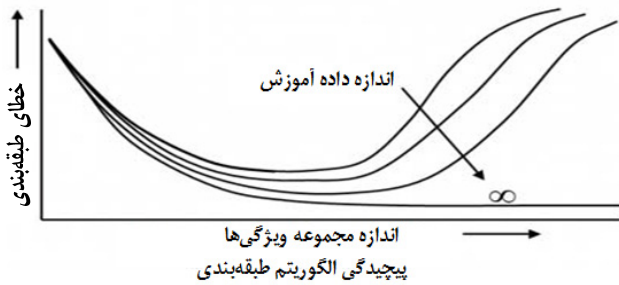


# کاهش ابعاد ویژگی مبتنی بر تئوری راف با استفاده از الگوریتم رقابت لیگ فوتبال

مجید عبدالرزاق نژاد و علی ادیبیان



شکل ۱: پدیده پیکینگ؛ با افزایش تعداد ویژگی‌ها از یک نقطه به بعد خطای طبقه‌بندی افزایش می‌یابد [۱].

اولین و جدی‌ترین چالش پیش روی استخراج دانش پنهان این داده‌ها می‌باشد. به منظور عبور از این چالش، کاهش ابعاد داده تحت عناوین انتخاب ویژگی<sup>۱</sup> و کاهش ویژگی<sup>۲</sup> در دستور کار محققان زیادی قرار گرفته است. تحقیقات نشان می‌دهد که کاهش ابعاد داده با پدیده‌ای به نام پدیده پیکینگ<sup>۳</sup> (شکل ۱) روبه‌رو است که از دو جهت قابل توجه می‌باشد: اول بالابودن حجم محاسبات و افزایش پیچیدگی الگوریتم‌های طراحی شده به منظور استخراج دانش از داده‌های ابعاد بزرگ و دوم کاهش کیفیت دانش استخراج شده از این داده‌ها است [۱]. بنابراین یکی از ضروری‌ترین اقدامات به منظور مرتفع‌نمودن این چالش، کاهش ویژگی می‌باشد که در آن بر اساس یک فرایند انتخاب و کشف زیرمجموعه‌ای حداقلی از ویژگی‌های مجموعه اصلی با حداقل از دست دادن اطلاعات و حفظ یکپارچگی بین اطلاعات اصلی از بین  $2^n$  زیرمجموعه موجود از  $n$  ویژگی داده صورت می‌پذیرد.

از این فرایند تحت عنوان مسئله کاهش ویژگی یاد شده و ایده‌های متعددی در این زمینه ارائه شده که تئوری مجموعه راف<sup>۴</sup> (RST) [۲] یکی از کارآمدترین تئوری‌های موجود برای این مسئله می‌باشد. این تئوری به دو دسته کلاسیک [۳] برای داده‌های صحیح‌مقدار و کاراکترمقدار و فازی [۴] برای داده‌های حقیقی‌مقدار تقسیم می‌شود. در این تئوری با تفکیک ویژگی‌ها به ویژگی‌های شرطی و تصمیم‌گیری، به کمک تئوری نظریه مجموعه‌ها و رویکردهای ریاضی اقدام به شناسایی رابطه و وابستگی بین این دو گروه ویژگی‌ها می‌نمایند. وظیفه RST در مسئله کاهش ویژگی، محاسبه ضریب وابستگی زیرمجموعه ویژگی انتخاب‌شده از داده اصلی به عنوان ورودی RST با ویژگی‌های شرطی به عنوان اساس کنترل دانش می‌باشد. اما انتخاب زیرمجموعه ورودی این

چکیده: افزایش حجم پایگاه داده‌ها، مسئله کاهش ویژگی را ایجاد و پراهمیت می‌سازد. از این رو در علم داده‌کاوی، یکی از زیربخش‌های مهم، مسئله کاهش ابعاد (انتخاب ویژگی) می‌باشد که در آن زیرمجموعه‌ای از ویژگی‌هایی که بیشترین تأثیر را بر روی الگوهای پنهان داده اصلی دارند انتخاب می‌شوند. در سال‌های اخیر، تئوری مجموعه راف به عنوان یکی از ابزارهای موثر و کارآمد در کاهش ویژگی مورد توجه محققان قرار گرفته است. در این مقاله الگوریتم رقابت لیگ فوتبال برای اولین بار به منظور حل این مسئله مورد استفاده قرار گرفته شده است. توانایی فرار از بهین‌های موضعی، امکان بهره‌گیری از اطلاعات بازیکنان توزیع شده در فضای جستجو، همگرایی سریع به جواب‌های بهین و پارامترهای کم این الگوریتم، انگیزه این انتخاب بود. علاوه بر این نوآوری، ارائه تغییراتی همچون بهره‌گیری از مجموع توان بازیکنان ثابت و ذخیره در محاسبه توان هر تیم، در نظر گرفتن ترکیب ساختارهای پیوسته و گسسته برای بازیکن، ارائه یک مکانیزم ابتکاری گسسته‌سازی جدید، ارائه تحلیل هیدرولیک متناسب با مسئله تحقیق برای فرایند برازش هر بازیکن، ارائه اصلاح در عملگرهای تقلید و تهییج بر اساس چالش‌های نسخه اصلی را می‌توان به عنوان ایده‌های پیشنهادی به منظور اصلاح و سازگاری این الگوریتم برای مسئله کاهش ویژگی برشمرد. ایده‌های پیشنهادی بر روی سه سطح داده در ابعاد کوچک، متوسط و بزرگ به همراه چهار الگوریتم فرابتنکاری مشهور ژنتیک و بهینه‌سازی ازدحام ذرات، سیستم ایمنی مصنوعی و لیگ قهرمانان پیاده‌سازی گردیده و مقایسه نتایج به دست آمده نشان از مزیت‌های رقابتی قابل توجه الگوریتم رقابت لیگ فوتبال با سایر الگوریتم‌ها دارد.

کلیدواژه: داده‌کاوی، کاهش ویژگی، انتخاب ویژگی، تئوری مجموعه راف، الگوریتم رقابت لیگ فوتبال، طبقه‌بندی بیزین.

## ۱- مقدمه

با پیشرفت روزافزون تکنولوژی فناوری اطلاعات، نفوذ آن در لایه‌های مختلف زندگی انسان و توسعه فناوری‌های ذخیره‌سازی داده‌ها، بشر با بانک‌های متعدد و بسیار بزرگ اطلاعاتی که حاوی انواع مختلفی از داده‌ها می‌باشند، روبه‌رو شده است.

دانش پنهان درون این بانک‌های اطلاعاتی از اهمیت بسیار زیادی برای مراجع تولیدکننده آن داده برخوردار می‌باشد. اما ابعاد بزرگ داده‌ها،

این مقاله در تاریخ ۲۸ خرداد ماه ۱۳۹۸ دریافت و در تاریخ ۱۲ خرداد ماه ۱۳۹۹ بازنگری شد.

مجید عبدالرزاق نژاد (نویسنده مسئول)، دانشکده مهندسی فنی و مهندسی، دانشگاه بزرگمهر قانات، قان، ایران، (email: abdolrazzagh@buqaen.ac.ir).  
علی ادیبیان، دانشکده مهندسی فنی و مهندسی، دانشگاه بزرگمهر قانات، قان، ایران، (email: dallakeh@buqaen.ac.ir).

1. Feature Selection
2. Attribute Reduction
3. The Peaking Phenomenon
4. Rough Set Theory

کرده و جین و یوان [۹] نیز به ساختار الگوریتم ژنتیک یک جستجوی محلی اضافه نمودند. آتیس و همکاران [۱۱] نیز از ترکیب دوگانه الگوریتم ژنتیک و تئوری راف برای انتخاب ویژگی افزایشی از تابع ارزیابی تحقیقات گذشته [۶] تا [۹] استفاده نمودند که چالش جستجوی محلی و سرعت عملگرایی پایین را به همراه داشت. عبدالرزاق نژاد و همکاران [۳] با ارائه یک تابع برازش دوگانه، یک نسخه اصلاح شده از الگوریتم تکاملی فرهنگی<sup>۳</sup> (CA) را برای این مسئله مورد استفاده قرار دادند. با توجه به ساختار دوجمعی الگوریتم فرهنگی، آنها حساسیت این الگوریتم را نسبت به تعداد مختلف جمعیت مورد ارزیابی قرار دادند.

