*, 2017.
[22] Y. Li, X. Wang, S. Sun, X. Ma, and G. Lu, "Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks," *Transportation Research Part C: Emerging Technologies*, 2017.
[23] V. K. Chauhan, A. Sharma, and K. Dahiya, "Faster learning by reduction of data access time," *Application Intelligence*, 2018.
[24] G. Afendras and M. Markatou, "Optimality of training/test size and resampling effectiveness in cross-validation," *Journal of Statistical Planning and Inference*, vol. 16, pp. 1-16, 2018.
[25] V. Chouvatut, W. Jindaluang, and E. Boonchieng, "Training set size reduction in large dataset problems," *Intelligent Computation Science Engineering Conference*, pp. 1-5, 2015.
[26] W. A. Yousef and S. Kundu, "Learning algorithms may perform worse with increasing training set size: Algorithmdata incompatibility," *Computational Statistics and Data Analysis*, vol. 74, pp. 181-197, 2014.
[27] G. Zhang, C. Zhang, and H. Zhang, "Improved K-Means algorithm based on density Canopy," *Knowledge-Based System*, 2018.
[28] W. L. Zhao, C. H. Deng, and C. W. Ngo, "K-Means: A revisit," *Neurocomputing*, 2018.
[29] T. Wangchamhan, S. Chiewchanwattana, and K. Sunat, "Eifficient algorithms based on the K-Means and Chaotic League Championship Algorithm for numeric, categorical, and mixed-type data clustering," *Expert Systems with Applications*, 2017.
[30] S. Maldonado, E. Carrizosa, and R. Weber, "Kernel Penalized K-Means: A feature selection method based on Kernel K-Means," *Information Science (Ny)*, 2015.
[31] H. Ismkhan, "K-Means−+: An iterative clustering algorithm based on an enhanced version of the K-Means," *Pattern Recognition*, 2018.
[32] S. S. Yu, S. W. Chu, C. M. Wang, Y. K. Chan, and T. C. Chang, "Two improved K-Means algorithms," *Applied Soft Computing*, 2018.
[33] J. Jędrzejowicz and P. Jędrzejowicz, "Distance-based online classifiers," *Expert Systems with Applications*, vol. 60, pp. 249-257, 2016.
[34] S. Kant and I. A. Ansari, "An improved K means clustering with Atkinson index to classify liver patient dataset," *International Journal of System Assurance Engineering Management*, 2016.
[35] Y. Ding, Y. Zhao, X. Shen, M. Musuvathi, and T. Mytkowicz, "Yinyang K-Means: A drop-in replacement of the classic K-Means with consistent speedup," *Icml-2015*, 2015.
[36] R. Jothi, S. K. Mohanty, and A. Ojha, "DK-Means: a deterministic K-Means clustering algorithm for gene expression analysis," *Pattern Analysis and Applications*, 2017.
[37] G. Tzortzis and A. Likas, "The MinMax k-Means clustering algorithm," *Pattern Recognition*, 2014.
[38] I. Melnykov and V. Melnykov, "K-Means algorithm with the use of mahalanobis distances," *Statistics and Probability Letters*, 2014.
[39] A. Shirkhorshidi, S. Aghabozorgi, T. Wah, and T. Herawan, "Big data clustering: a review," *International Conference on Computational Science and its Applications*, Springer, pp. 707–720, 2014.
[40] B. LIU, "A fast density-based clustering algorithm for large databases," *International Conference on Machine Learning and Cybernetics*, IEEE, pp. 996–1000, 2006.
[41] Y. Wu, J. Guo, and X. Zhang, "A linear DBSCAN algorithm based on Location Sensitive Hashing," *International Conference on Machine Learning and Cybernetics*, IEEE, vol. 5, pp. 2608–2014.
[42] Y. Dogan, D. Birant, and A. Kut, "SOM++: Integration of self-organizing map and K-Means++ algorithms," *International Conference on Machine Learning and Data Mining in Pattern Recognition*, Springer, pp. 246–259, 2013.
[43] A. Bakr, N. Ghanem, and M. Ismail, "Efficient incremental density-based algorithm for clustering large datasets," *Alex Engineering Journal*, vol. 54, no. 4, pp. 1147–1154, 2015.
[44] T. Xu, H. Chiang, G. Liu, and C. Tan, "Hierarchical K-Means method for clustering large-scale advanced metering infrastructure data," *IEEE Transaction Power Delivery*, vol. 32, no. 2, pp. 609–616, 2015.
[45] H. Ismkhan, "K-Means++: An iterative clustering algorithm based on an enhanced version of the k-means," *PattRecogn*, no. 79, pp. 402–413, 2018.
[46] D. Brown, A. Japa, and Y. Shi, "A fast density-grid based clustering method Daniel Brown," *IEEE 9th Annual Computing and Communication Workshop and Conference*, IEEE, pp. 48–54, 2019.
[47] V. Mathur, J. Mehta, and S. Singh, "HCA-DBSCAN: HyperCube accelerated density based spatial clustering for applications with noise," *33rd Conference on Neural Information Processing Systems (arXiv preprint)*, 2019.
[48] D. Luchi, A. Rodrigues, and F. Varejao, "Sampling approaches for applying DBSCAN to large datasets," *Pattern Recognition Letters*, no. 117, pp. 90–96, 2019.
[49] Y. Chen, L. Zhou, S. Pei, Z. Yu, Y. Chen, X. Liu, J. Du, and N. Xiong, "KNN-BLOCK DBSCAN Fast Clustering for Large-Scale Data," *IEEE Transactions on Systems*, Man, and Cybernetics: Systems, pp. 1–15, 2019.
[50] Y. Chen, L. Zhou, N. Bouguila, C. Wang, Y. Chen, J. Du, "BLOCK-DBSCAN Fast clustering for large scale data," *Pattern Recognition*, vol. 109, no. 107627, 2020.
[51] X. Cui, P. Zhu, X. Yang, K. Li, and C. Ji, "Optimized big data K-Means clustering using MapReduce," *J Supercompu*, vol. 70, no. 3, pp. 1249–1259, 2014.
[52] A. Sinha, and P. Jana, "A novel K-Means based clustering algorithm for big data," *Conference on Advances in Computing, Communications and Informatics*, IEEE, pp. 1875–1879, 2016.
[53] H. Song, and J. Lee, "RP-DBSCAN: A superfast parallel DBSCAN algorithm based on random partitioning," *International Conference on Management of Data*, pp. 1173–1187, 2018.
[54] S. Li, "An Improved DBSCAN Algorithm Based on the Neighbor Similarity and Fast Nearest Neighbor Query," *IEEE Access*, no. 8, pp. 47468–47476, 2020.
[55] https://archive.ics.uci.edu/ml/index.php
[56] B. Hajji, A. Mellit, G. Marco Tina, A. Rabhi, J. Launay, and S. E. Naimi, *Proceedings of the 2nd International Conference on Electronic Engineering and Renewable Energy Systems*. Lecture Notes in Electrical Engineering, 2021.
[57] N. Gholizadeh, H. Saadatfar, and N. Hanafi, "K-DBSCAN: An improved DBSCAN algorithm for big data," *The Journal of Supercomputing*, no. 77, pp. 6214-6235, 2021.

