DOR: 20.1001.1.2538161.1403.9.3.1.4

DOI: 10.22034/csj.2024.209865

پاییز ۱۴۰۳، دوره ۹، شماره ۳، پیاپی ۳۴

دریافت مقاله: ۱۴۰۳/۰۴/۱۱

پذیرش مقاله: ۱۴۰۳/۰۶/۰۵

**نوع مقاله: پژوهشی**

# بهینه‌سازی الگوریتم k–میانگین در خوشه‌بندی داده‌ها با استفاده از الگوریتم هوش جمعی بهینه‌سازی گرگ خاکستری و الگوریتم تشخیص داده پرت جنگل جداسازی

**فائزه مرتضائی آغوزبنی**

دانشجوی کارشناسی ارشد گروه مهندسی کامپیوتر، دانشگاه بین‌المللی امام خمینی (ره)، قزوین، ایران

پست الکترونیکی: s996169003@edu.ikiu.ac.ir

**مرتضی محمدی زنجیره***

استادیار گروه مهندسی کامپیوتر، دانشگاه بین‌المللی امام خمینی (ره)، قزوین، ایران

پست الکترونیکی: Zanjireh@eng.ikiu.ac.ir

## چکیده

امروزه علم داده و خوشه‌بندی داده‌ها به‌عنوان ابزارهای حیاتی برای تحلیل و پردازش داده‌های خام و استخراج دانش به منظور تصمیم‌سازی و تصمیم‌گیری‌های کلان شناخته می‌شوند. یکی از روش‌های اصلی طبقه‌بندی داده‌ها، خوشه‌بندی است که در آن عناصر درون هر خوشه باید با یکدیگر مشابه و با عناصر خوشه‌های دیگر متفاوت باشند. الگوریتم k- میانگین به‌عنوان یکی از روش‌های پرکاربرد در خوشه‌بندی داده‌ها به‌شمار می‌آید و در بسیاری از کاربردهای عملی مورد استفاده قرار می‌گیرد. با این حال، این الگوریتم دارای دو نقص اساسی است: اول، وابستگی شدید کیفیت خوشه‌ها به انتخاب مراکز اولیه، و دوم، تأثیر نقاط پرت بر عملکرد خوشه‌بندی. در این مقاله، یک روش پیشرفته برای بهینه‌سازی الگوریتم k- میانگین با استفاده از الگوریتم بهینه‌سازی گرگ خاکستری (GWO) برای انتخاب اولیه مراکز خوشه و همچنین الگوریتم جنگل جداسازی (IF) برای حذف نقاط پرت معرفی شده است. در این مقاله آزمایش‌های مختلف بر روی مجموعه داده‌های سنتزی و داده‌های واقعی انجام شد و هر کدام از آزمایش‌ها با دو سنجه شاخص رند اصلاح شده و خطای تعداد خوشه‌ها (میانگین اختلاف بین تعداد خوشه‌های واقعی و تعداد براورد شده توسط الگوریتم) مورد ارزیابی قرار گرفتند. آزمایش‌های انجام شده نشان داد که الگوریتم پیشنهادی هم از نظر شاخص رند اصلاح شده و هم از نظر خطای تعداد خوشه‌ها پیشرفت قابل ملاحظه‌ای را در مقایسه با k–میانگین پایه نشان می‌دهد. به گونه‌ای که در شاخص رند اصلاح شده، امتیاز آن از ۰/۷٤ به ۰/۹۳ ارتقا یافت. همچنین از نظرمیانگین تعداد خطای خوشه‌ها، میانگین خطای آن از عدد ۰/٤۷ به ۰/۱٦ کاهش یافت این آزمایش‌های گسترده و متعدد بر روی داده‌های متنوع نشان داد که این الگوریتم ترکیبی توانسته است به شکل

* نویسندهٔ مسئول

قابل ملاحظــه‌ای الگوریتم k- میانگیــن اولیه را از جهات گوناگون بهبود ببخشــد و بتواند راه‌حلی نویدبخش برای کاربردهای آتی خوشه‌بندی داده‌ها باشد.

واژه‌هــای کلیــدی: خوشــه‌بندی داده‌هــا، الگوریتــم k-میانگین، الگوریتم بهینه‌سازی گرگ خاکستری، الگوریتم جنگل جداسازی

## ۱. مقدمه

خوشه‌بندی یکی از ابزارهای اساسی تحلیل داده‌ها است کــه به طبقه‌بندی داده‌ها می‌پردازد، به گونه‌ای که داده‌های هر خوشه بیشترین شــباهت را به یکدیگر داشته باشند و هم زمان از داده‌های ســایر خوشه‌ها تا حد امکان متفاوت باشــند. یکی از روش‌های خوشه‌بندی پرکاربرد، الگوریتم k-میانگین[1] اســت که به‌طور گســترده برای بســیاری از کاربردهای عملی استفاده می‌شــود. الگوریتم k-میانگین یکی از محبوب‌ترین روش‌های خوشــه‌بندی اســت که به دلیل سادگی و کارآیی بسیار مورد استفاده قرار می‌گیرد. این الگوریتم یک روش تقسیم‌بندی است که داده‌ها را به k گروه جداگانه تفکیک می‌کند. خروجی الگوریتم k-میانگین به مقادیر مرکزی انتخاب شــده برای خوشه‌بندی وابسته است. بنابراین دقت این الگوریتم به شدت به انتخاب مراکز اولیه بستگی دارد [۱، ۲، ۳]. الگوریتم استاندارد k-میانگین به حداقل محلی همگرا می‌شود نه بهینه سراسری [٤].

با وجود کاربرد گسترده، الگوریتم k-میانگین از برخی اشکالات مســتثنی نیست. این مشــکلات به طور گسترده در تحقیقات مروری گزارش شــده است. اول این که، تنها می‌تواند خوشــه‌های فشــرده و ابرکره‌ای که به خوبی از هم جدا شــده‌اند را تشخیص دهد [۲]. دوم این که، به دلیل اســتفاده از فاصله اقلیدســی مجذور، به نوفه و نقاط پرت حساس است؛ حتی تعداد کمی از این نقاط می‌توانند به‌طور قابل توجهی بر میانگین خوشه‌های مربوطه تأثیر بگذارند، این مشکل می‌تواند با هرس کردن برطرف شود. سوم این که، این الگوریتم حساسیت بالایی نسبت به انتخاب مراکز اولیه دارد. انتخاب نادرســت مراکز اولیه می‌تواند منجر به خوشه‌های خالی، همگرایی آهسته‌تر و احتمال بیشتر گیر افتادن در حداقل‌های محلی شود. خوشبختانه، این ایرادات را می‌توان با استفاده از روش انتخاب مراکز اولیه برطرف کرد [۵].

در این پژوهش، تصمیم گرفتیم که مراکز اولیه خوشه‌ها را با استفاده از الگوریتم بهینه‌سازی گرگ خاکستری۲ پیدا کرده و سپس خوشه‌بندی را اجرا کنیم زیرا الگوریتم گرگ خاکستری و البته سایرالگوریتم‌های هوش جمعی۳ به دلیل داشــتن توانایی فرار از بهینه‌های محلــی، نتایج موفقی را در خوشه‌بندی داده‌ها نشــان داده‌اند. الگوریتم‌های هوش جمعی بــا انتخاب مراکز خوب در میان حجم وســیعی از داده‌ها، خوشــه‌بندی را انجام می‌دهند و از نظر پیچیدگی زمانــی و مصــرف حافظه، شــرایط بهتری را نســبت به الگوریتم‌های دقیق فراهم می‌کنند. از قابلیت جستجوی این الگوریتم‌ها برای جســتجوی مراکز خوشه‌های مناسب در فضای ویژگی داده شده استفاده می‌گردد.

از ســوی دیگر، نقاط پرت یا ناهنجار، بخشی از ماهیت داده‌های جهان واقعی است که به علل مختلف مانند، خطاهای اندازه‌گیری، خطاهای انسانی، دستکاری اطلاعات یا ماهیت طبیعی داده‌ها، در داده‌ها ظاهر می‌شوند. وظیفه تشخیص ناهنجاری به‌عنوان تشخیص نقاط پرت در داده‌های بزرگ، کــه از جنبه‌های مختلف زندگی واقعــی، به ویژه داده‌های تجاری به دســت می‌آینــد، به دلیل افزایش ســریع اندازه مجموعه داده‌هــا و پیچیدگی‌های آن‌ها در ســازمان‌های مدرن، یکی از چالش برانگیزترین مشکلات بوده است.

توانایی تشــخیص ناهنجــاری توســط الگوریتم‌های خوشــه‌بندی یا طبقه‌بندی، معمولاً یک اثر جانبی یا نتیجه جانبی این الگوریتم‌ها اســت، زیرا آن‌ها برای هدفی غیر از تشخیص ناهنجاری مانند طبقه‌بندی یا خوشه‌بندی، طراحی شــده‌اند. این منجر به دو اشــکال عمده می‌شــود: (۱) این روش‌ها برای تشخیص ناهنجاری بهینه نیستند، در نتیجه

1-K-Means

2-Grey Wolf Optimizer (GWO)
3-Swarm Intelligence (SI)

آن‌ها اغلب ضعیف عمل می‌کنند و هشدارهای کاذب بسیار زیادی ایجاد می‌شود که موارد نرمال را به‌عنوان ناهنجاری شناسایی می‌کنند یا ناهنجاری‌های بسیار کمی شناسایی می‌شود. (۲) بسیاری از روش‌های موجود محدود به داده‌هایی با ابعاد کم و اندازه کوچک هستند.

در میان الگوریتم‌های مختلف تشخیص ناهنجاری، الگوریتم جنگل جداساز[۴]، الگوریتمی با قابلیت منحصر به فرد است. الگوریتمی رایگان که از نظر محاسباتی، کارآمد بوده و در تشخیص ناهنجاری‌ها بسیار موثر است. به همین دلیل، نقاط ناهنجار داده‌ها را با استفاده از الگوریتم جنگل جداساز، شناسایی و حذف کردیم. پس از انجام آزمایش‌های مختلف روی مجموعه داده‌های متنوع، بهبود کارایی سیستم پیشنهادی، نسبت به روش پایه الگوریتم K–میانگین اثبات گردید. در این مقاله ترکیب این سه الگوریتم با یک دیگر برای بار اول انجام شده است.

مبنای مقایسه و ارزیابی ایده پژوهش جاری، الگوریتم اصلی خوشه‌بندی k–میانگین است. این الگوریتم را به زبان پایتون پیاده‌سازی کردیم. پس از پیاده‌سازی این الگوریتم، نوبت به الگوریتم بهینه‌سازی گرگ خاکستری می‌رسد. برای این الگوریتم نیز ردهٔ مورد نیاز الگوریتم و یک تابع برازش (فیت‌نس[۵]) طراحی کردیم. برای الگوریتم جنگل جداساز، از کتابخانه استاندارد و آماده سایکیت‌لرن[۶] استفاده کردیم. در این پژوهش، هدف ما بهینه‌سازی الگوریتم k–میانگین بوده و به همین دلیل، این الگوریتم را به صورت دستی پیاده‌سازی کرده و برای الگوریتم‌های دیگر از کتابخانه‌های استاندارد پایتون استفاده شده است.

در این پژوهش، از هر دو نوع مجموعه داده سنتزی و واقعی استفاده کردیم. داده‌های سنتزی، بوسیله تابع بلاب[۷] از ماژول مجموعه داده[۸] از کتابخانه سایکیت‌لرن در چندین مقدار متفاوت، تعداد نمونه، تعداد خوشه و تعداد ویژگی تولید شدند. از دو مجموعه داده واقعی نیز استفاده کردیم:

۱) داده‌های قطعه‌بندی تصویر که مربوط به تصاویری از ۷ خوشه متفاوت است. ۲) داده‌های مربوط به اشیای واقع در زیر آب که در یکی از خوشه‌های سنگ یا مین زیردریایی قرار می‌گیرند. مجموعه داده‌های واقعی، از داده‌های موجود در اینترنت، برای خوشه‌بندی، در سامانه گیت‌هاب تهیه شده است. برای هر مجموعه داده، آزمایش‌های زیر اجرا گردید:

- اجرای الگوریتم خوشه‌بندی k–میانگین روی مجموعه داده (مبنای مقایسه و ارزیابی ایده‌ها)
- اجرای الگوریتم بهینه‌سازی گرگ خاکستری و سپس خوشه‌بندی k–میانگین (بدون هیچ اقدامی روی نقاط پرت)
- اجرای الگوریتم بهینه‌سازی گرگ خاکستری برای یافتن مراکز بهینه و سپس خوشه‌بندی با الگوریتم k–میانگین و در آخر حذف نقاط پرت هر خوشه
- حذف نقاط پرت کل مجموعه داده (در ابتدای کار) با الگوریتم جنگل جداساز، سپس اجرای الگوریتم بهینه‌سازی گرگ خاکستری برای یافتن مراکز خوشه بهینه و در آخر اجرای الگوریتم خوشه‌بندی k–میانگین
- حذف نقاط پرت کل مجموعه داده (در ابتدای کار) با الگوریتم جنگل جداساز، اجرای الگوریتم گرگ خاکستری و سپس خوشه‌بندی با الگوریتم k–میانگین و در آخر انتساب هر نقطه پرت به نزدیک‌ترین خوشه

معیارهای ارزیابی متعددی برای سنجش کارایی خوشه‌بندی وجود دارد. معیارهای ارزیابی خوشه‌بندی، در ماژول متریک[۹]، از کتابخانه سایکیت‌لرن موجود است. با مراجعه به مستندات این کتابخانه در سایت سایکیت‌لرن[۱۰]، معیار شاخص رند اصلاح شده یا تعدیل شده[۱۱] را برای ارزیابی نتایج هر دسته از آزمایش‌ها و نیز به‌عنوان تابع شایستگی الگوریتم گرگ خاکستری استفاده کردیم. همچنین در مجموعه داده‌ی شامل ۱۲۰۰ مشاهده، میانگین خطای تعداد خوشه‌ها معیاری دیگر بود که برای ارزیابی کیفیت خوشه‌ها مورد استفاده قرار گرفت. آزمایش‌های

4-Isolation Forest
5-Fitness
6-Sklearn
7-Blob
8-Datasets
9-Metric
10-https://scikit-learn.org/stable/modules/clustering.html
11-Adjusted-Rand-Score

انجام شـــده نشـــان داد که امتیاز الگوریتم پیشـــنهادی در شـــاخص رند اصلاح شـــده از ۰/۷٤ بـــه ۰/۹۳ ارتقا یافت. همچنین میانگین تعداد خطای خوشه‌ها در آن از عدد ۰/٤۷ به ۰/۱٦ کاهش یافت.