اولین اقدام در بهره‌گیری از الگوریتم‌های فراابتکاری مبتنی بر هوش تجمعی ذرات اختصاص به جنسن و شن [۱۲] دارد که در سال ۲۰۰۳ از الگوریتم بهینه‌سازی کلونی مورچگان<sup>۴</sup> (ACO) برای حل این مسئله استفاده نمودند. آنها رویکردی جدید از راف ارائه کردند که اصلاحیه مبتنی بر آنتروپی بود. در [۱۳] یک مکانیزم به منظور به روز رسانی فرمون مسیر بین دو ویژگی بر اساس بهترین جواب یافته ارائه شد و در [۱۴] یک فرمول احتمالی انتقال حالت جدید و نیز یک فرمول به منظور مدیریت افزایش فرمون ارائه گردید. حتی برای پرهیز از شروع‌های تصادفی در [۱۵]، یک ساختار هسته ویژگی تعریف شد و در [۱۶] با ترکیب این الگوریتم با شبکه عصبی تلاش شد از رویکردهای بسته‌بندی و فیلترینگ به صورت هم‌زمان استفاده شود و یک نظام هدایت صحیح مورچگان تعریف شد. اما نسخه‌های توسعه‌یافته کلونی مورچگان در حل مسئله کاهش ویژگی [۱۳] تا [۱۶] نتوانست چالش تصمیمات تصادفی متوالی و لزوم مدیریت این تصمیمات متوالی به منظور رسیدن به جواب‌های خوب را برطرف سازد.

در کلاس الگوریتم‌های هوش تجمعی ذرات، وانگ و همکارانش [۱۷] از الگوریتم ازدحام ذرات<sup>۵</sup> (PSO) برای حل مسئله کاهش ویژگی مبتنی بر تئوری راف بهره بردند. در این مقاله یک زیرمجموعه از ویژگی‌ها به عنوان موقعیت یک ذره در نظر گرفته شد و جمعیت زیادی از این ذرات با هدف رسیدن به بهترین موقعیت (بهترین زیرمجموعه) اقدام به پیمایش فضای جستجو و حرکت به سمت هدف نمودند. تغییر موقعیت ذرات در این الگوریتم پیوسته مقدار بوده و تعیین ضرایب حرکت در امتداد بهترین موقعیت‌های محلی و سراسری نقش کلیدی در کیفیت جستجوی این الگوریتم داشتند. لذا نسخه‌های باینری الگوریتم PSO [۱۸] و [۱۹] و نسخه گسسته آن [۲۰] برای حل این مسئله ارائه گردید و مشکل تعیین ضرایب حرکت در جای خود باقی ماند.

الگوریتم کلونی زنبور عسل مصنوعی<sup>۶</sup> (ABC) یکی دیگر از الگوریتم‌های هوش تجمعی ذرات است که از رفتار جستجوگرانه زنبوران عسل الهام گرفته است. یکی از مزیت‌های این الگوریتم پیاده‌سازی آسان و داشتن حداقل پارامترهای کنترلی می‌باشد. این الگوریتم را به عنوان یک ابزار قدرتمند بهینه‌سازی می‌شناسند که توان نسبتاً خوبی در حل مسئله کاهش ویژگی از خود نشان داده است [۲۱]. اما این الگوریتم نیز دارای رفتار پیوسته می‌باشد و لذا نسخه گسسته [۲۲] و باینری آن [۲۳] با اعمال تغییراتی در ساختار جستجوی محلی و شناسایی منابع غذا به منظور حل مسئله کاهش ویژگی بر اساس تئوری راف ارائه گردیده است. اگرچه

تئوری، با توجه به رشد نمایی تعداد زیرمجموعه ویژگی‌ها ( $2^n$ ) آن را به یک چالش NP-Hard تبدیل کرده است. برای برطرف نمودن این چالش نیز از ظرفیت الگوریتم‌های فراابتکاری استفاده شده است. این دسته از الگوریتم‌های بهینه‌سازی بر اساس ایده‌های هوشمند طبیعی الهام گرفته شده، دارای پتانسیل‌های گوناگونی در فرار از بهین‌های موضعی، امکان بهره‌گیری از اطلاعات ذرات توزیع شده در فضای جستجو، همگرایی سریع به جواب‌های بهین، حافظه سوابق عملکرد و پویایی می‌باشند.

در این مقاله و با توجه به چالش‌های الگوریتم‌های بهینه‌سازی فراابتکاری مرور شده در بخش بعدی که برای مسئله کاهش ویژگی بر اساس تئوری مجموعه راف تا کنون مورد استفاده قرار گرفته شده‌اند، برای اولین بار الگوریتم رقابت لیگ فوتبال<sup>۱</sup> (SLC) با ماهیتی پیوسته رفتار برای حل مسئله کاهش ویژگی گسسته رفتار سازگار و مورد استفاده قرار گرفته شده است. همچنین در این الگوریتم به منظور دستیابی به کوچکترین زیرمجموعه کاهش با بالاترین میزان دقت نتایج، یک تابع ارزیابی جدید طراحی شده است. الگوریتم سازگار شده پیشنهادی بر روی سه گروه داده در اندازه‌های کوچک، متوسط و بزرگ پیاده‌سازی شده و صحت طبقه‌بندی زیرمجموعه‌های به دست آمده توسط طبقه‌بندی Naive Bayes با صحت طبقه‌بندی داده‌های اصلی محاسبه و مقایسه شده است. مقایسه عملکرد الگوریتم SLC با الگوریتم‌های معروف فراابتکاری و تحلیل نتایج به دست آمده گویای برتری و مزیت نسبی این الگوریتم در حل مسئله کاهش ویژگی به خصوص برای داده‌های ابعاد بزرگ نسبت به سایر الگوریتم‌ها می‌باشد.

در ادامه این مقاله، در بخش دوم به بررسی و تحلیل الگوریتم‌های فراابتکاری به کار گرفته شده برای مسئله کاهش ویژگی بر اساس تئوری مجموعه راف پرداخته شده و مزایا و معایب این الگوریتم‌ها را مستخرج نموده‌ایم. نحوه عملکرد تئوری مجموعه راف به عنوان رکن اصلی مسئله تحقیق در بخش سوم تشریح شده است. در بخش چهارم، ساختار الگوریتم رقابت لیگ فوتبال، تابع ارزیابی آن و نحوه سازگاری این الگوریتم برای مسئله تحقیق مطرح شده است. نتایج پیاده‌سازی ایده پیشنهادی، مقایسه و تحلیل این نتایج در بخش پنجم ارائه شده و نهایتاً در بخش ششم، خلاصه و دستاوردهای این تحقیق تشریح شده است.

## ۲- پیشینه پژوهش

تا کنون الگوریتم‌های متعددی به منظور کاهش حداقلی ویژگی‌ها ارائه شده است. نخست، دای و همکاران [۵] در سال ۲۰۰۲ الگوریتم ژنتیک را به منظور کمینه‌سازی کاهش مبتنی بر تئوری راف پیشنهاد دادند. الگوریتم ژنتیک، یک الگوریتم گسسته‌رفتار می‌باشد که بیشترین سازگاری با مسئله کاهش ویژگی را داراست ولی به سبب ضعف در جستجوی همسایگی، نداشتن حافظه و متأثر بودن از رفتارهای تصادفی زیاد باعث شد نسخه‌های دیگری نیز از این الگوریتم برای حل مسئله کاهش ویژگی ارائه گردد [۶] تا [۹]. از منظر ژو و همکاران [۱۰] چالش تحقیقات قبلی، تابع برازش کروموزوم بود که عملکرد خوبی در محاسبه کمینه‌سازی کاهش ویژگی نداشت. آنها برای حل این چالش، یک تابع برازش جدید تعریف کردند که سرعت همگرایی را افزایش می‌داد. همچنین به منظور مرتفع نمودن چالش جستجوی ضعیف همسایگی، سالوانی و جدی [۸] نسخه دوگانه‌شده الگوریتم ژنتیک و الگوریتم سیل بزرگ<sup>۲</sup> (GD) را ارائه

3. Cultural Algorithm  
4. Ant Colony Optimization  
5. Particle Swarm Optimization  
6. Artificial Bee Colony Algorithm

1. Soccer League Competition Algorithm  
2. Great Deluge Algorithm

داشتن توانایی فرار از بهین‌های موضعی، امکان جستجوی سراسری با کمترین پارامترهای تنظیم‌کننده بر اساس حداقل تصمیمات تصادفی هدف این تحقیق تعریف شده است. الگوریتمی که ویژگی‌های اشاره‌شده را برای حل مسئله کاهش ویژگی دارا باشد خواهد توانست رفتاری بهتر از خود بر روی داده‌های بزرگ نسبت به الگوریتم‌های فراابتکاری مشهور بررسی‌شده در این بخش داشته باشد.