1. Radial Basis Function (RBF)
2. Atkinson
3. Mahala Nobis
4. Location Sensitive Hashing (LSH)

شماره تماس: ۰۹۱۱۲۷۸۱۷۱۲

سرگذشت: فرشته حاج‌قاضی دارای مدرک کارشناسی‌ارشد مهندسی کامپیوتر گرایش نرم‌افزار از موسسه غیرانتفاعی میرداماد گرگان در سال ۱۳۹۹ و درحال‌حاضر، دانشجوی دکتری در دانشگاه آزاد اسلامی واحد نیشابور است. زمینه‌های تحقیقاتی او محاسبات گرید و شبکه عصبی است.

نام و نام خانوادگی: یعقوب آراد

شماره تماس: ۰۹۱۵۱۸۸۸۴۹۹

سرگذشت: یعقوب آراد دارای مدرک کارشناسی‌ارشد مهندسی کامپیوتر گرایش نرم-افزار از دانشگاه آزاد اسلامی واحد شیروان در سال ۱۴۰۰ و درحال‌حاضر، دانشجوی دکتری در دانشگاه آزاد اسلامی واحد نیشابور است. زمینه‌های تحقیقاتی او شبکه‌های حسگر بی‌سیم است.

5. Self-Organizing Map (SOM)
6. K-Nearest Neighbors (KNN)

معرفی نویسندگان:

نام و نام خانوادگی: رضا قائمی

شماره تماس: ۰۹۱۵۳۱۱۵۷۷۲

سرگذشت: رضا قائمی دارای مدرک دکتری مهندسی کامپیوتر گرایش هوش مصنوعی از دانشگاه ملی یوپی‌ام مالزی در سال ۲۰۱۱ و درحال‌حاضر، استادیار گروه مهندسی کامپیوتر و کارشناسی‌ارشد هوش مصنوعی در دانشگاه آزاد اسلامی واحد قوچان است. زمینه‌های تحقیقاتی او یادگیری ماشین، داده‌کاوی، محاسبات نرم، کلان داده‌ها و اینترنت اشیاء است.

نام و نام خانوادگی: فرشته حاج قاضی استرآبادی