ساختار بقیه مقاله به این صورت است: بخش ۲ شامل معرفـــی، بیان تعاریف و مفاهیم و همچنین کارهای مرتبط گذشـــته و بخش ۳ حـــاوی توضیحات مربـــوط به روش پیشـــنهادی اســـت. بخش ٤ مربوط به ارزیابـــی الگوریتم پیشنهادی و در نهایت نتیجه‌گیری در بخش ٥ آمده است.

## ۲.تعاریف و مفاهیم اولیه

در این بخـــش تعاریف و مفاهیم اولیـــه و نیز کارهای مرتبط با این تحقیق ارایه شده است.

### ۲-۱ تبیین تعاریف و مفاهیم

تعاریف و مفاهیم اولیه به کار رفته در تحقیق بصورت زیر است:

**۲-۱-۱ الگوریتم k-میانگین**

در الگوریتم استاندارد k-میانگین، ابتدا k نقطه به‌عنوان مراکز اولیه خوشه‌ها به صورت تصادفی انتخاب می‌شوند. ســـپس، فاصله اقلیدســـی ســـایر نقاط مجموعـــه داده از هریک از این مراکز اولیه محاســـبه می‌شود و هر نقطه به نزدیکترین مرکز اختصاص می‌یابد. پس از تخصیص همه نقاط، مراکز خوشـــه‌ها به روزرسانی می‌شوند به گونه‌ای که میانگین نقاط هر خوشـــه، به‌عنوان مرکز جدید خوشه در نظر گرفته می‌شـــود. این فرآیند تا زمانی که تخصیص نقاط به خوشه‌ها در دو مرحله متوالی تغییری نکند، تکرار می‌شـــود و در نهایت الگوریتم متوقف و خوشـــه‌ها نهایی می‌شوند [۷ و ٦].

الگوریتم k-میانگین از نظر محاسباتی سنگین است و به زمانی متناسب با حاصل ضرب تعداد فقره‌های داده، تعداد خوشـــه‌ها و تعداد تکرار نیاز دارد. اگر n تعداد نمونه‌ها، k تعداد خوشـــه‌ها و t تعداد تکرار الگوریتم باشد، پیچیدگی زمانی و فضایـــی آن به ترتیب برابـــر O(nkt) و O(n+ k) خواهند بود [۸، ۹ و ۱۰].

الگوریتـــم k-میانگین یکی از الگوریتم‌هـــای پرکاربرد برای خوشه‌بندی داده‌ها اســـت. که استفاده از آن به دلیل بهترین عملکرد برای مجموعه داده‌های بزرگ، بسیار رایج است [۱۱]. با این که این الگوریتم یک روش یادگیری بدون نظارت برای خوشـــه‌بندی در تشـــخیص الگو و یادگیری ماشـــین اســـت، اما مقداردهی اولیـــه و آگاهـــی از تعداد موردنیاز خوشه‌ها، همواره بر عملکرد آن تاثیر می‌گذارد. بنابراین، الگوریتم k-میانگین دقیقاً یک روش خوشـــه‌بندی بدون نظارت نیست [۱۲].

**۲-۱-۲-الگوریتم‌های هوش جمعی**

الگوریتم‌هـــای هـــوش جمعی بخشـــی از حوزه هوش محاسباتی[12] هستند که برای حل مسائل بهینه‌سازی به کار می‌روند. این الگوریتم‌ها از رفتار ازدحام در طبیعت، مانند ازدحام ماهی‌ها، پرندگان، حشرات و گله‌های حیوانات تقلید می‌کنند. بسیاری از راه‌حل‌های هوش جمعی از رفتارهای جمعی اعضـــای جوامع مختلف الهام گرفته‌اند که شـــامل روش زندگی، الگوهای ارتباطی و سلسله مراتب اجتماعی آنها می‌شـــود. از جمله الگوریتم‌های مختلف هوش جمعی می‌توان به الگوریتم ژنتیک[13]، بهینه‌سازی کلونی مورچه‌ها[14]، کلونی زنبورهای مصنوعی[15]، بهینه‌سازی ازدحام ذرات[16]، جستجوی فاخته[17]، الگوریتم خفاش[18]، بهینه‌سازی نهنگ[19]، الگوریتم کرم شب تاب[20]، الگوریتم جستجوی گرانشی[21] و الگوریتم بهینه‌سازی گرگ خاکستری اشاره کرد [۱۳].

هوش جمعی را می‌توان به‌عنوان مجموعه‌ای ســـازمان یافته از عامل‌ها تعریف کرد که با یکدیگر همکاری می‌کنند. در کاربردهای محاسباتی هوش جمعی، از عامل‌هایی مانند مورچه‌ها، زنبورهـــا، موریانه‌ها، ماهی‌ها، پرندگان یا حتی الگوی جریان آب الگوبرداری می‌شـــود. هر یک از عامل‌ها

12-Computational Intelligence
13-Genetic Algorithm (GA)
14-Ant Colony Optimization (ACO)
15-Artificial Bee Colony (ABC)
16-Particle Swarm Optimization (PSO)
17-Cuckoo Search (CS)
18-Bat algorithm (BA)
19-Whale Optimization Algorithm (WOA)
20-Firefly Algorithm (FA)
21-Gravitational Search Algorithm (GSA)

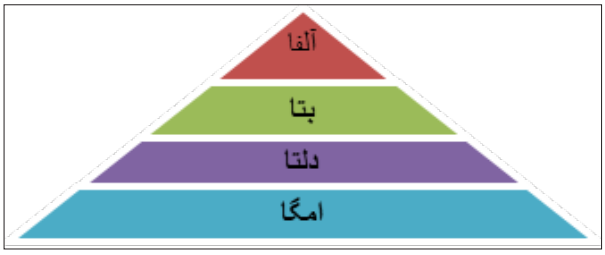
دارای ساختاری ساده هستند، اما رفتار گروهی آن‌ها پیچیده است. برای مثال، در کولونی مورچه‌ها، هر مورچه وظیفه خاص و ساده‌ای را انجام می‌دهد، اما به‌طور گروهی رفتار آنها شامل، ساختن لانه، نگهبانی از ملکه و نوزادان، پاکسازی لانه، یافتن بهترین منابع خوراکی و بهینه‌سازی راهبرد جنگی است[۱۳].

رفتار کلی یک هوش جمعی به صورت غیرخطی از هم‌آمیختگی رفتارهای تک‌تک اعضای اجتماع حاصل می‌شود. به عبارت دیگر، رابطه پیچیده‌ای میان رفتار گروهی و رفتار فردی اجتماع وجود دارد. رفتار گروهی تنها وابسته به رفتار فردی اعضا نیست، بلکه به نحوه تعامل میان آنها نیز بستگی دارد. تعامل میان اعضا، تجربه‌های آنها را درباره محیط پیرامون افزایش می‌دهد و باعث پیشرفت می‌شود. ساختار اجتماعی ازدحامی میان افراد مجموعه‌ای از کانال‌های ارتباطی را ایجاد می‌کند که از طریق آنها افراد می‌توانند تجربه‌های شخصی خود را مبادله کنند. مدل‌سازی محاسباتی هوش جمعی، کاربردهای امیدبخشی را به ویژه در زمینه بهینه‌سازی داشته است [۱٤].

**۱-۲-۱-۲- الگوریتم بهینه‌سازی گرگ خاکستری**

الگوریتم بهینه‌سازی گرگ خاکستری در مقایسه با دیگر الگوریتم‌های هوش جمعی برتری‌هایی دارد. این الگوریتم عملکرد بسیار کارآمدی در فضای جستجوی ناشناخته دارد و توانایی بالایی در اجتناب از بهینه محلی هنگام آزمایش بر روی توابع ترکیبی نشان می‌دهد به همین دلیل، برای حل مسائل دنیای واقعی بسیار مناسب است و تقریباً در تمامی حوزه‌های بهینه‌سازی توابع، پیش‌بینی، پردازش تصویر، مسائل مهندسی، مشکلات خوشه‌بندی و غیره قابل اجرا است [۱۳].

الگوریتم بهینه‌سازی گرگ خاکستری یکی از الگوریتم‌های محبوب هوش جمعی است که با انتخاب مراکز مناسب در میان حجم وسیعی از داده‌ها، فرآیند خوشه‌بندی را انجام می‌دهد. این الگوریتم از نظر پیچیدگی زمانی و




شکل ۱: سلسله مراتب گرگ‌های آلفا، بتا، دلتا و امگا [۱۹]

مصرف حافظه شرایط بهتری نسبت به الگوریتم‌های دقیق فراهم می‌کند. قابلیت جستجوی این الگوریتم برای یافتن مراکز خوشه‌های مناسب در فضای ویژگی داده شده استفاده می‌شود، که این امر منجر به بهبود کارآیی و دقت در خوشه‌بندی داده‌ها می‌گردد [۱۵].

در الگوریتم بهینه‌سازی گرگ خاکستری، گرگ‌ها به چهار طبقه اجتماعی آلفا، بتا، دلتا و امگا تقسیم می‌شوند[۱٦]. در تکرارهای مختلف الگوریتم، سه راه‌حل برتر هر دوره به ترتیب به‌عنوان گروه آلفا، بتا و دلتا نامیده می‌شوند. فرآیند شکار بهترین راه‌حل، توسط این سه طبقه گرگ برتر هدایت می‌شود. گرگ‌های امگا برای رسیدن به بهترین راه‌حل‌ها به دور این سه طبقه حلقه می‌زنند و موقعیت خود را تغییر می‌دهند. در واقع، مرحله شکار توسط گرگ‌های آلفا، بتا و دلتا انجام می‌شود، در حالی که گرگ‌های امگا، از آن‌ها پیروی می‌کنند [۱۸ و ۱۷].

شکل ۱، سلسله مراتب گرگ‌های آلفا، بتا، دلتا و امگا را نشان می‌دهد:

با توجه به شکل، کارآمدترین گرگ‌ها از دسته گرگ‌های آلفا هستند که در میان تمامی طبقات گرگ‌ها، نقش رئیس را دارند. گرگ‌های بتا به گرگ‌های آلفا در ساخت استراتژی، حفظ نظم و ترتیب در بین سایر دسته‌ها کمک می‌کنند. در گله گرگ‌ها، گرگ‌های امگا ضعیف‌ترین اعضا هستند وجزئی از گرگ‌های اصلی محسوب نمی‌شوند. این گرگ‌ها از دستورات گرگ‌های آلفا، بتا و دلتا پیروی می‌کنند و در هر مرحله توسط سه دسته دیگر هدایت می‌شوند تا به هدف حمله کنند.

آخرین نوع گرگ‌ها، دلتا نام دارند و به چهار نوع

تقسیم می‌شـــوند. پیشاهنگان۲۲: مســـئولیت در مرزها را بر عهـــده دارند. نگهبان۲۳: از رفـــاه گرگ‌های داخل گروه اطمینـــان حاصل می‌کننـــد. بزرگان۲٤: داناتریـــن افراد در میان گرگ‌های آلفا یا بتای جوان هستند. سرایداران۲۵: از گرگ‌های بدحال مراقبت می‌کنند [۲۰].

**۲-۲-۱-۲ الگوریتم تشخیص داده پرت جنگل جداسازی**

در میـــان الگوریتم‌های مختلف تشـــخیص داده پرت، الگوریتـــم جنگل جداســـازی دارای قابلیت‌های منحصر به فرد اســـت. این الگوریتم از نظر محاسباتی کارآمد بوده و در تشخیص داده‌های پرت بسیار موثر است. این الگوریتم از درختان دودویی برای تشخیص داده‌های پرت استفاده می‌کند که منجر به پیچیدگی زمانی خطی و اســـتفاده کم از حافظه می‌شود. این ویژگی‌ها به الگوریتم اجازه می‌دهد تا اندازه داده‌های بسیار بزرگ و مسائل با ابعاد بالا و تعداد زیادی ویژگی نامربوط را مدیریت کند [۲۱].

الگوریتم جنگل جداسازی از زمان معرفی به‌عنوان یک روش سریع و قابل اعتماد برای تشخیص داده‌های پرت در زمینه‌های مختلف مانند امنیت سایبری، امور مالی، تحقیقات پزشکی و غیره محبوبیت پیدا کرده است و الگوریتم جنگل جداسازی، مشابه الگوریتم جنگل‌های تصادفی[۲۶]، بر اساس درخت‌های تصمیم[۲۷] ســـاخته می‌شود. به دلیل عدم وجود برچسب‌های از پیش تعریف شده، این الگوریتم مدل بدون نظارت اســـت. الگوریتم جنگل جداســـازی بر این واقعیت استوار اســـت که داده‌های پرت نقاطی هســـتند که نادر و متفاوت هستند [۱۰].

بیشتر الگوریتم‌های تشخیص داده‌های پرت، داده‌های پرت را با درک توزیع خواص و جداســـازی آن‌ها از سایر نمونـــه داده‌های عادی پیـــدا می‌کننـــد. در الگوریتم جنگل جداسازی، داده‌ها نمونه‌برداری شده و در ساختار درختی بر اســـاس برش‌های تصادفـــی در مقادیر انتخاب شـــده ویژگی‌های موجود در مجموعه داده، پردازش می‌شـــوند. نمونه‌هایی که به عمق بیشتری در شاخه‌های درخت سفر می‌کنند، احتمـــال پرت بودن آن‌ها کمتر اســـت؛ به عبارت دیگر، شـــاخه‌های کوتاه‌تر نشـــان‌دهنده داده پرت هستند. به این ترتیـــب، مجموع طول شـــاخه‌های درخت به‌عنوان مقیاسی برای اندازه‌گیری داده‌های پرت یا امتیاز داده‌های پرت برای هر نقطه داده شده فراهم می‌شود [۱۰ و ۲۲].