### ۳- تئوری مجموعه راف (RST)

پالوکا در سال ۱۹۹۱ تئوری مجموعه راف  $(RST)$  را به عنوان یک ابزار ریاضی برای پیدا کردن زیرمجموعه‌های حداقلی از یک مجموعه بزرگ به گونه‌ای که ماهیت و ارتباطات بین عناصر مجموعه اصلی در زیرمجموعه حداقلی حفظ شده باشد، ارائه کرد [۲]. اساس RST مبتنی بر اطلاعات نهفته در ویژگی‌های شرط و انطباق آن با اطلاعات موجود در ویژگی تصمیم می‌باشد. اطلاعات هر ویژگی از ویژگی‌های شرط را که منطبق بر اطلاعات ویژگی تصمیم باشد اطلاعات مفید خطاب کرده و در کاهش ویژگی بر اساس تئوری راف به دنبال کوچک‌ترین زیرمجموعه از ویژگی‌های شرط هستیم که بیشترین اطلاعات مفید را دارا هستند [۳۲]. به بیان دیگر، فرض کنیم یک سیستم اطلاعاتی همچون  $S = (U, A)$  که در آن  $U$  مجموعه اشیا و  $A$  مجموعه ویژگی‌ها این سیستم تقسیم شده بر دو گروه ویژگی‌های شرط  $C$  و ویژگی تصمیم  $D$  یعنی  $C \cap D = \emptyset$  و  $C \cup D = A$  مفروض باشد. تعداد کل موجودیت‌ها برابر  $n$  و تعداد ویژگی‌های شرط برابر  $m$  است. اگر دو موجودیت  $x$  و  $y$  عضو  $U$  باشند، رابطه تفکیک‌ناپذیری روی مجموعه ویژگی‌های شرط  $C$  به صورت زیر تعریف می‌شود

$$IND(C) = \{(x, y) \in U * U \mid \forall Attr \in C, Attr(x) = Attr(y)\} \quad (1)$$

همان طور که مشاهده می‌شود در یک رابطه تفکیک‌ناپذیری، مجموعه‌ای از ویژگی‌ها (ستون‌ها) همچون  $C$  و مجموعه‌ای از موجودیت‌ها (سطرها) همچون  $x$  و  $y$  مفروض هستند به گونه‌ای که مقادیر ویژگی‌های  $C$  این موجودیت‌ها با یکدیگر متناظراً برابر باشد. در صورتی که مجموعه موجودیت  $(x, y) \in U$  به کل مجموعه مرجع  $U$  تعمیم یابد، رابطه تفکیک‌ناپذیری  $IND(C)$  تشکیل افزای هم‌ارزی  $U/IND(C)$  بر اساس مجموعه ویژگی  $C$  را خواهد داد که با  $[X]_C$  نمایش می‌دهند. فرض کنیم افزای هم‌ارزی  $[X]_C$  برابر با  $X_1, \dots, X_n$  که  $N \leq n$  باشد. این بدان معنی است که کل موجودیت‌های  $U$  به  $N$  افزای به گونه‌ای تقسیم شده‌اند که مقادیر ویژگی‌های  $C$  هر افزای با یکدیگر برابر هستند.

با هر مجموعه در RST، یک جفت مجموعه به نام‌های تقریب بالا و پایین همراه می‌شود. تقریب پایین شامل همه موجودیت‌هایی است که عضو کلاس هم‌ارزی  $C$  بوده و مطمئناً به مجموعه مربوط نیز متعلق باشد. تقریب بالا شامل همه موجودیت‌هایی است که شاید به مجموعه متعلق باشند و اختلاف بین تقریب بالا و پایین را ناحیه مرز می‌نامند. لذا با فرض  $X \subseteq U$  و  $Q \subseteq C$  زیرمجموعه‌ای از ویژگی‌ها، این تقریب‌ها را می‌توان به صورت زیر بازتعریف نمود

$$\underline{Q}X = \{x \mid [x]_Q \subseteq X\} \quad (2)$$

چالش ضعف در جستجوی محلی این الگوریتم در نسخه [۲۳] برطرف گردید ولی چالش همگرایی زودرس برای آن همچنان برای حل مسئله کاهش ویژگی بر اساس تئوری راف باقی ماند. از دیگر الگوریتم‌های فراابتکاری هوش تجمعی ذرات که برای حل این مسئله پیاده‌سازی و اصلاح شده‌اند می‌توان به الگوریتم کرم شب‌تاب  $(FF)$  [۲۴]، الگوریتم ازدحام ماهی‌ها  $(AFS)$  [۲۵]، الگوریتم خفاش  $(BA)$  [۲۶]، الگوریتم قورباغه جهنده  $(SFL)$  [۲۷]، الگوریتم بهینه‌سازی اجتماع عنکبوت‌ها  $(SSOA)$  [۲۸]، الگوریتم بهینه‌سازی شعله و پروانه  $(MFOA)$  [۲۹] و الگوریتم جستجوی کلاغ  $(CSA)$  [۳۰] اشاره کرد که در این الگوریتم‌ها نیز چالش پیوسته رفتار بودن، ضعف در جستجوهای محلی (به غیر از SFL) و حساسیت به بهین‌های محلی را دارا می‌باشند.

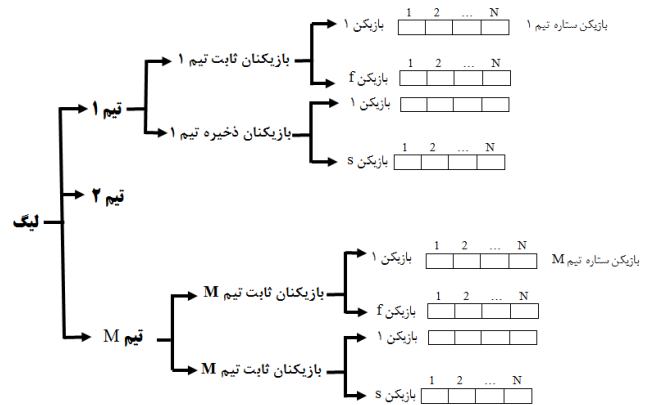
سالوانی عبدالله و همکاران [۳۱] الگوریتم انجماد تدریجی  $(SA)$  را به عنوان یکی از الگوریتم‌های الهام گرفته شده از علوم فیزیک مورد بازنگری قرار دادند و نسخه اصلاح‌شده آن را به منظور حل مسئله کاهش ویژگی استفاده کردند. این الگوریتم بر خلاف الگوریتم‌های هوش تجمعی ذرات، جمعیت محور نبوده و بر اساس تک‌نقطه محوری عمل می‌نماید. با توجه به ساختار آن، این الگوریتم با چالش‌هایی همچون عدم حافظه و امکان بهره‌گیری از اطلاعات سایر نقاط جستجو و نیز وابستگی الگوریتم به جدول دمایی به گونه‌ای که با افزایش بیش از حد دما، سرعت همگرایی کاهش یافته و با کاهش دما در تله بهینه محلی می‌افتد، روبه‌رو است. الگوریتم الکترومگنتیک  $(EM)$  یکی دیگر از الگوریتم‌های الهام گرفته شده از علوم فیزیک می‌باشد که در نسخه [۳۲] حافظه‌مند و گسسته‌سازی شد تا ضمن پشت سر گذاشتن چالش‌های SA توانایی جستجوی جامع‌تری برای مسئله کاهش ویژگی داشته باشد. اما این الگوریتم نیز با توجه به ساختار خود با چالش تغییرات بسیار کوچک در جواب‌های جدید با توجه به کیفیت ارزیابی جواب‌های ماقبل خود روبه‌رو است.

محققان یک روش مبتنی بر حافظه اکتشافی برای حل مسئله کاهش ویژگی در تئوری مجموعه راف بر اساس الگوریتم جستجوگر ممنوعه ارائه کردند و نام آن را جستجوگر ممنوعه کاهش ویژگی  $(TSAR)$  نامیدند [۳۳]. این الگوریتم برای حل مسئله کاهش ویژگی از یک ساختار باینری برای تک‌ذره جستجوگر خود استفاده کرده و نقش تئوری مجموعه راف در این الگوریتم، اندازه‌گیری کیفیت جواب‌های این الگوریتم است. به منظور کاهش ویژگی، TS دو روش را دنبال می‌کند: اولاً برای فرار از بهینه‌های محلی در سراسری حرکت می‌کند و ثانیاً به منظور برنگشتن به جواب‌های بازدیدشده لیستی به نام Tabulist تهیه می‌نماید. اما ضعف در جستجوی سراسر به منظور کشف کوچک‌ترین زیرمجموعه با ضریب راف ۱ را برای داده‌های بزرگ با توجه به تک نقطه محور بودن آن [۳۴]، این الگوریتم را با چالش جدی روبه‌رو کرده است.

با توجه به الگوریتم‌های ارائه‌شده برای حل مسئله کاهش ویژگی بر اساس تئوری مجموعه راف، پیشنهاد الگوریتم جستجوگری که ضمن

1. FireFly Algorithm
2. Artificial Fish Swarm Algorithm
3. Bat Algorithm
4. Shuffled Frog Leaping Algorithm
5. Social Spider Optimization Algorithm
6. Moth Flame Optimization Algorithm
7. Crow Search Algorithm
8. Simulated Annealing Algorithm
9. Electromagnetic\_Like Mechanism Algorithm
10. Tabu Search Attribute Reduction

قهرمانی فوتبال و رقابت بین تیم‌ها و بازیکنان به منظور تبدیل هر تیم به قهرمان لیگ و هر بازیکن به ستاره لیگ در طول  $maxItr$  فصل طراحی شده است. این الگوریتم، یک الگوریتم جمعیت‌محور دوسطحی می‌باشد که در سطح اول تیم‌ها و در سطح دوم بازیکنان تیم‌ها قرار می‌گیرند. هر کدام از دو سطح یادشده فرایند جستجو را به نحو مؤثری در راستای یافتن بهینه سراسری هدایت می‌کنند. ساختار لیگ و ارتباط آن با پاسخ‌های یک مسئله جستجو در الگوریتم رقابت لیگ فوتبال در شکل ۲ نشان داده شده است.



شکل ۲: ساختار لیگ و ارتباط آن با پاسخ‌های یک مسئله جستجو در الگوریتم رقابت لیگ فوتبال.

#### ۴-۱ مفاهیم اولیه نسخه کلاسیک الگوریتم SLC

در الگوریتم SLC هر بازیکن بیانگر یک پاسخ می‌باشد. در هر تیم بازیکنان به دو گروه ثابت (FP) و ذخیره (P) تقسیم می‌شوند که در بین بازیکنان ثابت، بازیکن با بهترین عملکرد (برازش) را بازیکن ستاره (SP) می‌نامند. بازیکنان ثابت هر تیم،  $f$  بازیکنان برتر آن تیم هستند. همچنین برترین بازیکن در یک لیگ، بازیکن فوق ستاره (SSP) نامیده می‌شود. بازیکنان ثابت تحت عملگر تقلید<sup>۴</sup> جهت رسیدن به بازیکنان ستاره و فوق ستاره قرار گرفته و ضعیف‌ترین بازیکن ذخیره نیز تحت عملگر تهیج<sup>۵</sup> در جهت رسیدن به یک بازیکن ثابت تصادفی قرار می‌گیرد. در واقع می‌توان گفت هم‌زمان با برگزاری رقابت میان تیم‌ها، یک جستجو و بهبود عملکرد مؤثر بر روی بازیکنان در حال اجراست.

از طرفی مجموعه‌ای از پاسخ‌ها (بازیکنان) تشکیل تیم را می‌دهند و تیم‌ها در مقابل یکدیگر دو بار بازی (یک بار به عنوان میزبان و یک بار به عنوان مهمان) انجام می‌دهند. تعداد کل مسابقات یک لیگ برابر با  $M \times (M-1) / 2$  که  $M$  تعداد تیم‌ها می‌باشد، است. توان هر تیم که احتمال پیروزی آن تیم در مقابل تیم دیگر است با توجه به مجموع توان (برازش) بازیکنان (پاسخ‌های) ثابت آن تیم محاسبه می‌گردد. اما در این نسخه اصلاح‌شده، استفاده از مجموع توان بازیکنان ثابت و ذخیره پیشنهاد می‌گردد. علت آن ورود اطلاعات بازیکنان ذخیره که با وجود محاسبه تابع برازش در فرایند بهبود سایر بازیکنان نسخه اولیه این الگوریتم هیچ تأثیری نداشته‌اند (به جز ضعیف‌ترین بازیکن ذخیره) می‌باشد. این عملگرها بعد از اتمام هر بازی و مشخص شدن تیم برنده، روی بازیکنان تیم برنده اعمال می‌شوند. در نهایت تیم‌ها بر اساس مجموعه امتیازات کسب‌شده در جدول امتیازها در پایان یک فصل (تکرار الگوریتم) رتبه‌بندی شده و تیم نخست به عنوان قهرمان فصل معرفی و دو تیم پایان جدول با سقوط به دسته پایین‌تر از لیست تیم‌ها حذف شده و دو تیم جدید (که به صورت تصادفی تولید شده‌اند) از لیگ دسته پایین‌تر به جمع تیم‌ها برای شروع فصل بعد اضافه خواهند شد. در شکل ۳، شبه‌کد نسخه استاندارد این الگوریتم نمایش داده شده است.

#### ۴-۲ مراحل الگوریتم اصلاح‌شده SLC برای کاهش ویژگی

نسخه اصلاح‌شده و سازگار شده الگوریتم رقابت لیگ فوتبال به منظور حل مسئله کاهش ویژگی بر اساس تئوری مجموعه راف در پنج مرحله اصلی به شرح ذیل تشریح می‌گردد.

1. Fix & Saved Players (FP) & (P)
2. Star Player
3. Super Star Player
4. Imitation
5. Provocation

$$\bar{Q}X = \{x | [x]_Q \cap X\} \quad (3)$$

در RST، ضریب وابستگی  $K$  بین ویژگی تصمیم  $D$  و زیرمجموعه‌ای از ویژگی‌های شرط مانند  $Q \subseteq C$  به صورت زیر تعریف می‌شود

$$POS_Q(U, D) = \bigcup_{X \in U/Q} \bar{Q}X = \bigcup \{X \in \frac{U}{IND(Q)} : \exists i, X \subseteq Y_i, Y_i \in \frac{U}{IND(Q)}\} \quad (4)$$

$$\gamma_Q = \frac{|POS_Q(U, D)|}{|U|} = \frac{|\bigcup_{X \in U/Q} \bar{Q}X|}{|U|} = \frac{\sum_{i=1}^k |\bar{Q}X_i|}{|U|} \quad (5)$$

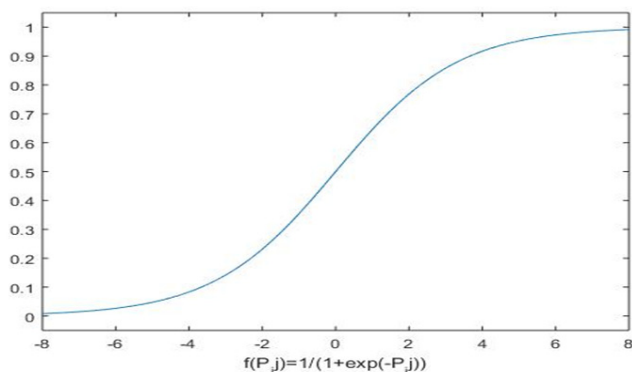
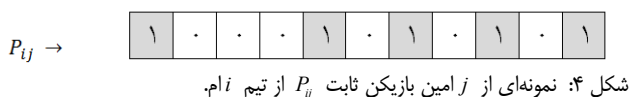
در (۴)، به  $POS_Q(U, D)$  ناحیه مثبت هم‌ارزی از  $U$  گفته می‌شود و شامل تمام موجودیت‌های اشتراکی بین مجموعه کلاس‌های هم‌ارزی  $U/IND(D)$  و  $U/IND(Q)$  می‌باشد. اگر ضریب وابستگی  $\gamma_Q$  برابر با یک باشد، به معنای وابستگی کامل زیرمجموعه ویژگی‌های شرط  $Q$  و ویژگی تصمیم  $D$  خواهد بود. با توجه به تعاریف ذکر شده، می‌توان مسئله کاهش ویژگی بر اساس تئوری مجموعه راف را که هدف آن بیشینه‌نمودن ضریب وابستگی راف و هم‌زمان کمینه‌سازی تعداد ویژگی‌های انتخاب‌شده از ویژگی‌های شرط می‌باشد به صورت زیر فرموله نمود

$$\begin{aligned} \min f_1 &= \frac{|Q|}{m} \\ \max f_2 &= \gamma_Q \\ \text{s. to :} \\ Q &\in \{0, 1\}^m \\ \gamma_Q &= \frac{\sum_{i=1}^k |\bar{Q}X_i|}{|U|} \\ 0 &\leq Q = \sum_{i=1}^m X_i \leq m \end{aligned} \quad (6)$$

که در رابطه فوق،  $f_1$  نرمال شده تعداد ویژگی‌های انتخاب‌گرفته است. به عنوان مثال فرض کنید یک زیرمجموعه شامل ۴۰ ویژگی از ۱۵۰ ویژگی شرط انتخاب‌شده است. مقدار نرمال شده تعداد ویژگی انتخاب‌شده ۰/۲۷ که باید کمینه شود.

#### ۴- الگوریتم رقابت لیگ فوتبال

الگوریتم رقابت لیگ فوتبال (SLC) [۳۵] اولین بار توسط موسویان و رودسری معرفی شد. این الگوریتم با الهام‌گرفتن از ساختار یک لیگ

شکل ۵: نمودار تابع سیگموئید طراحی شده در بازه  $[-۸, ۸]$ .