در جنگل جداســـازی تشـــخیص نقاط پـــرت مبتنی بر مجموعه‌ای از درخت‌های تصمیم دودویی است. هر درخت در یک جنگل جداســـازی به‌عنوان درخت جداساز شناخته می‌شـــود. این الگوریتم با آموزش داده‌هـــا از طریق تولید درختان جداسازی شروع می‌شود. برای بررسی مرحله به مرحله این الگوریتم ابتدا یک مجموعه داده دریافت می‌شود و یک زیرنمونه تصادفی از داده‌ها انتخاب می‌شود و به یک درخت دودویی اختصاص می‌یابد.

انشـــعاب درخت با انتخـــاب تصادفی یـــک ویژگی از مجموعه n ویژگی موجود آغاز می‌شـــود. سپس انشعاب بر روی یک آســـتانه تصادفی، که هر مقـــدار در محدوده حداقل و حداکثر مقادیر ویژگی انتخاب شـــده است، انجام می‌شـــود. اگر مقدار یک نقطه داده کمتر از آستانه انتخاب شده باشد، به شـــاخه چپ و در غیر این‌صورت به شاخه راســـت می رود. به این ترتیب، یک گره به شاخه‌های چپ و راست تقســـیم می‌شود. این فرآیند به صورت بازگشتی ادامه می‌یابد تا هر نقطه داده کاملاً جدا شـــود یا تا رسیدن به حداکثر عمق (در صورت تعریف) ادامه پیدا کند [۲۳].

مراحل فوق برای ساخت درخت‌های دودویی تصادفی تکرار می‌شـــوند. پس از ایجاد مجموعـــه‌ای از درخت‌های جداســـاز (جنگل جداسازی)، آموزش مدل تکمیل می‌شود. در مرحله امتیازدهی، یک نقطه داده از میان تمام درختانی که قبلا آموزش داده شده بودند، عبور داده می‌شود. اکنون، بر اساس عمق درخت مورد نیاز برای رسیدن به آن نقطه، امتیـــاز داده پرت، به هر یک از نقـــاط داده اختصاص داده می‌شـــود. این امتیاز از مجموع عمق به دســـت آمده از هر

22-Scouts
23-Sentinels
24-Elders
25-Caretakers
26-Random Forest
27-Decision Tree

یک از درختان جداساز محاسبه می‌شود. بر اساس امتیاز داده‌های پرت (درصد داده‌های پرت موجود در کل داده‌ها)، به داده‌های پرت امتیاز ۱- و به نقاط نرمال ۱ اختصاص داده می‌شود [۲۳].

## ۳- کارهای مرتبط

در سال‌های اخیر، با توسعه فناوری و افزایش حجم داده‌ها، کاوش داده‌های حجیم بیش از پیش مورد توجه محققان قرار گرفته است. به موازات این پیشرفت، مسائل و مشکلات مرتبط با تشخیص داده‌های پرت در این داده‌های حجیم نیز به وجود آمده است. بنابراین، تحقیقات فراوانی برای حل این مشکل انجام شده است. روش‌های موجود برای تشخیص داده‌های پرت اغلب زمان محاسباتی بالایی دارند؛ بنابراین الگوریتم‌های بهبود یافته‌ای ارایه شده‌اند که کارایی بالاتری در آشکارسازی داده‌های پرت دارند.

در مقایسه با الگوریتم k-میانگین، الگوریتم مینی‌بَچ k-میانگین [۲۸] به جای کار بر روی یک نقطه داده تکی، براساس یک مجموعه نمونه کوچک عمل می‌کند. در این الگوریتم، هر عملیات به جای انجام روی یک نقطه تکی، روی یک دسته کوچک از داده‌ها صورت می‌گیرد. برای هر دسته کوچک داده، به‌روزرسانی مرکز خوشه‌بندی با محاسبه مقدار میانگین انجام می‌شود و دسته کوچک داده به مرکز خوشه جدید اختصاص داده می‌شود. با افزایش تعداد تکرارها، مرکز خوشه به تدریج پایدار می‌شود تا جایی که دیگر تغییر نکند و سپس محاسبه متوقف می‌شود. اگرچه مینی‌بَچ k-میانگین برای محاسبه مجموعه داده‌های حجیم مناسب است و زمان محاسبات را به شدت کاهش می‌دهد و سرعت همگرایی در تعیین خوشه‌ها را شتاب می‌بخشد، اما از آن‌جایی که این الگوریتم در فرآیند محاسبه فقط یک دسته کوچک از داده‌ها را در نظر می‌گیرد، دقت آن کاهش می‌یابد. با این حال، مینی‌بَچ k-میانگین به طور کلی یک الگوریتم بهینه‌شده از k-میانگین است ک بسیاری از مزایای این الگوریتم را حفظ می‌کند [۲۴].

الگوریتم دیگری با نام اِس‌اِم k-میانگین [۲۹]، که ترکیبی از مینی‌بَچ k-میانگین و الگوریتم ذوب شبیه‌سازی شده [۳۰] است، برای تشخیص داده‌های پرت در داده‌های حجیم مورد استفاده قرار می‌گیرد. این الگوریتم قابلیت مشخص کردن تعداد خوشه‌ها، کاهش تعداد تکرارهای الگوریتم k-میانگین و بهبود دقت خوشه‌بندی را دارد و از گیرافتادن الگوریتم در یک راه‌حل بهینه محلی جلوگیری می‌کند. [۲۴].

به منظور بهبود دقت و پایداری الگوریتم k-میانگین و حل مشکل تعیین مناسب‌ترین تعداد خوشه‌ها و بهترین مقداردهی اولیه مراکز خوشه‌ها، یک الگوریتم بهبود یافته مبتنی بر چگالی کانوپی [۳۱] ارائه شده است. مراحل الگوریتم به این صورت است که ابتدا چگالی مجموعه داده‌های نمونه، میانگین فاصله نمونه در خوشه‌ها و فاصله بین خوشه‌ها محاسبه می‌شود. سپس، نقطه‌ای با حداکثر چگالی به‌عنوان مرکز خوشه اول انتخاب می‌شود و خوشه چگال‌تر از مجموعه داده باقی‌مانده حذف می‌شود. چگالی کانوپی، به‌عنوان پیش پردازش در الگوریتم k-میانگین استفاده می‌شود و نتیجه آن به‌عنوان تعداد خوشه‌ها و مراکز خوشه اولیه برای الگوریتم k-میانگین در نظر گرفته می‌شود. نتایج شبیه‌سازی نشان می‌دهد که این الگوریتم، نتایج خوشه‌بندی بهتری را به دست می‌آورد و در مقایسه با الگوریتم k-میانگین نسبت به داده‌های پرت حساسیت کمتری دارد [۲۵].

الگوریتم k-میانگین استاندارد برای آشکارسازی نفوذ استفاده می‌شود، ولی عیب آن این است که باید از قبل تعداد خوشه‌ها مشخص باشد. این روش برای مجموعه داده‌های کوچک مشکلی ندارد و آسان و موثر است. اما وقتی مجموعه داده‌ها بزرگ باشند، این روش خسته‌کننده می‌شود. از این‌رو از الگوریتم k-میانگین ژنتیک بهبود یافته [۳۲] در سیستم تشخیص نفوذ برای آشکارسازی انواع نفوذها استفاده می‌شود. در این الگوریتم، تعداد خوشه‌ها

28-Mini Batch K-Means

29-SMK-Means
30-Simulated Annealing Algorithm
31-Canopy
32-Improved Genetic K-Means

از قبل مشخص نمی‌گردد و مقدار بهینه آن توسط الگوریتم ژنتیک تعیین می‌شـــود. تابع ارزیابی در این الگوریتم تعداد خوشه‌ها را تا حد امکان کم نگه داشته و نتیجه خوشه‌بندی را موثرتر و میزان جداسازی خوشه‌ها را بهینه‌تر می‌کند. هدف این اســـت کـــه وقتی خوشـــه‌ها هم‌پوشـــانی دارند، الگوریتم k–میانگین به‌طور قابل توجهی با اســـتفاده از دو رویکرد اصلی مقداردهی اولیـــه بهتر و تکرار چندین باره الگوریتم با کمک مقداردهی اولیه متفاوت، بهبود یابد [۲۶].

الگوریتم k–میانگین در برابر داده‌های پرت و داده‌های نوفه‌ای آسیب‌پذیر است و همچنین به مراکز اولیه خوشه‌بندی حساسیت دارد. الگوریتم k–میانگین سه سطحی[۳۳] می‌تواند بر این مشکلات غلبه کند. از آنجا که داده‌ها در یک مجموعه داده اغلب تغییر می‌کنند، مراکز خوشه به دست آمده بعد از گذشـــت زمان نمی‌توانند به صورت دقیق داده‌های موجود در هر خوشه را توصیف کنند. بنابراین، مراکز خوشه باید به‌روز شـــوند. داده‌های نوفه‌ای، داده‌های پرت و داده‌های با مقادیر کاملا متفاوت در یک خوشـــه ممکن اســـت باعث کاهش عملکرد سیســـتم‌های تطابق الگو شـــوند. الگوریتم k–میانگین دو سطحی[۳۴] می‌تواند با این مشکلات مقابله کند [۲۷].

الگوریتم خوشه‌بندی k–میانگین اغلب برای خوشه‌بندی داده‌های ثابت استفاده می‌شـــود. به طور کلی، داده‌های با اختلاف زیاد باید به خوشـــه‌های متفاوت اختصاص یابند. علاوه بر این، یک خوشـــه حاوی تعداد زیادی داده، معمولا باید به چندین خوشـــه کوچک‌تر تقســـیم شـــود. الگوریتم k–میانگین سه سطحی، برای این عملیات مناسب است. این روش همچنین به داده‌های نوفه‌ای و داده‌های پرت حساس نیســـت، بنابراین می‌تواند داده‌ها را موثرتر از k–میانگین استاندارد خوشه‌بندی کند [۲۷].

الگوریتـــم k–میانگین ســـه ســـطحی نمی‌تواند به طور کارآمـــد با داده‌هـــای درحال تکامل یـــا داده‌هایی که مدام به‌روز می‌شـــوند، مقابله کند، بنابرایـــن الگوریتم یادگیری ماشین بر خط[۳۵] مبتنی بر k–میانگین سه سطحی برای رفع این مشـــکل ارائه شد. این الگوریتم باعث کاهش زمان اجرا در تکرارهای الگوریتم می‌شود [۲۷].

الگوریتم k–میانگین با وجود ســـادگی و کارآمدی، در انتخاب مراکز خوشـــه‌بندی اولیه و حساســـیت به نوفه با مشکلاتی مواجه اســـت که باعث کاهش دقت خوشه‌بندی می‌شود. برای رفع این مشـــکلات، بهبودی برای الگوریتم k–میانگیـــن ارائه داده شـــد کـــه از مفهوم شـــبکه کلیک[۳۶] استفاده می‌کند. در این بهبود، داده‌های نوفه‌ای حذف شده و چگالی نواحی مشـــخص می‌شود. ســـپس نقاط مرکزی اولیه با روش خوشـــه‌بندی مبتنی بر جســـتجوی سریع و یافتن قله‌های چگالی به دست می‌آیند. همچنین، اثر خطای چگالی شـــبکه بر انتخاب نقاط مرکزی اولیه با استفاده از ایده دانه‌بنـــدی[۳۷]، کاهش می‌یابد. در مقایســـه با الگوریتم k–میانگین استاندارد، این الگوریتم بهبود یافته، دقت بالاتر، تفاوت کمتر در خوشه‌بندی داده‌های گوناگون و وابستگی کمتر به پارامترها را نشان می‌دهد [۲۸].

الگوریتـــم مک کوئین[۳۸]، در اصل یک نســـخه برخط از الگوریتم k–میانگین اســـت. ایراد ایـــن روش در پیچیدگی محاسباتی است، زیرا چندین تکرار از الگوریتم k–میانگین پـــس از تخصیص هر نمونه مورد نیاز اســـت که برای یک پایگاه داده بزرگ، بسیار سنگین است [۲۹].

در [۳۰]، الگوریتم اِف‌ســـی k–میانگین [۳۹] معرفی شـــده اســـت که با تثبیت برخی مراکز خوشه‌ای با در نظر گرفتن شرایط واقعی، خوشه‌بندی را امکان‌پذیر می‌سازد. در حالی که برخی از مراکز خوشـــه‌ای ثابت هستند، تلاش می‌شود مناسب‌ترین مراکز خوشه برای بقیه داده‌ها و بهترین توزیع داده‌ها در خوشه‌ها به دست آید. الگوریتم k–میانگین با دو الگوریتم خوشـــه‌بندی ثابت– مرکز دیگر مقایسه می‌شود: اِف‌ســـی k–میانگین و اِف‌ســـی k–میانگین۲. نتایج تجربی نشان می‌دهد که اگرچه اِف‌سی k–میانگین محدودیت‌های

33-Tri– Level K-Means Algorithm
34-Bi- Level K-Means Algorithm

35-Online Machine Learning
36-CLIQUE Grid
37-Granularity
38-Mac Queen Algorithm
39-FCK-Means

بیشتری نسبت به k–میانگین دارد، اما عملکرد بهتری ارایه می‌دهد.