سپس مقادیر مناظر باینری نسخه گسسته بازیکن  $z$  از تیم  $i$  ام  $DP_{ij}$  به صورت زیر محاسبه می‌شود

$$DP_{ij}(r) = \begin{cases} 1 & \text{if } \text{sig}(p_{ij}(r)) \geq \text{rand} \\ 0 & \text{otherwise} \end{cases} \quad (۹)$$

که در آن  $\text{rand}$  یک عدد تصادفی بین ۰ و ۱ است و در این تحقیق با توجه به بررسی میزان مقادیر عناصر هر بازیکن در اجراهای آزمایشی الگوریتم،  $\text{rand} = ۰,۴$  انتخاب شده است.

**مرحله دوم (محاسبه قدرت بازیکنان و تیم‌ها):** تابع برآزش قدرت بازیکن در نسخه استاندارد الگوریتم SLC یک تابع بیشینه‌سازی تعریف شده و در نسخه اصلاحی در این مقاله با توجه به مسئله کاهش ویژگی (رابطه (۶)) یک تابع کمینه‌سازی تعریف شده است به گونه‌ای که هرچه مقدار تابع قدرت بازیکن کمتر باشد، آن بازیکن قدرتمندتر و هرچه مقدار عددی تابع قدرت هر تیم کمتر، آن تیم قوی‌تر خواهد بود. لذا تابع پیشنهادی در این تحقیق به منظور محاسبه قدرت بازیکن  $PP$  شماره  $j$  از تیم  $i$  به صورت زیر ارائه می‌گردد

$$PP_{ij} = \alpha \times f_1(DP_{ij}) + (1 - \alpha) \times (1 - f_1(DP_{ij})) + CP(DP_{ij}) \quad (۱۰)$$

که در آن  $f_1$  مقدار نرمال شده تعداد ویژگی‌های انتخاب شده،  $f_2$  مقدار ضریب وابستگی تئوری راف بر اساس (۶)،  $CP$  مقدار جریمه ناشی از انتخاب زیرمجموعه ویژگی با ضریب وابستگی راف غیر از ۱ (آن را تحلیل هیدرولیک الگوریتم نیز می‌نامند) و  $\alpha$  یک عدد تصادفی بین ۰ و ۱ می‌باشد. با توجه به این که مقادیر  $f_1$  و  $f_2$  بین بازه  $[۰, ۱]$  تغییر کرده و هدف کمینه‌سازی  $f_1$  و بیشینه‌سازی  $f_2$  می‌باشد، در (۱۰)، ترکیب خطی  $f_1$  و  $1 - f_2$  همراه با یک تابع جریمه به عنوان مقدار قدرت بازیکن در نظر گرفته شده است. با توجه به اهمیت بیشتر تابع  $f_1$  و مکان اعمال جریمه بابت عدم تحقق ضریب راف ۱ توسط تحلیل هیدرولیک این الگوریتم، مقدار  $\alpha = ۰,۷$  انتخاب شده است. همچنین مقدار تحلیل هیدرولیک  $CP$  برای این نسخه از الگوریتم به صورت ذیل محاسبه می‌گردد

$$CP(DP_{ij}) = \sqrt{1 - f_2(DP_{ij})} \quad (۱۱)$$

```

Input : Populationsize, Problemsize, M, F, S
Output : Sbest

1. Pop ← ∅ ;
2. League Scoreboard ← 0 ;
3. for i = 1 to populationsize do
4.   Pi position ← Random Position(Problemsize);
5.   Pop ← Pi ;
6. end
7. Evaluate Population(Pop);
8. League Teams ← Make League Teams(Pop, M, F, S);
9. while ~ Stop Condition() do
10.  League Scoreboard(League Teams) ← 0 ;
11.  SSP ← Get Best Solution(League Teams);
12.  for each Teami ∈ League Teams do
13.    for each Teamn ∈ League Teams/Teami do
14.      Teamwinner ← Do Match(Teami, Teamn);
15.      League Scoreboard(Teamwinner) ← ++1;
16.      SP ← Get Best Solution(Teamwinner);
17.      for each FPj ∈ Teamwinner do
18.        FPj ← Imitation(FPj, SSP, SP, problemsize);
19.      end
20.      S ← Get Worst Solution(Teamwinner);
21.      G ← Average Value of Fix Players(Teamwinner);
22.      S ← Provocation(S, G, problemsize);
23.      Update Fix Players (Teamwinner);
24.    end
25.  end
26.  League Teams ← Sort Teams(League Teams, League Scoreboard);
27.  League Teams ← Relegation & Promotion(League Teams);
28. end
29. Sbest ← Get best Solution (Pop);
30. Return Sbest ;

```

شکل ۳: شبه‌کد الگوریتم استاندارد رقابت لیگ فوتبال [۳۵].

**مرحله اول (تشکیل تیم‌های لیگ):** در این مرحله عملیات مقداردهی اولیه به پارامترها و تولید بازیکنان تیم‌های فصل اول لیگ صورت می‌گیرد. چون الگوریتم SLC یک الگوریتم بهینه‌سازی پیوسته می‌باشد، لذا هر بازیکن اعم از ثابت  $z$  از تیم  $i$  ام ( $P_{ij}$ ) یا ذخیره  $z$  ام از تیم  $i$  ام ( $S_{ij}$ ) یک آرایه  $m$  بعدی حقیقی مقدار که  $m$  تعداد ویژگی‌های شرط است، به صورت زیر مقداردهی اولیه می‌شود

$$P_{ij}(r) = X_{\min}(r) + \text{rand}(0, 1) \times (X_{\max}(r) - X_{\min}(r)) \quad (۷)$$

در این رابطه  $X_{\max}(r) = \lambda$  و  $X_{\min}(r) = -\lambda$  و  $r = 1, \dots, M$  به ترتیب بیانگر حداقل و حداکثر مقدار ممکن در بعد  $r$  ام می‌باشند. از طرفی مسئله کاهش ویژگی به عنوان یک مسئله گسسته مقدار به دنبال یافتن کوچک‌ترین زیرمجموعه از ویژگی‌های شرط (نگاه شود به (۶)) است، لذا می‌بایست نسخه گسسته هر بازیکن  $DP_{ij}$  به صورت یک آرایه  $m$  بعدی باینری مناظر با نسخه پیوسته آن  $P_{ij}$  به گونه‌ای تعریف شود که ویژگی‌های شرطی مناظر با خانه‌های صفر در زیرمجموعه انتخابی حضور نداشته و ویژگی‌های شرطی مناظر با خانه‌های یک در این زیرمجموعه حضور خواهند داشت. فرض کنید مجموعه ویژگی‌های تصمیم شامل ۱۱ ویژگی باشند، لذا یک آرایه باینری به طول ۱۱ که در شکل ۴ نمایش داده شده، نمونه‌ای از  $z$  زمین بازیکن ثابت  $P_{ij}$  از تیم  $i$  ام می‌باشد که نقطه معادل آن در فضای جواب زیرمجموعه ویژگی‌های تصمیم  $\{1, 5, 7, 9, 11\}$  خواهد بود.

برای تحقق این مهم که آن را گسسته‌سازی الگوریتم به منظور سازگاری با مسئله کاهش ویژگی می‌گویند، عناصر حقیقی مقدار هر بازیکن  $P_{ij} = (p_{ij}(1), \dots, p_{ij}(m))$  که بر بازه  $[-\lambda, \lambda]$  تعریف شده را تحت تأثیر تابع سیگموئید که نمودار آن در شکل ۵ نمایش داده شده و به صورت ذیل تعریف می‌شود، قرار می‌دهیم

$$\text{sig}(p_{ij}(r)) = \frac{1}{1 + \exp(-p_{ij}(r))} \quad (۸)$$

می‌شود و در غیر این صورت بازیکن جدید بر اساس (۱۶) از بازیکن ستاره تیم خود تقلید کرده و در صورت بهبود توان آن جایگزین حالت اولیه خواهد شد. در غیر این صورت بازیکن جدید بر اساس (۱۷) از یک بازیکن تصادفی از تیم خود تقلید کرده و در صورت بهبود توان عمل جایگزینی بازیکن جدید صورت خواهد گرفت و اگر این تقلید نیز پاسخ نداد، بازیکن  $i$  از تیم برنده بدون تغییر از عملگر تقلید خارج خواهد شد.

**عملگر تهییج:** این عملگر بر روی ضعیف‌ترین بازیکن ذخیره تیم برنده ( $P_{i,worst}$ ) و با هدف بهبود آن بازیکن و تبدیل شدن به یک بازیکن ثابت انجام می‌شود و بر اساس روابط ذیل اعمال می‌گردد

$$P_{i,worst} = P_{i,worst} + rand(0.9, 1) \times (G_i - P_{i,worst}) \quad (18)$$

$$P_{i,worst} = P_{i,worst} + rand(0.4, 0.6) \times (P_{i,worst} - G_i) \quad (19)$$

که در این روابط  $G_i$  متوسط مقادیر بازیکنان تیم  $i$  ام می‌باشد و به صورت زیر محاسبه می‌شود

$$G_i(r) = \frac{1}{f+s} \sum_{j=1}^{f+s} P_{ij}(r) \quad , \quad r = 1, \dots, M \quad (20)$$

در صورتی که توان ضعیف‌ترین بازیکن تیم برنده بر اساس (۱۸) بهبود یافت، ساختار جدید این بازیکن جایگزین خواهد شد و در غیر این صورت (۱۹) اجرا و توان این بازیکن تغییر یافته محاسبه می‌گردد. در صورت بهبود توان، ساختار جدید جایگزین شده و در غیر این صورت یک بازیکن جدید به صورت تصادفی تولید و جایگزین خواهد شد. با تکمیل مسابقات و اعمال عملگرهای تقلید و تهییج، تیم‌ها بر اساس جدول امتیازات رتبه‌بندی می‌شوند.

**مرحله چهارم (صعود و سقوط تیم‌ها در پایان فصل):** در این مرحله دو تیم که در پایین‌ترین رده جدول حضور دارند از لیگ حذف شده و به دسته پایین‌تر سقوط می‌کنند و به جای آنها دو تیم جدید که بازیکنان آنها به صورت تصادفی تولید شده‌اند (اجرای مراحل اول و دوم) از دسته پایین‌تر به لیگ صعود می‌نمایند.

**مرحله پنجم (کنترل شرط خاتمه):** شرط خاتمه، رسیدن به حداکثر تعداد فصول تعریف شده یعنی  $MaxItr$  است. مرحله چهارم تا رسیدن تعداد فصول به شرط خاتمه تکرار خواهد شد.

## ۵- نتایج و ارزیابی

در این مقاله، پیاده‌سازی الگوریتم پیشنهادی، ارزیابی و مقایسه نتایج به دست آمده را در پنج گام طراحی کرده‌ایم. در گام نخست مشخصات داده‌های محک، نحوه تنظیم الگوریتم‌ها و جزئیات نتایج به دست آمده ارائه و تشریح می‌گردد. در گام دوم، تحلیل حساسیت روش پیشنهادی به پارامترهای ساختاری الگوریتم اصلاح شده لیگ فوتبال مورد بررسی قرار گرفته است. مقایسه سرعت همگرایی در قالب ترسیم نمودارهای همگرایی گام سوم ارزیابی نتایج خواهد بود. اجرای آزمون‌های فرض به منظور تعیین سطح معناداری مقایسه الگوریتم‌های پیاده‌سازی شده و نیز اجرای الگوریتم طبقه‌بندی Naïve Bayes بر روی داده‌های اصلی و بهترین زیرمجموعه کاهش یافته آنها به ترتیب گام‌های چهارم و پنجم این بخش خواهند بود.

### ۵-۱ نمایش نتایج پیاده‌سازی

پیاده‌سازی الگوریتم اصلاح شده رقابت لیگ فوتبال به منظور کاهش ویژگی بر اساس تئوری مجموعه راف توسط Matlab ۲۰۱۷a و آنالیز

که در آن، اگر ضریب وابستگی راف  $DP_{ij}$  برابر ۱ باشد، آن گاه  $CP$  برابر صفر و در غیر این صورت اثر تشدید شده فاصله مقدار ضریب راف تا ۱ به عنوان جریمه به تابع قدرت بازیکن اضافه شده و یا به عبارت دیگر بازیکن را ضعیف‌تر خواهد کرد. همچنین توان تیم  $i$  ام ( $TP_i$ ) که در نسخه استاندارد برابر با متوسط توان بازیکنان ثابت آن تیم تعریف شده است، در این نسخه اصلاح شده برابر با متوسط توان کل بازیکنان آن تیم به شرح ذیل در نظر گرفته شده است

$$TP_i = \frac{1}{f+s} \sum_{j=1}^{f+s} PP_{ij} \quad (12)$$

دلیل این تغییر، ورود اطلاعات (اثر) بازیکنان ذخیره که با وجود محاسبه تابع برآزش در فرایند بهبود سایر بازیکنان نسخه استاندارد این الگوریتم هیچ تأثیری نداشته‌اند (به جز ضعیف‌ترین بازیکن ذخیره) می‌باشد.

**مرحله سوم (انجام رقابت‌های یک فصل لیگ):** مجموعه دستورات خط ۱۳ الی ۲۴ شکل ۳ فرایند رقابت بین تیم‌ها، انتخاب تیم برنده در هر بازی (مقایسه توان تیم‌ها) و انجام عملگرهای تقلید و تهییج می‌باشد. برای تحقق این مهم، ابتدا توان تیم  $i$  ( $i = 1, \dots, M$ ) با توان سایر تیم‌ها مانند تیم  $k$  که  $k \neq i, k = 1, \dots, M$  مقایسه شده و بر اساس احتمال پیروزی هر کدام از دو تیم، تیم برنده انتخاب می‌شود. از طرفی با توجه به کمینه‌سازی تابع برآزش قدرت، تیمی بهتر خواهد بود که مقدار برآزشی کمتری داشته باشد. مقدار احتمال برنده شدن تیم‌های  $i$  و  $k$  به صورت زیر در این نسخه اصلاح شده پیشنهاد گردیده است

$$Pv_i = 1 - \frac{TP_i}{TP_i + TP_k} \quad (13)$$

$$Pv_k = 1 - \frac{TP_k}{TP_i + TP_k} \quad (14)$$

فرض کنیم که تیم برنده، تیم  $i$  باشد. ضمن آن که یک امتیاز به جدول امتیاز<sup>۱</sup> این تیم اضافه می‌شود، عملگرهای تقلید و تهییج به ترتیب بر روی بازیکنان ثابت و ضعیف‌ترین بازیکن ذخیره به شرح ذیل اعمال خواهد شد: **عملگر تقلید:** بازیکنان ثابت ( $FP_{ij}$ ) تیم برنده از بازیکن ستاره تیم خود ( $SP_i$ ) و همچنین از بازیکن فوق ستاره کل لیگ (SSP) تقلید می‌کنند. در نسخه استاندارد این الگوریتم، بازیکنان ثابت تحت عملگر تقلید در امتداد فاصله SSP تا یک بازیکن تصادفی از تیم خود همچون  $RP_{\lambda_i}$  تغییر می‌کنند، در صورتی که توان بازیکن هدف بهبود نیافت، این تقلید در امتداد فاصله  $SP_i$  از  $RP_{\lambda_i}$  مجدد محاسبه شده و اگر توان بهبود نیافت در امتداد فاصله  $RP_{\lambda_i}$  از  $RP_{\lambda_i}$  (دومین بازیکن تصادفی از تیم  $i$  ام) به شرط بهبود توان تغییر خواهد کرد. با توجه به عدم تأثیر اطلاعات بازیکن هدف در محاسبه فاصله از بازیکن فوق ستاره و ستاره و یا یک بازیکن تصادفی از تیم، در این نسخه اصلاح شده، عملگر تقلید به صورت ذیل بازطراحی گردیده است

$$FP_{ij} = FP_{ij} + rand(0.2, 0.8) \times (SSP - FP_{ij}) \quad (15)$$

$$FP_{ij} = FP_{ij} + rand(0.2, 0.8) \times (SP_i - FP_{ij}) \quad (16)$$

$$FP_{ij} = FP_{ij} + rand(0.2, 0.8) \times (RP_{\lambda_i} - FP_{ij}) \quad (17)$$

اگر توان بازیکن ثابت جدید  $FP_{ij}$  بر اساس (۱۵) بهتر از حالت قبل از تقلید از فوق ستاره لیگ بود، ساختار بازیکن جدید جایگزین حالت اولیه

هر الگوریتم به تعداد ۱۰ بار بر روی هر کدام از داده‌های محک اجرا شد که نتایج به دست آمده در جدول ۳ بر اساس کوچک‌ترین زیرمجموعه ویژگی یافت‌شده و در جدول ۵ بر اساس تابع برازش (تابع قدرت بازیکن فوق ستاره) گزارش شده است. در جدول ۴، ویژگی‌های بهترین جواب یافت‌شده توسط هر کدام از الگوریتم‌های پیاده‌سازی شده آمده است.