الگوریتم k–میانگین توانایی بالایی برای مقابله با داده‌هایی با مرزهای نامشخص دارد. با این حال، دارای محدودیت‌هایی مانند حساسیت به انتخاب داده‌های اولیه و استفاده از وزن‌ها و آستانه‌های ثابت است که منجر به نتایج خوشه‌بندی ناپایدار و کاهش دقت می‌شود. برای حل این مشکل، ترکیب الگوریتم k–میانگین با الگوریتم کرم شب‌تاب[40] معرفی شده است که از سه جنبه بهبود یافته است. ابتدا، روشی معقول‌تر برای تنظیم پویا وزن‌های تقریب و مجموعه مرزی براساس نسبت تعداد اشیا در مجموعه داده به حاصل‌ضرب تفاوت اشیا در مجموعه داده طراحی شده است. در مرحله دوم، روشی برای تحقق تطبیقی آستانه مرتبط با تعداد تکرارها ارائه شده است. سپس با ساختن یک تابع هدف جدید و استفاده از مقدار آن به‌عنوان شدت روشنایی کرم شب‌تاب برای انجام جستجو و تکرار به روزرسانی نقطه مرکزی خوشه اولیه، راه‌حل بهینه به‌دست آمده توسط هر تکرار کرم شب‌تاب به‌عنوان موقعیت مرکزی اولیه در نظر گرفته می‌شود. نتایج الگوریتم نشان می‌دهد که این روش جدید، اثر خوشه‌بندی را بهبود بخشیده است [۳۱].

یک الگوریتم جدید به نام انتخاب دانه فعال[41] معرفی شده است که با انواع الگوریتم‌های خوشه‌بندی نیمه نظارت شده کارآیی دارد. این الگوریتم به ساختار گراف k–نزدیک‌ترین همسایه[42] متکی است و از گراف k–نزدیک‌ترین همسایه برای تولید اولین خوشه‌بندی مجموعه داده‌ها براساس چگالی استفاده می‌کند و امکان انتشار موثر برچسب‌های نمونه‌ها را فراهم می‌آورد. روش فوق از این واقعیت بهره می‌برد که احتمال انتخاب یک نقطه نزدیک به مرکز خوشه، بیشتر است، زیرا چگالی نقاط در اینجا بالاتر است. آزمایش‌ها روی مجموعه داده‌های واقعی نشان می‌دهد که این روش در مقایسه با روش‌های دیگر، کارآیی بالاتری دارد، خصوصا هنگامی که با خوشه‌بندی مبتنی بر نماینده[43] و مبتنی بر چگالی[44] ترکیب می‌شود. یکی از مشکلات این روش این است که اگر یک نقطه پرت به‌عنوان مرکز خوشه اولیه انتخاب شود، ممکن است هیچ نقطه دیگری به آن اختصاص نیابد و خوشه‌ای که مرکز آن پرت است، خالی بماند. همچنین، سازوکاری برای اجتناب از انتخاب نقاط داده بسیار نزدیک به یکدیگر وجود ندارد [۳۲].

در روش خوشه‌بندی ساده، اولین دانه با اولین مقدار در پایگاه داده مقداردهی اولیه می‌شود. سپس فاصله بین دانه انتخاب شده و نقطه بعدی در پایگاه داده محاسبه می‌شود. اگر این فاصله بیشتر از مقدار آستانه باشد، این نقطه به‌عنوان دانه دوم انتخاب می‌شود. در غیر این صورت به نمونه بعدی در پایگاه داده انتقال می‌یابد و روند تکرار می‌شود. پس از انتخاب دانه دوم، فاصله بین این نمونه و دو دانه از قبل انتخاب شده، محاسبه می‌شود؛ اگر هر دو فاصله از آستانه بیش‌تر باشند، نمونه به‌عنوان دانه سوم انتخاب می‌شود. این فرآیند تا زمانی که دانه‌های k انتخاب شوند تکرار می‌شود. مزیت این روش این است که به کاربر امکان کنترل فاصله بین مراکز مختلف خوشه را می‌دهد. اما این روش از محدودیت‌هایی مانند وابستگی به ترتیب نقاط در پایگاه داده و نیاز به تصمیم‌گیری کاربر درباره مقدار آستانه رنج می‌برد [۵].

در روش خوشه‌بندی کی‌کی‌زد[45]، ابتدا یک نقطه x به‌عنوان اولین دانه انتخاب می‌شود، که ترجیحا در لبه داده‌ها قرار دارد. سپس دورترین نقطه از x پیدا شده و به‌عنوان دانه دوم انتخاب می‌شود. در ادامه، فاصله تمام نقاط مجموعه داده تا نزدیک‌ترین دانه اول و دوم محاسبه می‌شود و دانه سوم نقطه‌ای است که از نزدیک‌ترین دانه خود دورتر است. این فرآیند تا زمانی که دانه‌های k انتخاب شوند، تکرار می‌شود. این روش در عمل جذاب است زیرا تصمیم‌گیری برای مراکز اولیه منحصر به فرد و ساده است. با این حال، گاهی اوقات خوشه‌های نامطلوبی را

40-Firefly
41-A New Active Seed Selection Method (S-kNNG)
42-K-Nearest Neighbors
43-Representative-Based Clustering
44-Density-Based Clustering
45-KKZ

پیدا می‌کند، زیرا به نقاط پرت وابسته است. هرگونه نوفه در داده‌ها به شکل نقاط داده دور از دسترس، برای این روش مشکل ایجاد می‌کند. چنین نقاط دورافتاده‌ای توسط الگوریتم ترجیح داده می‌شوند اما لزوماً نزدیک به مرکز خوشه نخواهد بود [۲۹].

در روش چیترا، به جای انتخاب تصادفی مراکز اولیه، از یک روش رویه‌ای برای یافتن مراکز استفاده می‌شود تا خوشه‌بندی برای هر اجرا یکسان باقی مانده و تعداد تکرارها کاهش یابد. این الگوریتم نیاز به تکرار زیادی ندارد، زیرا مراکز در مرحله اول به خوبی محاسبه می‌شوند و گروه‌بندی نقاط داده با هر اجرا تغییر نمی‌کند. پس از پایان خوشه‌بندی داده‌ها، اگر نمونه جدید وارد شود، به جای انجام مجدد کل خوشه‌بندی، نمونه جدید می‌تواند با مقادیر مشخصه‌اش، در خوشه مناسب قرار گیرد [۳۳].

در روش دورترین اول[۴۶]، اولین مرکز به صورت تصادفی از مجموعه داده انتخاب می‌شود. سپس، نقطه‌ای که از مرکز اول دورتر است انتخاب می‌شود. مرکز سوم، نقطه‌ای است که از هر دو مرکز قبلی، دورتر است. انتخاب نقاط به این ترتیب ادامه می‌یابد، تا زمانی که k مرکز وجود داشته باشد. ایراد این روش این است که تمایل به انتخاب نقاط پرت دارد که نامزد مناسبی برای مراکز خوشه‌ای نیستند. هنگامی که دانه اول به طور یکنواخت و تصادفی از مجموعه داده انتخاب می‌شود، دانه بعدی نقطه داده‌ای است که از دانه اول دورتر است و می‌تواند یک نقطه پرت باشد. همچنین، هنگام انتخاب مراکز باقیمانده، همچنان شانس زیادی وجود دارد که نقاط پرت انتخاب شوند، زیرا این روش به دنبال نقاط داده‌ای است که دورتر از مراکز انتخاب شده قبلی هستند. خوبی این روش این است که مراکز را در سراسر مجموعه داده پخش می‌کند، زیرا نقاط داده‌ای را انتخاب می‌کند که دورتر از مراکز انتخاب شده قبلی هستند. پخش کردن مراکز به طورکلی خوب است، زیرا این احتمال را کاهش می‌دهد که نقاط داده‌ای که خیلی از یکدیگر دور هستند، در یک خوشه قرار گیرند [۱].

46-Furthest-First Method

یکی دیگر از اشکالات عمده الگوریتم استاندارد k–میانگین عدم توانایی آن در کار با نقاط داده بزرگ و متراکم است. هدف روش جدید بهبود الگوریتم k–میانگین برای ایجاد خوشه‌های دقیق‌تر در نقاط داده متراکم است. استفاده از تراکم پنجره[۴۷] موجب همگن‌تر شدن خوشه‌ها می‌شود، که الگوریتم k–میانگین را برای انواع شکل و اندازه خوشه‌ها مناسب‌تر می‌سازد. این روش همچنین مدیریت نوفه و تشخیص نقاط پرت را بهبود می‌بخشد و نتایج دقیق‌تری نسبت به الگوریتم اصلی k–میانگین ایجاد می‌کند [۳٤].

در میان روش‌های متعددی که برای بهبود الگوریتم خوشه‌بندی k–میانگین و افزایش کیفیت مراکز اولیه خوشه‌ها، پیشنهاد شده‌اند، تعداد کمی از محققین به ضعف این الگوریتم در کشف خوشه‌های دلخواه پرداخته‌اند. یکی از راه حل‌های موثر برای رفع این مشکل، استفاده از فاصله نمودار[۴۸] برای اندازه‌گیری عدم تشابه بین اشیاء است. با این حال، محاسبه فاصله نمودار فرآیندی زمان‌بر است [۳۵].

برای مقابله با این چالش، الگوریتم جدیدی به نام ان‌دی‌پی k–میانگین [۴۹] براساس مفهوم قله‌های چگالی طبیعی[۵۰] پیشنهاد شده است. این روش با الهام از ایده توپ دانه‌ای[۵۱] که از توپ برای نمایش داده‌های محلی استفاده می‌کند، نمایندگانی از یک همسایه محلی را انتخاب می‌کند که به آنها قله‌های طبیعی چگالی گفته می‌شود. نتایج تجربی نشان می‌دهد که این روش نه تنها قادر به تشخیص خوشه‌های کروی است، بلکه می‌تواند خوشه‌های چندگانه را نیز شناسایی کند. بنابراین، ان‌دی‌پی k–میانگین در تشخیص خوشه‌های دلخواه مزایای بیشتری نسبت به سایر الگوریتم‌های موجود دارد [۳۵].

روش راه‌اندازی مرکز خوشه[۵۲] برای یافتن مراکز

47-Window Density Method
48-Graph Distance (GD)
49-NDPK-Means
50-Natural Density Peaks (NDPs)
51-Granular ball
52-Cluster Center Initialization Method

خوشـــه‌های اولیه در الگوریتم k–میانگین ارایه شـــد. این روش اســـتفاده از تراکم داده‌های چند مقیاســـی مبتنی بر چگالی[۵۳] اســـت و برای تخمین چگالـــی داده‌ها در یک نقطه اســـتفاده شـــده و بر اســـاس چگالی آن‌ها، نقاط را مرتب می‌کند. مزیت این روش این اســـت که، مراکز خوشه اولیه محاسبه شده، بسیار نزدیک به مراکز خوشه مورد نظر با نتایج خوشـــه‌بندی بهبود یافته و سازگار است. محدودیت اصلی روش این است که منجر به هزینه محاسباتی بالاتر می‌شود، زیرا شامل محاسبات چگالی است [۲۹].

در [۳۶] الگوریتمی برای محاســـبه مراکز خوشه اولیه به منظور خوشه‌بندی k–میانگین، بر اساس رتبه‌بندی زد–اســـکور[۵۴] بیان شـــد. این الگوریتم، یک روش آماری برای رتبه‌بندی ویژگی‌های عددی و اســـمی بر اســـاس میانگین فاصله است. پیچیدگی زمانی این روش، چند جمله‌ای درجه دو بدون کاهش دقت خوشـــه‌ها است. این روش، الگوریتم k–میانگیـــن را تنها از یک نظر دو معیـــار دقت و یا کارایی بهبود می‌بخشد.

روش دیگر به این صورت اســـت کـــه اگر از مجموعه داده‌های بزرگ برای خوشه‌بندی استفاده شود، پیچیدگی زمانی برای اســـتفاده از الگوریتم k–میانگین، بیشتر است. بنابراین، پرداختن به داده‌هایی با ابعاد بالا، با اســـتفاده از تکنیک‌های خوشـــه‌بندی از نظر تعداد بیشـــتر داده‌ها، کار پیچیده‌ای اســـت. به منظور بهبود کارایی، داده‌های نوفه‌ای و پرت ممکن اســـت رد شوند و زمان اجرا را کاهش دهند، بنابراین لازم اســـت ابعاد داده، کاهـــش یابد. در مجموعه داده اصلـــی، تجزیه و تحلیل مولفه‌هـــای اصلی، یک روش رایج برای شناسایی الگوها در داده‌های با ابعاد بالا است. از روش تجزیـــه و تحلیـــل مؤلفه‌های اصلی هســـته برای کاهش ابعاد اســـتفاده می‌شـــود و به مقداردهی اولیه بهتر مراکز برای خوشـــه‌بندی کمک می‌کند. روش پیشـــنهادی، خوشـــه‌بندی داده‌ها را با کمک مؤلفه اصلی به دست آمده از روش تجزیـــه و تحلیل مؤلفه‌های اصلی هســـته، انجام می‌دهد. مجموعه داده‌های ارائه شده را در مجموعه‌های k، خوشه‌بندی می‌کند. میانه هر مجموعه را می‌توان به‌عنوان مراکز خوشه اولیه استفاده کرد و سپس هر نقطه داده را به نزدیک‌ترین مرکز خوشه آن اختصاص داد [۳۷].

در روش بخش‌بنـــدی تصادفـــی، مجموعـــه داده به k خوشه تقسیم می‌شود. به عبارت دیگر، یک خوشه به طور تصادفی به هر نقطه داده اختصاص داده می‌شود و سپس وارد مرحله به‌روزرسانی الگوریتم k–میانگین می‌شود. این روش، ســـریع و دارای پیچیدگی زمانی خطی اســـت. ایراد اصلی این روش این اســـت که در مجموعـــه داده‌های هر خوشه، این امکان وجود دارد که نقاط داده‌ای که متعلق به همان خوشـــه واقعی است، پس از استفاده از این روش در خوشه‌های مختلف قرار گیرد [۱].