همان‌طور که از بررسی اولیه جداول ۳ و ۴ مشاهده می‌شود، هرچه اندازه داده‌های محک افزایش یافته است، تفاوت عملکرد الگوریتم‌ها بارزتر شده است. کیفیت نتایج به دست آمده توسط الگوریتم پیشنهادی برای داده‌های بزرگ همچون Musk، LSVT، Urban، Spong و Urban به نسبت چهار الگوریتم دیگر فاصله قابل توجهی هم در بهترین و هم در متوسط عملکرد را نشان می‌دهد. این شرایط برای متوسط عملکرد در داده سائز متوسط Lung و داده‌های سائز کوچک WQ، Derm، Mushroom، M-of-N و قابل مشاهده است. اگرچه به طور کلی عملکرد الگوریتم SLC برای داده‌هایی با اندازه کوچک تقریباً مشابه با سایر الگوریتم‌ها و یا حتی مثلاً در داده‌های Heart و Derm<sup>۲</sup> ضعیف‌تر نیز بوده و الگوریتم پیشنهادی در آنها موفق به کسب بهترین جواب یافت‌شده توسط چهار الگوریتم دیگر نشده است. علت بروز این تفاوت در نتایج ضمن وجود تفاوت‌های ماهیتی در نحوه جستجوی این الگوریتم‌ها، می‌توان به وجود زیرگروه‌های جمعیتی به نام تیم در SLC اشاره کرد که خود شامل ۸ عضو جمعیت در دو کلاس بازیکن ثابت و ذخیره می‌باشند. این امر باعث شده ضمن جستجوی سراسری بر روی بازیکنان ثابت کل تیم‌ها، هر تیم به عنوان یک زیرگروه جمعیتی نیز قابلیت جستجو و بهبود ضمنی داشته باشد. این رویکرد تودرتوی چندگانه و کسب نتایج بهتر برای داده‌های بزرگ توسط SLC در حالی رخ می‌دهد که این الگوریتم به نسبت الگوریتم‌های PSO، AIS، LCA و GA دارای پارامترهای کمتری بوده و تنها نیاز به توزیع جمعیت در ساختار لیگ (تیم-بازیکن) دارد. جزئیات بهترین زیرمجموعه‌های یافت‌شده توسط سه الگوریتم پیاده‌سازی شده در جدول ۴ نمایش داده شده است.

## ۲-۵ تحلیل حساسیت روش پیشنهادی

در تحلیل حساسیت روش پیشنهادی لازم است به طور هم‌زمان به پیچیدگی‌های محاسباتی مسئله کاهش ویژگی و الگوریتم رقابت لیگ فوتبال توجه شود. پیچیدگی زمانی در محاسبه درجه وابستگی تئوری مجموعه راف  $O(m^2 |U| \log(|U|))$  و پیچیدگی حافظه آن نیز  $O(m|U|)$  است که  $|U|$  تعداد اشیا و  $m$  تعداد ویژگی‌های سیستم اطلاعاتی می‌باشند [۳۸]. از طرفی الگوریتم‌های بهینه‌سازی فراابتکاری جمعیت‌محور که دارای جمعیت دوگان نمی‌باشند مانند الگوریتم رقابت لیگ فوتبال و چهار الگوریتم PSO، GA، AIS و LCA با پیچیدگی زمانی  $O(N)$  روبه‌رو هستند که در آنها  $N$  اندازه جمعیت عامل‌های جستجوگر آنها می‌باشد. اما مسئله کاهش ویژگی مبتنی بر تئوری مجموعه راف مفروض در این تحقیق (رابطه (۶))، یک مسئله بهینه‌سازی دوهدفه است. لذا اگر تلاش شود بهترین جواب توسط یک الگوریتم بهینه‌سازی فراابتکاری بر اساس شناسایی جواب‌های بهین پاریتو به دست آید با پیچیدگی زمان محاسباتی  $O(2N^2)$  روبه‌رو خواهد شد [۳۹]. در این تحقیق بر اساس (۱۰)، ترکیب خطی ابتکاری از دو تابع هدف مسئله کاهش ویژگی اولیه (رابطه (۶)) به منظور ارزیابی کیفیت عوامل جستجوگر پیشنهاد شده که باعث گردیده این پیچیدگی زمانی محاسباتی به  $O(N)$  کاهش یابد که با توجه به در نظر گرفتن اندازه جمعیت مساوی با ۲ برابر تعداد ویژگی‌ها این پیچیدگی محاسباتی برابر با  $O(2m)$  خواهد بود.

جدول ۱: مشخصات مجموعه داده‌های محک.

مجموعه داده‌های کوچک (داده‌های مرسوم)				
مشخصات	تعداد اشیا	تعداد ویژگی‌ها	نام داده	شماره
حاوی چند نوع مقدار	۱۰۰۰	۱۳	M-of-N	۱
حاوی چند نوع مقدار	۲۹۴	۱۳	HEART	۲
حاوی چند نوع مقدار	۵۲۱	۳۸	WQ	۳
حاوی چند نوع مقدار	۸۱۲۴	۲۱	MUSHROOM	۴
حاوی چند نوع مقدار	۳۶۶	۳۴	DERM <sup>۱</sup>	۵
حاوی چند نوع مقدار	۳۵۸	۳۴	DERM <sup>۲</sup>	۶
مجموعه داده‌های متوسط				
حاوی چند نوع مقدار	۷۶	۴۵	Sponge	۷
حاوی چند نوع مقدار	۳۲	۵۶	Lung <sup>۳</sup>	۸
مجموعه داده‌های بزرگ				
حاوی چند نوع مقدار	۵۰۷	۱۴۷	Urban <sup>۴</sup>	۹
حاوی چند نوع مقدار	۴۷۶	۱۶۶	Musk <sup>۵</sup>	۱۰
حاوی چند نوع مقدار	۱۲۶	۳۱۰	LSVT <sup>۶</sup>	۱۱

1. Dermatology
2. Dermatology Version 2
3. Lung Cancer
4. Urban Land Cover
5. Musk-Version 1
6. LSVT Voice Rehabilitation

نتایج به دست آمده توسط Microsoft Excel ۲۰۱۲ بر روی یک پردازشگر ۴ هسته ۲/۹۹ گیگاهرتزی و حافظه RAM ۴ گیگابایتی انجام شده است. به منظور تست عملکرد الگوریتم پیشنهادی، مجموعه داده‌های محک استاندارد مشتمل بر ۱۱ داده در سه اندازه کوچک، متوسط و بزرگ که مشخصات آنها در جدول ۱ آمده است، از پایگاه داده معتبر UCI انتخاب شده‌اند.

عملکرد الگوریتم رقابت لیگ فوتبال اصلاح‌شده (SLC) بر روی یازده داده‌های محک مفروض شده با الگوریتم‌های بهینه‌سازی ازدحام ذرات (PSO) [۱۷] و الگوریتم لیگ قهرمانان<sup>۱</sup> (LCA) [۳۶] به عنوان دو الگوریتم پیوسته‌رفتار و الگوریتم ژنتیک (GA) [۱۱] و الگوریتم سیستم ایمنی مصنوعی<sup>۲</sup> (AIS) [۳۷] به عنوان دو الگوریتم گسسته‌رفتار مقایسه شده است. برای الگوریتم‌های PSO و LCA فرایند گسسته‌سازی مشابه فرایند گسسته‌سازی مطرح‌شده در این مقاله انجام گرفته است.

همچنین اندازه جمعیت برای چهار الگوریتم PSO، GA، LCA و AIS برابر با دوبرابر تعداد ویژگی‌های شرطی  $(2 \times m)$  در نظر گرفته شده است. تعداد تیم‌ها و تعداد بازیکنان ثابت و ذخیره SLC پارامترهای آن می‌باشند که با توجه به انتخاب اندازه جمعیت  $2 \times m$  برای الگوریتم‌های PSO، LCA، AIS و GA، می‌بایست این پارامترها به گونه‌ای مناسب تنظیم شوند که از  $2 \times m$  بزرگ‌تر نشود. این مهم به صورت آزمون و خطا برای تعداد بازیکنان ثابت و ذخیره هر کدام ۴ نفر و ۲۵٪ تعداد ویژگی‌ها به عنوان تعداد تیم‌ها تعیین گردید. تنظیم سایر پارامترهای الگوریتم‌های در نظر گرفته شده بر اساس منابعی که این الگوریتم‌ها را برای مسئله کاهش ویژگی مورد استفاده قرار داده بودند به شرح جدول ۲ صورت پذیرفت.

1. League Championship Algorithm
2. Artificial Immune System

جدول ۲: تنظیم پارامترهای الگوریتم‌های پیاده‌سازی شده.