روش پیشـــنهادی دیگر، مبتنی بـــر ویژگی‌های عددی اســـت. دامنه مقادیر، بـــه تفریق مقادیر حداقـــل و حداکثر محدود می‌شـــود. مشـــکل اصلی این روش، عدم توجه به توزیـــع داده‌ها در فضای داده اســـت. به‌عنـــوان مثال، هم در داده‌های توزیع شـــده و هم در داده‌های با چگالی بالا، دانه‌های اولیه، یکســـان در نظر گرفته می‌شوند. همان‌طور که قبلا گفته شد، در این الگوریتم، از حداقل و حداکثر مقدار برای محاســـبه دانه‌های اولیه اســـتفاده می‌شود بدون آن که هیچ دانشـــی در مورد توزیع داده‌ها داشـــته باشد. پس از به‌دســـت آوردن محدوده مقادیر، به k قســـمت تقســـیم می‌شـــود. فاصله هر دانه، نســـبت به دانه‌های قبلی یکسان اســـت. در برخی داده‌ها، فاصله دانه‌ها ممکن است یکسان نباشد، در نتیجه دقت خوشـــه‌بندی را کاهش داده و زمان هم‌گرایـــی الگوریتم را افزایش می‌دهد. عدم توجه به میزان توزیع داده‌ها باعث ایجاد برخی مشـــکلات ناخواسته مانند ایجاد خوشه‌های خالی می‌شود [۳۸].

در میان الگوریتم‌های خوشـــه‌بندی موجود، الگوریتم k–میانگیـــن، بـــه دلیل ســـادگی و موثر بودن، بـــه یکی از گســـترده‌ترین روش‌های مورد استفاده برای خوشه‌بندی داده‌ها تبدیل شـــده است. این الگوریتم، با وجود سادگی و

53-Density-Based Multi Scale Data Condensation
54-Z-Score

گستردگی استفاده از آن، دارای مشکلاتی چون حساس بودن به نوفه، مقداردهی تصادفی مراکز اولیه خوشه‌بندی، وجود داده‌های پرت و اثرات آن بر خوشه‌بندی، دقت و پایداری الگوریتم، تعیین مناسب تعداد خوشه‌ها قبل از انجام خوشه‌بندی، ناتوانی الگوریتم در مدیریت داده‌هایی با مقادیر کاملا متفاوت می‌باشد. با توجه به مشکلات این الگوریتم، در سال‌های اخیر، بهبودهایی بر روی آن انجام شده است تا مشکلات مربوط به این الگوریتم را به حداقل برساند. در این بخش، ما به مرور کارهای چندین سال اخیر در جهت بهبود الگوریتم خوشه‌بندی k–میانگین پرداختیم و در ادامه و در بخش بعدی، روش پیشنهادی خود را که بهینه‌سازی الگوریتم خوشه‌بندی k–میانگین با استفاده از الگوریتم بهینه‌سازی گرگ خاکستری و الگوریتم تشخیص داده پرت جنگل جداسازی است را ارائه خواهیم داد.

## ۴. روش پیشنهادی

ایده پیشنهادی ما، بهبود الگوریتم k–میانگین با استفاده از الگوریتم هوش جمعی بهینه‌سازی گرگ خاکستری و الگوریتم جنگل جداسازی است. در الگوریتم k–میانگین، اولین مراکز خوشه‌بندی به صورت تصادفی انتخاب می‌شوند و در مراحل بعدی، با تغییر این مراکز به سمت خوشه‌های بهتر حرکت می‌شود. از آن‌جایی که دقت نهایی خوشه‌بندی k–میانگین، وابستگی قوی با اولین مراکز خوشه‌بندی دارد، قصد داریم در این پژوهش، مراکز خوشه اولیه را با استفاده از الگوریتم گرگ خاکستری پیدا کرده و به‌عنوان ورودی به الگوریتم k–میانگین بدهیم. از سوی دیگر، داده‌ها معمولا دارای نقاط پرت هستند. از آن‌جایی که هدف خوشه‌بندی، یافتن گروه داده‌هایی است که کم‌ترین فاصله را با هم و بیشترین فاصله را با خوشه‌های دیگر دارند، پس، یافتن داده‌های پرت می‌تواند خوشه‌های فشرده‌تر و منسجم‌تری را فراهم آورد. به همین دلیل، از الگوریتم جنگل جداسازی برای یافتن نقاط پرت استفاده کرده‌ایم.

مراحل روش پیشنهادی ما به طور خلاصه به شرح زیر است:

۱) اجرای الگوریتم خوشه‌بندی k–میانگین روی مجموعه داده (مبنای مقایسه و ارزیابی ایده‌ها).

۲) اجرای الگوریتم بهینه‌سازی گرگ خاکستری و سپس خوشه‌بندی k–میانگین (بدون هیچ اقدامی روی نقاط پرت).

۳) اجرای الگوریتم بهینه‌سازی گرگ خاکستری برای یافتن مراکز بهینه و سپس خوشه‌بندی با الگوریتم k–میانگین و در آخر حذف نقاط پرت هر خوشه.

٤) حذف نقاط پرت کل مجموعه داده (در ابتدای کار) با الگوریتم جنگل جداسازی، سپس اجرای الگوریتم بهینه‌سازی گرگ خاکستری برای یافتن مراکز خوشه بهینه و در آخر اجرای الگوریتم خوشه‌بندی k–میانگین.

۵) حذف نقاط پرت کل مجموعه داده (در ابتدای کار) با الگوریتم جنگل جداسازی، اجرای الگوریتم گرگ خاکستری و سپس خوشه‌بندی با الگوریتم k–میانگین و در آخر اختصاص هر نقطه پرت، به نزدیک‌ترین خوشه.

پس همان‌طور که گفتیم، قرار است پنج آزمایش متفاوت را روی ده مجموعه داده اجرا کنیم. در چهار آزمایش، ابتدا باید الگوریتم گرگ خاکستری اجرا شود. در دو آزمایش، این الگوریتم روی کل نقاط داده و در دو آزمایش دیگر، روی داده‌های غیر پرت اجرا می‌شود. هر یک از الگوریتم‌ها بیست دفعه اجرا شدند و نتیجه بیان شده، میانگین این بیست اجرا است. نتیجه آن شامل، مراکز خوشه، شماره خوشه‌ها و تعداد خوشه‌ها، ذخیره می‌شود. سپس برای آزمایش بعدی (آزمایش حذف نقاط پرت خوشه‌ها)، به جای اجرای مجدد الگوریتم گرگ خاکستری (که به زمان زیادی نیاز دارد)، از نتایج ذخیره شده آن استفاده می‌شود.

در دو آزمایش دیگر هم، کار به همین روال صورت می‌پذیرد، یعنی یک بار الگوریتم گرگ خاکستری روی داده‌هایی که نقاط پرت آن حذف شده اجرا می‌شود، و سپس نتیجه آن ذخیره می‌شود. در آزمایش دیگر که نقاط

پرت را به نزدیک‌ترین خوشـــه انتساب می‌دهیم، نیازی به اجرای مجدد الگوریتم گرگ خاکستری نیست، بلکه از نتایج ذخیره شده استفاده می‌کنیم.

در این آزمایش‌ها از شاخص رند اصلاح شده به‌عنوان معیار ارزیابی اســتفاده شــد. علاوه بر این معیار، میزان خطای k، که برابر با اختلاف kی حاصل از روش‌ها و kی واقعی داده‌ها، تعریف و محاسبه شده است، در نظر گرفته می‌شــود. برای این منظور، مجمــوع خطای kی هر روش، روی همه مجموعه داده‌ها، روی داده‌های ســنتزی و روی داده‌های واقعی، به طــور جداگانه ذخیره و در انتهای کار نمایش داده می‌شود.

الگوریتــم گرگ خاکســتری، به طور بهینــه نقاطی را به‌عنوان مرکز خوشه انتخاب می‌کند که با حذف خوشه‌های تهــی، بتوان بــه نزدیک‌ترین مقدار k، بــه داده‌های واقعی دست یافت. همان‌طور که می‌دانیم یکی از مشکلات مهم در الگوریتم k-میانگین یافتن مقدار k است که باید به صورت آزمایــش خطا یا روش‌های غیرخودکار به طور تقریبی به دست آید. در پژوهش جاری، در آزمایش پایه k-میانگین، بهتریــن k را با تکرار الگوریتم روی مقادیر مختلف k از ۲ تا i/۲ (که i، تعداد نمونه داده‌ها اســت)، به دست آوردیم و هــرگاه تا پس از ده تکرار، مقدار دقت قبلی الگوریتم بهبود نیابــد، الگوریتم را متوقف کردیم. اما در چهار آزمایشــی که مراکز خوشــه با استفاده از الگوریتم بهینه‌سازی گرگ خاکستری، پیدا شده‌اند، هیچ عملی در این خصوص انجام نشــده است و خود الگوریتم بهینه‌سازی گرگ خاکستری، مراکز خوشــه‌ای را پیدا کرده که پس از حذف خوشه‌های تهــی، نزدیک‌ترین فاصلــه را با تعداد خوشــه‌های واقعی داشته است.

در الگوریتــم اصلی گرگ خاکســتری، جمعیت اصلی بــه ترتیب کم‌ترین شایســتگی به بیشــترین شایســتگی، مرتب‌سازی می‌شود، از آن‌جایی که مسئله مربوطه از نوع کمینه‌سازی بوده است، ولی در کار جاری، هدف ما، کسب بیشــترین مقدار برای شاخص رند اصلاح شده است، چرا که راه‌حل با امتیاز بزرگ‌تر بهتر اســت. به همین دلیل، در زمان مرتب‌سازی، از گزینه معکوس استفاده کردیم، به این صــورت، جمعیت راه‌حل‌ها، به‌ترتیب بزرگ‌تر به کوچک‌تر مرتب می‌شوند. جمعیت ساخته شده در سیستم پیشنهادی پنجاه در نظر گرفته شــده است. لازم به ذکر است که پس از اجرای الگوریتم با تعداد جمعیت‌های مختلف، عدد پنجاه انتخاب شــد که توازن خوبی بین دقت حاصل و پیچیدگی زمانی ایجاد می‌کنــد. تعداد تکرار الگوریتــم نیز به همین ترتیــب برابر با صد انتخاب گردید. در پایان پس از اجرای الگوریتم، بهترین راه‌حل‌های یافت شــده یعنی راه‌حل‌هایی با بیشترین مقدار شایستگی (بیشتر مقدار امتیاز شاخص رند اصلاح شده) برگشت داده می‌شود تا برای خوشه‌بندی الگوریتم k-میانگین مورد استفاده قرار گیرند.

**۴-۱ مراحل تفصیلی روش پیشنهادی به شرح زیر است:**

**ورودی:**

۱. مجموعه‌ای از n نقطه داده و m بعد ویژگی.

۲. تعداد گرگ‌ها در دسته = ۵۰.

۳. ضریب کنترل، a، A و C.

٤. حداکثر تعداد تکرار = ۱۰۰.

**خروجی:**

۱.بهتریــن موقعیت سراســری گــرگ، بهترین ارزش شایستگی در نهایت ایجاد مجموعه‌ای از k خوشه با مراکز خوشه بهینه.

**مراحل:**

۱. اجرای الگوریتم جنگل جداســازی بر روی مجموعه داده‌ها ( آزمایش ٤ و ٥).

۲. تولید جمعیت اولیه گرگ‌ها.

۳. تنظیم تعداد تکرار.

٤. محاسبه ارزش شایستگی برای هر گرگ.

٥. شناسایی سه گرگ برتر از نظر بالاترین شایستگی، یعنی آلفا، بتا و دلتا.

٦. تــا وقتی که تعداد تکرار کمتر از حداکثر تعداد تکرار باشد:

٦,١. بروزرسانی موقعیت گرگ‌های آلفا، بتا و دلتا برای هر گرگ.

٦,٢. بروزرسانی ضریب کنترل، a، A و C.

٦,٣. محاسبه ارزش شایستگی برای تمامی گرگ‌ها.

٦,٤. مرتب‌سازی برحسب بالاترین ارزش شایستگی.

۷. برگرداندن گرگ با بالاترین ارزش شایستگی.

۸. اجـرای الگوریتم k-میانگین با مراکز خوشـه بهینه ایجاد شـده توسط گرگ خاکستری (گرگ‌هایی با بالاترین ارزش شایستگی).

۹. انتقال داده‌ها به خوشه‌ها.

۱۰. اجرای الگوریتم جنگل جداسـاز برای حذف نقاط پرت (آزمایش ۳).

۱۱. انتسـاب نقاط پرت به نزدیک‌ترین مراکز خوشـه (آزمایش ٥).

۱۲. پایان.

**نکتـه**: مدل کردن فرآیند شـکار، زمانی کـه حمله به رهبری گرگ آلفا شـروع می‌شود، با اسـتفاده از کاهش بـردار a انجام می‌شـود. A بـرداری تصادفـی در بازه [a۲، ۲a -] است که با کاهش a، بردار ضرایب A هم کاهش می‌یابد. اگر قدر مطلق A بزرگ‌تر از عدد یک باشد، گرگ‌ها برای ردیابی شکار از یک‌دیگر دور شده‌اند و اگر قدر مطلق A کوچک‌تر از این عدد باشـد، گرگ‌ها برای حمله به شکار به‌هم نزدیک شـده‌اند. بردار C هم به‌عنوان موانع موجود در طبیعت به شـکار، وزن داده و شـکار را برای گرگ‌ها غیرقابل دسـتیابی‌تر می‌کند. روندنمای الگوریتم فوق در پیوست۲ آمده است.

### ۴-۲ ارزیابی الگوریتم پیشنهادی

مبنـای مقایسـه و ارزیابی ایدهٔ پیشـنهادی، الگوریتم اصلی خوشـه‌بندی k-میانگین است. این الگوریتم به زبان پایتون در تابعی پیاده‌سازی شد که یکی از ورودی‌های آن، آرایه مراکز اولیه خوشه‌بندی با مقدار پیش فرض تهی ([]) است. به این صورت، برای خوشه‌بندی بر اساس الگوریتم بهینه‌سازی گرگ خاکسـتری، مراکز خوشه تولید شده با این الگوریتم، به‌عنوان مقدار این پارامتر اسـتفاده می‌شود و برای حالت مبنای مقایسـه، مقداری برای این آرایه وارد نمی‌شود تا مراکز خوشه اولیه، به صورت تصادفی تولید شوند.

پس از پیاده‌سـازی این الگوریتم، نوبـت به الگوریتم بهینه‌سازی گرگ خاکستری می‌رسد. در این الگوریتم ابتدا جمعیت اولیه گرگ‌ها ساخته شـود. جمعیت ساخته شده که در سیسـتم پیشنهادی، پنجاه در نظر گرفته شده است، بر اسـاس میزان شایستگی، مرتب‌سـازی می‌شود. برای الگوریتم جنگل جداسازی هم از کتابخانه استاندارد و آماده سایکیت‌لرن استفاده شد.

#### ۴-۲-۱ مجموعه داده‌ها

در ایـن پژوهش، از هر دو نوع مجموعه داده سـنتزی و واقعی اسـتفاده کردیم. داده‌های سنتزی، به‌وسیلهٔ تابع بلاب[55] از ماژول مجموعه داده[56] از کتابخانه سـایکیت‌لرن در چندین مقدار متفاوت، تعداد نمونه، تعداد خوشه و تعداد ویژگی تولید شدند. مشخصات مجموعه داده‌های سنتزی در جدول ٤-۱، خلاصه شده است:

جدول ۴-۱. مشخصات مجموعه داده‌های سنتزی

| تعداد نمونه | تعداد ویژگی | تعداد خوشه |
|---|---|---|
| ۱۰۰ | ۳ | ۵ |
| ۱۰۰ | ۶ | ۵ |
| ۱۰۰ | ۳ | ۲۰ |
| ۱۰۰ | ۶ | ۲۰ |
| ۲۰۰ | ۳ | ۵ |
| ۲۰۰ | ۶ | ۵ |
| ۲۰۰ | ۳ | ۲۰ |
| ۲۰۰ | ۶ | ۲۰ |

علـت اسـتفاده از داده‌های سـنتزی، انعطاف موجود در تولید داده‌هایی با مشـخصات مختلف، برای بررسـی تاثیر هر آزمایش اسـت. به عبـارت بهتر، در یک مجموعه داده واقعـی، تعداد ویژگی‌هـا ثابت بـوده و نمی‌توان به

55-Blob
56-DataSets

طور تصادفی ویژگی‌ها را افزایش یا کاهش داد. ولی در مجموعه داده سنتزی، با انعطاف کامل، مجموعه‌هایی با مشخصات دلخواه تولید می‌کنیم. به‌عنوان مثال، اگر میزان بهبود سیستم پیشنهادی روی داده‌هایی با ویژگی‌های بیشتر یا خوشه‌های بیشتر، کاهش یابد، متوجه آن خواهیم شد (یا هرگونه روند دیگری که در ماهیت الگوریتم‌ها، وجود داشته باشد).

از دو مجموعه داده واقعی نیز استفاده کردیم:

۱) داده‌های قطعه‌بندی تصویر که مربوط به تصاویری از ۷ خوشه متفاوت است.

۲) داده‌های مربوط به اشیای واقع در زیر آب، که در یکی از خوشه‌های سنگ یا مین زیردریایی قرار می‌گیرند.

با توجه به تعداد ویژگی‌های زیاد مجموعه داده‌های واقعی (۱۹ ویژگی در مجموعه اول و ۶۰ ویژگی در مجموعه دوم) و نیز هدف آزمایش، یعنی بررسی تاثیر الگوریتم‌ها در هر آزمایش، برای هر خوشه، ۲۰ نمونه داده انتخاب شد تا کارایی روش‌ها با شرایط یکسان مورد ارزیابی قرار گیرد. مجموعه داده‌های واقعی از داده‌های موجود در اینترنت، برای خوشه‌بندی از سامانه گیت‌هاب، تهیه شده است. برای هر مجموعه داده، پنج آزمایش گفته شده در قسمت‌های قبل اجرا گردید.

**۴-۲-۲ معیارهای ارزیابی**

معیارهای متعددی برای ارزیابی کارایی الگوریتم‌های خوشه‌بندی وجود دارد. برخی از این معیارها نیازمند اطلاعات مربوط به تعداد خوشه‌ها در نمونه‌ها هستند. در حالی که برخی دیگر بدون نیاز به این اطلاعات، قابل استفاده می‌باشند. به منظور مشاهده تاثیر الگوریتم‌های پیشنهادی (گرگ خاکستری و جنگل جداسازی) نسبت به الگوریتم پایه k-میانگین، از مجموعه داده‌های برچسب دار استفاده کردیم، یعنی خوشه هر نمونه داده از قبل مشخص است. معیارهای ارزیابی خوشه‌بندی، در ماژول متریک، از کتابخانه سایکیت‌لرن، موجود است. با مراجعه به مستندات این کتابخانه شاخص رند اصلاح شده برای ارزیابی نتایج هر دسته از آزمایش‌ها و نیز به‌عنوان تابع شایستگی الگوریتم گرگ خاکستری استفاده کردیم. این معیار، از برچسب‌های داده شماره خوشه واقعی هر نمونه و شماره خوشه‌های پیش‌بینی شده در روش‌های پیشنهادی استفاده می‌کند.

هم‌چنین، بازه مقادیر شاخص رند اصلاح شده، بین ۱- تا ۱ است. اگر خوشه‌بندی کاملا دقیق باشد (یعنی همه داده‌ها به درستی خوشه‌بندی شوند)، امتیاز ۱ را دریافت می‌کنیم. هر چه مقدار این امتیاز به ۱ نزدیک‌تر باشد، خوشه‌بندی با کیفیت‌تری انجام شده است و برعکس. هم چنین، با توجه به هدف کار جاری که شناسایی روشی است که خوشه‌بندی نزدیک‌تری با تعداد خوشه‌های واقعی داده‌ها داشته باشد، معیار خطای k را تعریف و محاسبه کردیم که با قدر مطلق اختلاف تعداد خوشه واقعی و تعداد خوشه پیش‌بینی شده برابر است. به عبارت دیگر، روشی که تعداد خوشه‌های آن به تعداد خوشه‌های واقعی داده نزدیک‌تر است، از نظر ما بهتر است و خطای کمتری دارد.

### ۴-۳ نتایج پیاده سازی

قبل از ارائه نتایج، آزمایش‌های انجام شده را مورد بحث قرار می‌دهیم:

الف) آزمایش پایه k-میانگین

در این آزمایش، خوشه‌بندی داده‌ها با استفاده از روش k-میانگین و بدون هیچ تغییر اضافه‌ای انجام می‌شود. همان‌طور که می‌دانیم، در الگوریتم k-میانگین امکان برگشت خوشه‌های تهی وجود دارد، که معمولا برای حل این مسئله، یا نقاط تصادفی به خوشه‌های تهی اختصاص داده می‌شود یا خوشه‌های تهی حذف شده و در نظر گرفته نمی‌شوند. در این آزمایش، ما از روش دوم استفاده کرده و خوشه‌های تهی را حذف می‌کنیم. این راه‌حل برای همه روش‌ها به صورت یکسان انجام شده و در الگوریتم پایه k-میانگین پیاده‌سازی شده است.

ب) آزمایش گرگ خاکستری

این الگوریتم دقیقا همانند الگوریتم قبلی است، با این

تفـاوت که به جای انتخاب تصادفی اولین مراکز خوشـه، از الگوریتم بهینه‌سـازی گرگ خاکستری استفاده کرده و مراکز اولیه بهینه را پیدا می‌کنیم.

پ) آزمایـش گـرگ خاکسـتری و حذف نقـاط پرت خوشه‌ها

در این آزمایش پس از اجـرای آزمایش دوم، به کمک الگوریتم جنگل جداسازی، نقاط پرت خوشه‌ها را پیدا کرده و حذف می‌کنیم.

ت) حـذف نقـاط پـرت کل داده، قبـل از اعمال گرگ خاکستری

در این آزمایش، قبل از یافتن مراکز خوشه بهینه، نقاط پرت را با الگوریتم جنگل جداسازی، حذف می‌کنیم و ادامه کار آن مانند آزمایش دوم انجام می‌شود.

ث) شناسـایی و حذف نقاط پرت قبـل از اعمال گرگ خاکستری و سـپس اختصاص نقاط پرت به نزدیک‌ترین خوشه‌ها

در این آزمایش ابتدا نقاط پرت را شناسایی و از داده‌ها حذف می‌کنیم. سپس با استفاده از گرگ خاکستری، مراکز بهینه پیدا شده و خوشه‌بندی انجام می‌شود (مانند آزمایش قبل). در انتها، نقاط پرت شناسـایی شده و به نزدیک‌ترین مراکز خوشه اختصاص می‌یابند.

به منظور سهولت اشاره به آزمایش‌ها، در جدول ٤-١، برای هر کدام شماره‌ای در نظر گرفتیم:

جدول ۴-۱. شمارهٔ هر آزمایش

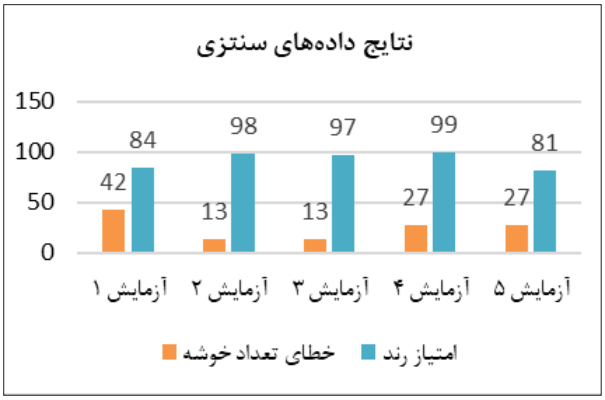
| نام آزمایش | شماره |
|---|---|
| k-میانگین | ۱ |
| گرگ خاکستری + k-میانگین | ۲ |
| گرگ خاکستری + k-میانگین + جنگل جداسازی | ۳ |
| جنگل جداسازی + گرگ خاکستری + k-میانگین | ۴ |
| جنگل جداسازی + گرگ خاکستری + k-میانگین + انتساب نقاط پرت به نزدیک‌ترین خوشه‌ها | ۵ |

در ادامـه این بخش، بـه ارائه نتایـج می‌پردازیم. ابتدا نتایج مربوط به داده‌های سـنتزی، بیان می‌شود. سپس به داده‌هـای واقعی می‌پردازیم. و در آخر نیز با تجمیع نتایج، کارایی کلی سیستم بیان می‌شود.

### ۱) نتایج آزمایش‌ها روی داده‌های سنتزی

همان‌طور که قبلا در بخش مجموعه داده‌ها گفتیم، علت استفاده از داده‌های سنتزی (در مقایسه با داده‌های واقعی) انعطاف کامـل در تولید داده‌هایی با مشـخصات دلخواه اسـت. به‌عنوان مثال دو مجموعه داده با تعداد صد نمونه و پنج خوشـه تولید کردیم که یکی از آن‌ها سـه و دیگری شـش ویژگی دارد. به این صورت می‌توانیم متوجه شویم که بهبود سیستم پیشنهادی با افزایش تعداد ویژگی‌ها چه تغییری می‌کند و در حقیقت به کاربردهای مناسب سیستم پـی ببریم. دو مجموعه با تعداد صد نمونه و سـه ویژگی داریم که یکی پنج خوشـه و دیگری بیست خوشه دارد و به این صورت متوجه می‌شـویم که میزان بهبود سیستم پیشنهادی با افزایش تعداد خوشه، کمتر یا بیشتر می‌شود.

در نمـودار ٤-١، کارایی هر یک از آزمایش‌ها بر روی تمام مجموعه داده‌ها به صورت کلی نشان داده شده است:




نمودار ۴-۱. کارایی هر یک از آزمایش‌ها

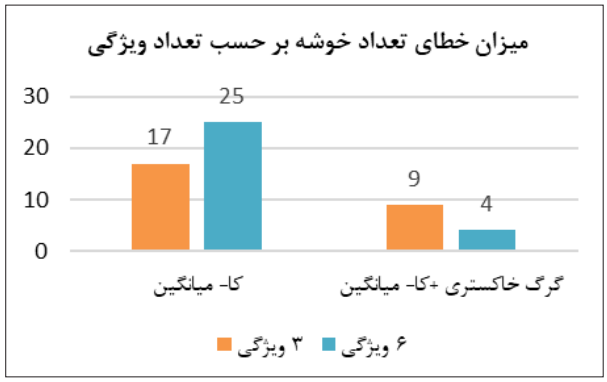
خطای تعداد خوشـه، میزان خطای k (مجموع اختلاف بین خوشـه‌های واقعـی و پیش‌بینی شـده) و امتیاز رند، میانگین امتیاز شـاخص رند اصلاح شـده را برای روش های مختلف نشان می‌دهد.

همان‌طور که در این نمـودار می‌بینیم، همه روش‌های پیشنهادی دارای میزان خطای کمتر (میله نارنجی) و دقت

بیشتر (میله آبی) است. کمترین مقدار خطای k، در آزمایش شماره دو و آزمایش شماره سه به دست آمده است. علت مشابه بودن تعداد خوشه‌ها در آزمایش‌های دو و سه و نیز آزمایش‌های چهار و پنج، آن است که آزمایش سه، از خوشه‌های به‌دست آمده در آزمایش دو استفاده می‌کند و صرفا نقاط پرت خوشه‌ها را حذف می‌کند، در نتیجه شاهد تغییری در تعداد خوشه‌ها نبوده‌ایم. برای آزمایش‌های چهار و پنج نیز، همین مسئله برقرار است؛ یعنی در آزمایش پنج، هیچ خوشه‌ای حذف یا اضافه نمی‌شود، بلکه نقاط پرت به خوشه‌های به دست آمده در آزمایش چهار اختصاص می‌یابند.

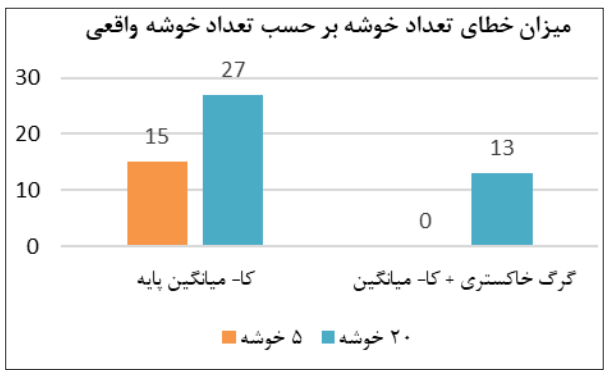
از نظر امتیاز شاخص رند اصلاح شده، اگرچه بیشترین امتیاز در آزمایش چهار به‌دست آمده است، ولی میزان خطای این آزمایش خیلی بیشتر از آزمایش‌های دو و سه است. به عبارت دیگر، اولویت اول کیفیت خوشه‌بندی، نزدیک‌تر بودن تعداد خوشه واقعی و تعداد خوشه پیش‌بینی شده است، و با در نظر داشتن این نکته، آزمایش دوم، یعنی ترکیب الگوریتم گرگ خاکستری به همراه k-میانگین، بهترین نتیجه را داشته است. پس در داده‌های سنتزی، الگوریتم جنگل جداسازی، هیچ‌گونه بهبودی حاصل نداشته است یا به عبارت بهتر، حذف نقاط پرت در داده‌های سنتزی، خوشه‌های باکیفیت‌تر تولید نمی‌کند. دلیل این موضوع را می‌توان این‌گونه بیان کرد که، نقاط پرت در دنیای واقعی، در نتیجه خطاهای اندازه‌گیری، دستکاری اطلاعات، خطاهای انسانی، ماهیت پدیده‌های طبیعی و غیره در داده‌ها ظاهر می‌شوند که این مسئله در داده‌های تصادفی سنتزی صدق نمی‌کند. پس باید عملکرد الگوریتم جنگل جداساز و حذف نقاط پرت در عملیات خوشه‌بندی را در داده‌های واقعی مورد بررسی قرار دهیم که این کار در بخش بعدی انجام می‌شود.

در نمودار ٤-٢، میزان خطای تعداد خوشه برحسب تعداد ویژگی نشان داده شده است:




نمودار ۴-۲. میزان خطای تعداد خوشه برحسب تعداد ویژگی

به خوبی می‌بینیم که در هر دو حالت، گرگ خاکستری، خطای کمتری داشته، اما برای تعداد ویژگی بیشتر، میزان کاهش خطا یا بهبود دقت خوشه‌بندی به میزان قابل ملاحظه‌ای بیشتر بوده است، به عبارت بهتر، روش پیشنهادی، در داده‌های پیچیده‌تر، کارایی بالاتری دارد. در همه آزمایش‌های انجام شده، امتیاز شاخص رند اصلاح شده، در روش گرگ خاکستری، بالاتر از روش پایه k-میانگین بوده که باز هم گواه برتری روش پیشنهادی، نسبت به روش پایه بوده و به همین دلیل در تحلیل‌های فوق، به این امتیاز اشاره‌ای نکردیم. در تحلیل بعدی، میزان بهبود روش پیشنهادی را بر حسب تعداد خوشه‌های واقعی ارزیابی کردیم که نتیجه آن در نمودار ٤-٣، نشان داده شده است. هدف این تحلیل آن است که ببینیم کارایی سیستم در داده‌هایی با تعداد خوشه کمتر، یا در داده‌هایی با تعداد خوشه بیشتر، در کدام یک بالاتر بوده است:

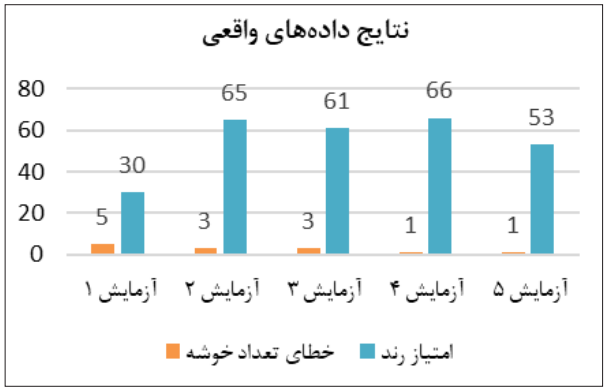



نمودار ۴-۳. میزان خطای تعداد خوشه بر حسب تعداد خوشه واقعی

نمودار فوق نشان می‌دهد که سیستم پیشنهادی، خطای

تعداد خوشـــه‌بندی را در داده‌هایی با پنج خوشـــه، از ۱۵ به ۰ کاهش داده اســـت. همچنین، خطـــا را در داده‌هایی با ۲۰ خوشـــه، از ۲۷ به ۱۳ کاهش داده اســـت و تفاوت قابل ملاحظه‌ای بین بهبـــود حاصل در این دو نوع داده متفاوت (از نظر تعداد خوشـــه) دیده نشد. حال به بررسی عملکرد سیستم، روی داده‌های واقعی می‌پردازیم.

نمودار ٤-٤، نتایج داده‌های واقعی را نشان می‌دهد:



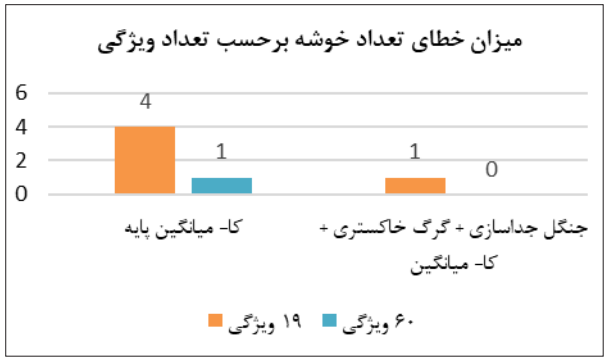

نمودار ۴-۴. نتایج داده‌های واقعی

بـــا توجه به داده‌های جدول و نمودار فوق، به روشـــنی می‌بینیم که، الگوریتم پیشـــنهادی به طـــور قابل ملاحظه‌ای توانسته است دقت و خطای خوشه‌بندی را در هر دو مجموعه داده کاملا متفاوت بهبود دهد. کم‌ترین خطا و بیشترین مقدار امتیاز شاخص رند اصلاح شـــده، در آزمایش شماره چهار به‌دست آمده است. در آزمایش شماره چهار (جنگل جداسازی + گرگ خاکستری + k-میانگین)، ابتدا داده‌های پرت به کمک الگوریتم جنگل جداسازی شناسایی و حذف شده‌اند. سپس با استفاده از الگوریتم گرگ خاکستری، مراکز خوشه‌ی بهینه پیدا شده و به وسیله الگوریتم k-میانگین، خوشه‌بندی انجام شده اســـت. بر خلاف مجموعه داده‌های سنتزی، که حذف نقاط پرت در آن‌ها بهبودی حاصل نداشـــته است، دیدیم که در داده‌های واقعی، شناسایی و حذف نقاط پرت، کمک قابل ملاحظه‌ای به خوشه‌بندی اطلاعات ارائه می‌دهد.

همان‌طـــور که گفتیم، دلیل این موضـــوع را می‌توان به منبع تولید نقاط پرت نسبت داد. نقاط پرت از منابع مختلفی مانند خطاهای انســـانی، خطاهای اندازه‌گیری، دســـتکاری داده‌ها و غیره در داده‌ها ظاهر می‌شوند، که هیچ یک از این منابع، در داده‌های سنتزی وجود ندارد، پس انتظار می‌رود که شناسایی و حذف نقاط پرت در داده‌های سنتزی کمکی به بهبود خوشه‌بندی نداشته باشد. اما در داده‌های واقعی، بـــا توجه به ماهیت و روش به دســـت آمـــدن آن‌ها، امکان بـــروز انواع خطا وجـــود دارد. حتی گاهـــی ماهیت طبیعی داده‌های واقعی به گونه‌ای است که دارای نقاط پرت باشند. از ایـــن رو، در چنین مواردی شناســـایی و حذف آن‌ها، به خوشه‌بندی بهتر داده‌ها کمک می‌کند.

برای تایید برتری سیســـتم پیشنهادی نسبت به روش پایه k-میانگین، در ادامه دو تحلیل قبلی انجام شـــده برای داده‌های سنتزی را روی داده‌های واقعی نیز اعتبارسنجی می‌کنیم. در تحلیـــل اول، به تاثیر تعداد ویژگی‌ها بر میزان بهبود حاصل از سیستم پیشنهادی می‌پردازیم.

نمودار ٤-٥، میزان خطای تعداد خوشه بر حسب تعداد ویژگی را نشان می‌دهد:

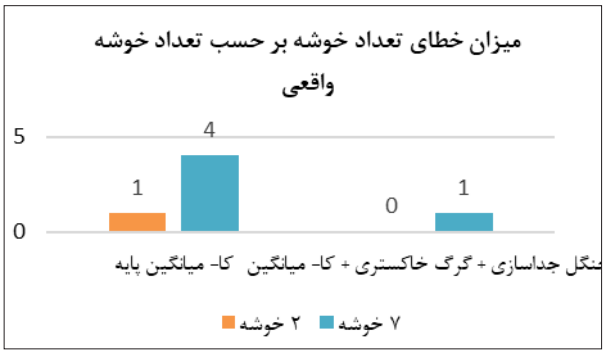



نمودار ۴-۵. میزان خطای تعداد خوشه بر حسب تعداد ویژگی

در نگاه اول به نمودار فوق متوجه می‌شـــویم که میزان کاهش خطا یا بهبود الگوریتم پیشـــنهادی برای داده‌هایی با تعداد ویژگی کمتر، خیلی بیشـــتر بوده اســـت (خطا از ٤ به ۱ کاهش یافته اســـت). اما با نـــگاه به نتایج آزمایش‌ها روی داده‌هـــای واقعی متوجه می‌شـــویم دقـــت روش پایه، برای داده‌هایی با تعداد ویژگی بالا، بسیار ناچیز و تقریبا نزدیک به صفر (۰/۰۸) است، در حالی که الگوریتم پیشنهادی (آزمایش چهار)، دقتی بسیار بالا (۰/۰۸٦) را کسب کرده است.

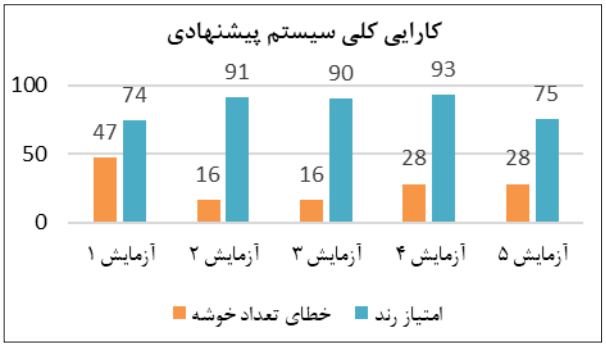
این موضوع گواه برکارایی الگوریتم پیشنهادی برای داده‌های پیچیده (با تعداد ویژگی زیاد) است. این موضوع تاییدکننده نتیجه قبلی ما در داده‌های سنتزی است. در تحلیل بعدی، به تاثیر تعداد خوشه‌های واقعی بر میزان بهبود روی سیستم پیشنهادی، پرداخته‌ایم.

نمودار ٤-٦، میزان خطای تعداد خوشه بر حسب تعداد خوشه واقعی را نشان می‌دهد:




نمودار ۴-۶. میزان خطای تعداد خوشه بر حسب تعداد خوشه واقعی

در داده‌های سنتزی دیدیم که تعداد خوشه‌های واقعی داده‌ها، تاثیر قابل ملاحظه‌ای بر میزان بهبود سیستم پیشنهادی نداشته است. در این جا می‌بینیم که، میزان کاهش خطا، در داده‌های واقعی با ۷ خوشه، خیلی بیشتر از داده‌هایی با ۲ خوشه است. به عبارت دیگر می‌توان گفت، هرچه تعداد خوشه‌های واقعی بیشتر و عملا داده‌ها پیچیدگی بیشتری داشته باشند، میزان بهبود حاصل از روش پیشنهادی، بیشتر خواهد بود. در نهایت نمودار ٤-٧، کارایی کلی الگوریتم پیشنهادی را نشان می‌دهد:




نمودار ۴-۷. کارایی کلی الگوریتم پیشنهادی

همان طور که در نمودار ٤-٧ می‌بینیم، روش پیشنهادی ارایه شده در این مقاله هم از نظر شاخص رند اصلاح شده و هم از نظر خطای تعداد خوشه‌ها پیشرفت قابل ملاحظه‌ای را در مقایسه با k-میانگین پایه نشان می‌دهد. به گونه‌ای که در شاخص رند اصلاح شده، امتیاز آن از ۰/۷٤ به ۰/۹۳ در آزمایش چهارم ارتقا یافت. همچنین از نظرمیانگین تعداد خطای خوشه‌ها، میزان خطای آن از عدد ۰/٤۷ به ۰/۱٦ در آزمایش‌های دوم و سوم رسید.

## ۵. نتیجه‌گیری

با توجه به تاثیر مراکز خوشه اولیه در کیفیت خوشه‌های تولید شده در الگوریتم خوشه‌بندی k-میانگین، تصمیم گرفتیم مراکز خوشه اولیه را با استفاده از الگوریتم گرگ خاکستری پیدا کرده و سپس خوشه‌بندی را اجرا کنیم. از سوی دیگر، نقاط پرت، بخشی از ماهیت داده‌های جهان واقعی است که به دلایل مختلف مانند خطاهای اندازه‌گیری، خطاهای انسانی، دستکاری اطلاعات و یا ماهیت طبیعی داده‌ها ظاهر می‌شوند. به همین دلیل، نقاط پرت داده‌ها را با استفاده از الگوریتم جنگل جداسازی شناسایی و حذف کردیم. پس از انجام آزمایش‌های مختلف روی مجموعه داده‌های متنوع در این آزمایش‌ها، کارایی قابل ملاحظه الگوریتم پیشنهادی نسبت به روش پایه الگوریتم k-میانگین اثبات گردید.

در آزمایش‌های انجام شده، الگوریتم گرگ خاکستری، همواره توانسته است بهترین مراکز خوشه را پیدا کرده که پس از خوشه‌بندی با الگوریتم k-میانگین و حذف خوشه‌های تهی، کمترین خطا را داشته است. از سوی دیگر، الگوریتم جنگل جداسازی توانسته است با شناسایی نقاط پرت در داده‌های واقعی، خوشه‌بندی دقیق و بسیار نزدیکی با خوشه‌های واقعی داده‌ها به دست آورد. در این پژوهش، دو پارامتر الگوریتم گرگ خاکستری یعنی، تعداد جمعیت اولیه و تعداد تکرار الگوریتم، به صورت آزمایش و خطا به دست آمد. انتظار می‌رود در کارهای آینده بتوان این پارامترها را با روش‌های دقیق‌تر و به صورتی بهینه‌تر

پیـــدا کرد تا بهبـــود حاصل از این الگوریتـــم افزایش یابد. همچنین، در کارهای آینده می‌تـــوان از مجموعه داده‌های واقعی بیشـــتری برای اثبات جامع‌تر کارایی این الگوریتم پیشنهادی استفاده کرد.

## مراجع


[1] Gan, G., & Ng, M. K. P. (2017). K-means clustering with outlier removal. Pattern Recognition Letters, 90, 8-14.

[2] Khan, F. (2012). An initial seed selection algorithm for k-means clustering of georeferenced data to improve replicability of cluster assignments for mapping application. Applied Soft Computing, 12(11), 3698-3700.

[3] Nazeer, K. A., & Sebastian, M. P. (2009, July). Improving the Accuracy and Efficiency of the k-means Clustering Algorithm. In Proceedings of the world congress on engineering (Vol. 1, pp. 1-3). London, UK: Association of Engineers London.

[4] Yedla, M., Pathakota, S. R., & Srinivasa, T. M. (2010). Enhancing K-means clustering algorithm with improved initial center. International Journal of computer science and information technologies, 1(2), 121-125.

[5] Singh, H., & Kaur, K. (2013). Review of existing methods for finding initial clusters in K-means algorithm. International Journal of Computer Applications, 68(14).

[6] Velmurugan, T., & Santhanam, T. (2011). A survey of partition-based clustering algorithms in data mining: An experimental approach. Information Technology Journal, 10(3), 478-484.

[7] Younus, Z. S., Mohamad, D., Saba, T., Alkawaz, M. H., Rehman, A., Al-Rodhaan, M., & Al-Dhelaan, A. (2015). Content-based image retrieval using PSO and k-means clustering algorithm. Arabian Journal of Geosciences, 8, 6211-6224.

[8] Pandey, A., & Shukla, M. (2014). Survey performance approach k-Mean and k-Mediod clustering algorithm. Binary Journal of Data Mining & Networking, 4(1), 14-16.

[9] Patel, A., & Singh, P. (2013). New Approach for K-mean and K-medoids Algorithm. International Journal of Computer Applications Technology and Research, 2(1), 1-5.

[10] Hariri, S., Kind, M. C., & Brunner, R. J. (2019). Extended isolation forest. IEEE Transactions on Knowledge and Data Engineering, 33(4), 1479-1489.

[11] Wu, M., Li, X., Liu, C., Liu, M., Zhao, N., Wang, J., ... & Zhu, L. (2019). Robust global motion estimation for video security based on improved k-means clustering. Journal of Ambient Intelligence and Humanized Computing, 10, 439-448.

[12] Sinaga, K. P., & Yang, M. S. (2020). Unsupervised K-means clustering algorithm. IEEE access, 8, 80716-80727.

[13] Sharma, I., Kumar, V., & Sharma, S. (2022). A comprehensive survey on grey wolf optimization. Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 15(3), 323-333.

[14] Selvaraj, S., & Choi, E. (2020, January). Survey of swarm intelligence algorithms. In Proceedings of the 3rd International Conference on Software Engineering and Information Management (pp. 69-73).

[15] Purushothaman, R., Rajagopalan, S. P., & Dhandapani, G. (2020). Hybridizing Gray Wolf Optimization (GWO) with Grasshopper Optimization Algorithm (GOA) for text feature selection and clustering. Applied Soft Computing, 96, 106651.

[16] Tekerek, A. D. E. M., & Dörterler, M. U. R. A. T. (2020). The adaptation of gray wolf optimizer to data clustering. Politeknik Dergisi, 1-1.

[17] Ahmadi, R., Ekbatanifard, G., & Bayat, P. (2021). A modified grey wolf optimizer-based data clustering algorithm. Applied Artificial Intelligence, 35(1), 63-79.

[18] Mosavi, S. K., Jalalian, E., Soleimenian, F., & Branch, U. (2018). A comprehensive survey of grey wolf optimizer algorithm and its application. Int. J. Adv. Robot. Expert Syst., 1(6), 23-45.

[19] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software, 69, 46-61.

[20] Gao, Z. M., & Zhao, J. (2019). An improved grey wolf optimization algorithm with variable weights. Computational Intelligence and Neuroscience.

[21] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2012). Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data (TKDD), 6(1), 1-39.

[22] Karczmarek, P., Kiersztyn, A., Pedrycz, W., & Al, E. (2020). K-Means-based isolation forest. Knowledge-based systems, 195, 105659.

[23] Guo, T., Yan, J., Chen, J., & Yu, Y. (2023). Overflow Capacity Prediction of Pumping Station Based on Data Drive. Water, 15(13), 2380.

[24] Xiao, B., Wang, Z., Liu, Q., & Liu, X. (2018). SMK-means: an improved mini batch k-means algorithm based on mapreduce with big data. Computers, Materials & Continua, 56(3).

[25] Geng, Zhang, Chengchang Zhang, Huayu Zhang. (2018). Improved K-means Algorithm Based on Density Canopy. Knowledge-Based Systems.

[26] Sukumar, J. A., Pranav, I., Neetish, M. M., & Narayanan, J. (2018, September). Network intrusion detection using improved genetic k-means algorithm. In 2018 international conference on advances in computing, communications and informatics (ICACCI) (pp. 2441-2446). IEEE.

[27] Yu, S. S., Chu, S. W., Wang, C. M., Chan, Y. K., & Chang, T. C. (2018). Two improved k-means algorithms. Applied Soft Computing, 68, 747-755.

[28] Xu, H., Yao, S., Li, Q., & Ye, Z. (2020, September). An improved k-means clustering algorithm. In 2020 IEEE 5th international symposium on smart and wireless systems within the conferences on intelligent data acquisition and advanced computing systems (IDAACS-SWS) (pp. 1-5). IEEE.

[29] Kodinariya, T. (2014). Survey on existing methods for selecting initial centroids in K-means clustering. International

journal of Engineering Development And Research, vol. 2, pp. 2865-2868.
[30] Ay, M., Özbakır, L., Kulluk, S., Gülmez, B., Öztürk, G., & Özer, S. (2023). FC-Kmeans: Fixed-centered K-means algorithm. Expert Systems with Applications, 211, 118656.
[31] Ye, T., Ye, J., & Wang, L. (2023). Improved rough K-means clustering algorithm based on firefly algorithm. International Journal of Computing Science and Mathematics, 17(1), 1-12.
[32] Vu, V. V., & Labroche, N. (2017). Active seed selection for constrained clustering. Intelligent Data Analysis, 21(3), 537-552.
[33] Chithra, P. L. (2017). Premeditated initial points for K-Means Clustering. IJCSIS, 15(9).
[34] Hu, H., Liu, J., Zhang, X., & Fang, M. (2023). An Effective and Adaptable K-means Algorithm for Big Data Cluster Analysis. Pattern Recognition, 139, 109404.
[35] Cheng, D., Huang, J., Zhang, S., Xia, S., Wang, G., & Xie, J. (2023). K-Means Clustering with Natural Density Peaks for Discovering Arbitrary-Shaped Clusters. IEEE Transactions on Neural Networks and Learning Systems.
[36] Kathiresan, V., & Sumathi, P. (2012, January). An efficient clustering algorithm based on Z-score ranking method. In 2012 International Conference on Computer Communication and Informatics (pp. 1-4). IEEE.
[37] M. Sakthi and A. S. Thanamani. (2011). An effective determination of initial centroids in K-means clustering using kernel PCA. International Journal of Computer Science and Information Technologies, vol. 2, no. 3, pp. 955–959.
[38] Khorasani, F., Zanjireh, M. M., Bahaghighat, M., & Xin, Q. (2022). A Tradeoff Between Accuracy and Speed for K-Means Seed Determination. Comput. Syst. Sci. Eng., 40(3), 1085-1098.

## پیوســـت ۱: نتایج آزمایش‌های انجام شده بر روی مجموعه داده‌های سنتزی و واقعی

**میانگین کل امتیاز شاخص رند اصلاح شده و مجموعه کل خطاهای تعداد خوشه برای کل مجموعه داده‌ها**

**نتایج آزمایش‌های انجام شده روی مجموعه داده‌های سنتزی**

| مجموعه داده | آزمایش ۱ | | | آزمایش ۲ | | | آزمایش ۳ | | | آزمایش ۴ | | | آزمایش ۵ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | تعداد خوشه | خطای تعداد خوشه | امتیاز شاخص رند | تعداد خوشه | خطای تعداد خوشه | امتیاز شاخص رند | تعداد خوشه | خطای تعداد خوشه | امتیاز شاخص رند | تعداد خوشه | خطای تعداد خوشه | امتیاز شاخص رند | تعداد خوشه | خطای تعداد خوشه | امتیاز شاخص رند |
| S_100_20_3 | ۳۱ | ۱۱ | ۰,۸۵ | ۲۲ | ۲ | ۰,۹۱ | ۲۲ | ۲ | ۰,۹۱ | ۱۳ | ۷ | ۰,۹۸ | ۱۳ | ۷ | ۰,۶۴ |
| S_100_20_6 | ۲۱ | ۱ | ۰,۸۵ | ۲۰ | ۰ | ۱,۰ | ۲۰ | ۰ | ۱,۰ | ۱۱ | ۹ | ۱,۰ | ۱۱ | ۹ | ۰,۴۹ |
| S_100_5_3 | ۶ | ۱ | ۰,۹۳ | ۵ | ۰ | ۱,۰ | ۵ | ۰ | ۱,۰ | ۵ | ۰ | ۱,۰ | ۵ | ۰ | ۱,۰ |
| S_100_5_6 | ۹ | ۴ | ۰,۷۸ | ۵ | ۰ | ۱,۰ | ۵ | ۰ | ۱,۰ | ۵ | ۰ | ۱,۰ | ۵ | ۰ | ۱,۰ |
| S_200_20_3 | ۲۴ | ۴ | ۰,۸۸ | ۲۷ | ۷ | ۰,۹۳ | ۲۷ | ۷ | ۰,۹ | ۱۳ | ۷ | ۰,۹۷ | ۱۳ | ۷ | ۰,۶۴ |
| S_200_20_6 | ۳۱ | ۱۱ | ۰,۹۸ | ۲۴ | ۴ | ۰,۹۸ | ۲۴ | ۴ | ۰,۹۷ | ۱۶ | ۴ | ۱,۰ | ۱۶ | ۴ | ۰,۷۲ |
| S_200_5_3 | ۴ | ۱ | ۰,۷۸ | ۵ | ۰ | ۱,۰ | ۵ | ۰ | ۱,۰ | ۵ | ۰ | ۱,۰ | ۵ | ۰ | ۱,۰ |
| S_200_5_6 | ۱۴ | ۹ | ۰,۷۹ | ۵ | ۰ | ۱,۰ | ۵ | ۰ | ۱,۰ | ۵ | ۰ | ۱,۰ | ۵ | ۰ | ۱,۰ |
| خطای کل/ میانگین امتیاز | | ۴۲ | ۰,۸۴ | | ۱۳ | ۰,۹۸ | | ۱۳ | ۰,۹۷ | | ۲۷ | ۰,۹۹ | | ۲۷ | ۰,۸۱ |

نتایج آزمایش‌های انجام شده روی مجموعه داده‌های واقعی

| مجموعه داده | آزمایش ۱ | | | آزمایش ۲ | | | آزمایش ۳ | | | آزمایش ۴ | | | آزمایش ۵ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | تعداد خوشه | خطای تعداد خوشه | امتیاز شاخص رند | تعداد خوشه | خطای تعداد خوشه | امتیاز شاخص رند | تعداد خوشه | خطای تعداد خوشه | امتیاز شاخص رند | تعداد خوشه | خطای تعداد خوشه | امتیاز شاخص رند | تعداد خوشه | خطای تعداد خوشه | امتیاز شاخص رند |
| قطعه بندی تصویر ۱۹ ویژگی و ۷ خوشه | ۱۱ | ۴ | ۰,۵۳ | ۸ | ۱ | ۰,۵۸ | ۸ | ۱ | ۰,۶ | ۸ | ۱ | ۰,۴۵ | ۸ | ۱ | ۰,۵ |
| مین و سنگ ۶۰ ویژگی و ۲ خوشه | ۳ | ۱ | ۰,۰۸ | ۴ | ۲ | ۰,۷۲ | ۴ | ۲ | ۰,۶۳ | ۲ | ۰ | ۰,۸۶ | ۲ | ۰ | ۰,۵۵ |
| خطای کل/ میانگین امتیاز | | ۵ | ۰,۳ | | ۳ | ۰,۶۵ | | ۳ | ۰,۶۱ | | ۱ | ۰,۶۶ | | ۱ | ۰,۵۳ |

| مجموعه داده | آزمایش ۱ | | آزمایش ۲ | | آزمایش ۳ | | آزمایش ۴ | | آزمایش ۵ | |
|---|---|---|---|---|---|---|---|---|---|---|
| | خطای تعداد خوشه | امتیاز شاخص رند | خطای تعداد خوشه | امتیاز شاخص رند | خطای تعداد خوشه | امتیاز شاخص رند | خطای تعداد خوشه | امتیاز شاخص رند | خطای تعداد خوشه | امتیاز شاخص رند |
| خطای کل/ میانگین امتیاز | ۴۷ | ۰,۷۴ | ۱۶ | ۰,۹۱ | ۱۶ | ۰,۹۰ | ۲۸ | ۰,۹۳ | ۲۸ | ۰,۷۵ |

## پیوست۲: روندنمای روش پیشنهادی