الگوریتم رقابت لیگ فوتبال (SLC)		الگوریتم بهینه‌سازی ازدحام ذرات (PSO)		الگوریتم ژنتیک (GA)		الگوریتم سیستم ایمنی مصنوعی (AIS)		الگوریتم لیگ قهرمانان (LCA)	
پارامتر	مقدار	پارامتر	مقدار	پارامتر	مقدار	پارامتر	مقدار	پارامتر	مقدار
تعداد تیم‌ها	$m/4$	اندازه جمعیت	$m \times 2$	اندازه جمعیت	$m \times 2$	اندازه جمعیت	$m \times 2$	اندازه جمعیت	$m \times 2$
تعداد بازیکنان ثابت	۴	ضریب اینرسی $P_c$	۰٫۹۹	عملگر انتخاب	چرخ رول	انتخاب $S_{size}$	۱۰	کنترلی $P_c$	۰٫۹۹
تعداد بازیکنان ذخیره	۴	ضریب حرکت در امتداد بهترین سراسری $\psi_1$	۱	نرخ لقاء $R_c$	۰٫۹	نرخ تکثیر $R_c$	۰٫۲	ضریب قدرت $\psi_1$	۱
محدوده	$[-8, 8]$	ضریب حرکت در امتداد بهترین موضعی $\psi_2$	۱٫۵	نرخ جهش $R_m$	۰٫۱	نرخ جهش $R_p$	۰٫۵	ضعف $\psi_2$	۰٫۵
محدوده	$[-8, 8]$	محدوده	$[-8, 8]$	محدوده	$\{0, 1\}$	محدوده	$\{0, 1\}$	محدوده	$[-8, 8]$

شرط توقف  
برای داده‌های با ابعاد کوچک و متوسط تعداد ۲۰۰ تکرار  
برای داده‌های با ابعاد بزرگ و خیلی بزرگ تعداد ۴۰۰ تکرار

جدول ۳: تعداد ویژگی‌های کاهش یافته توسط هر الگوریتم.

	SLC	PSO	GA	AIS	LCA
M-of-N	$6^{(A)}, 7^{(T)}$	$6^{(T)}, 7^{(T)}$	$7^{(1)}$	$6^{(T)}, 7^{(T)}$	$6^{(1)}, 7^{(S)}$
Heart	$9^{(1)}$	$8^{(T)}, 9^{(A)}$	$9^{(T)}, 10^{(S)}$	$8^{(T)}, 9^{(T)}$	$9^{(S)}, 10^{(S)}$
Mushroom	$4^{(A)}, 5^{(T)}$	$5^{(T)}, 6^{(T)}$	$5^{(A)}, 6^{(T)}$	$5^{(1)}, 6^{(1)}$	$5^{(1)}$
Derm	$8^{(1)}$	$9^{(T)}, 10^{(T)}$	$8^{(T)}, 9^{(A)}$	$9^{(1)}$	$8^{(T)}, 9^{(T)}$
Derm-2	$9^{(1)}$	$9^{(1)}$	$8^{(T)}, 9^{(A)}$	$8^{(T)}, 9^{(T)}$	$8^{(1)}, 9^{(S)}$
WQ	$12^{(T)}, 13^{(T)}$	$14^{(1)}$	$13^{(T)}, 14^{(T)}, 15^{(T)}$	$13^{(T)}, 14^{(T)}$	$13^{(A)}, 14^{(T)}$
Lung	$4^{(1)}$	$4^{(T)}, 5^{(T)}$	$5^{(1)}$	$5^{(1)}$	$4^{(T)}, 5^{(T)}$
Sponge	$5^{(T)}, 6^{(T)}$	$6^{(T)}, 7^{(A)}$	$8^{(1)}$	$6^{(1)}, 7^{(S)}$	$6^{(S)}, 7^{(S)}$
Urban	$7^{(T)}, 8^{(T)}$	$10^{(T)}, 12^{(T)}$	$13^{(A)}, 14^{(T)}$	$8^{(T)}, 9^{(S)}$	$8^{(T)}, 9^{(T)}$
Musk	$7^{(A)}, 8^{(T)}$	$11^{(A)}, 12^{(T)}$	$13^{(1)}$	$9^{(1)}$	$8^{(T)}, 9^{(T)}$
LSVT	$4^{(T)}, 5^{(T)}$	$9^{(1)}$	$12^{(T)}, 13^{(T)}$	$5^{(T)}, 6^{(T)}$	$5^{(T)}, 6^{(T)}$

بنابراین حل مسئله کاهش ویژگی مفروض در این تحقیق بر اساس الگوریتم‌های بهینه‌سازی فراابتکاری در نظر گرفته شده دارای پیچیدگی زمانی محاسباتی  $O(2m^2 |U| \log(|U|))$  است. با عنایت به یکسان بودن پیچیدگی زمانی الگوریتم‌های مفروض، تفاوت در کمیت و چستی پارامترهای آنها در تحلیل حساسیت عملکرد این الگوریتم در حل مسئله کاهش ویژگی از اهمیت ویژه‌ای برخوردار می‌شود. الگوریتم SLC با دارا بودن تعداد پارامتر کمتر نسبت به چهار الگوریتم دیگر از این جهت ممتاز می‌باشد. این پارامترها نه بر روی نسبت جستجوی سراسری و موضوعی عوامل جستجوگر بلکه بر روی نحوه گروه‌بندی عامل‌ها در تیم‌ها و بازیکن ثابت یا ذخیره‌بودن تأثیر داشته که مزیت دوم پارامترهای SLC نسبت به پارامترهای چهار الگوریتم دیگر می‌باشد. همچنین به منظور انتخاب مناسب تعداد تیم‌ها و تعداد بازیکنان ثابت و ذخیره، با فرض ثابت‌بودن تعداد کل بازیکنان (عامل‌های جستجوگر) معادل ۲ برابر تعداد ویژگی‌ها، بر اساس جدول ۶ اجراهای آزمایشی صورت پذیرفت که در نهایت عملکرد الگوریتم SLC بر اساس  $m/4$  تیم، هر تیم ۴ بازیکن ثابت و ۴ بازیکن ذخیره از سایر گروه‌بندی‌ها بهتر بود.

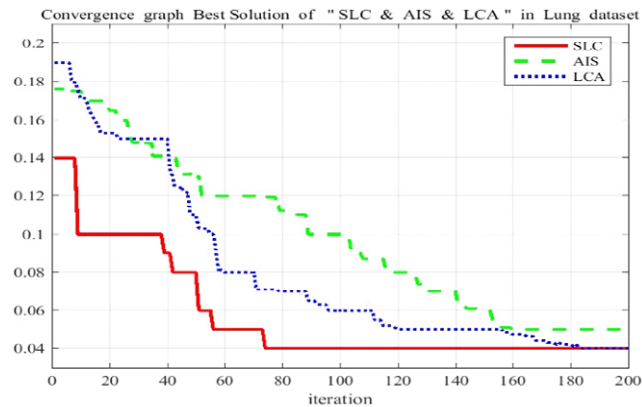
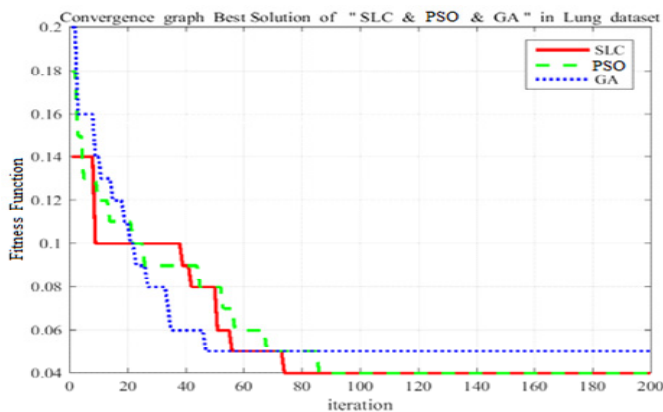
بنابراین حل مسئله کاهش ویژگی مفروض در این تحقیق بر اساس الگوریتم‌های بهینه‌سازی محک متوسط و بزرگ در شکل‌های ۶ تا ۱۰ نمایش داده شده و از نمایش نمودارهای همگرایی بهترین اجرای این الگوریتم‌ها بر روی داده‌های کوچک به سبب لزوم مدیریت حجم مباحث مقاله خودداری شد، اگرچه از جداول ۳ و ۵ می‌توان استنباط کرد که عملکرد الگوریتم SLC در خصوص مجموعه داده‌های کوچک تفاوت چندانی با سایر الگوریتم‌ها نداشته و حتی در مورد برخی از داده‌ها عملکرد ضعیف‌تری را از خود نشان داده است. این مهم در بررسی نمودارهای همگرایی مربوط به داده‌های کوچک نیز آشکار بود به گونه‌ای که برای داده‌های M-of-N و Heart الگوریتم PSO موفق به فرار از بهین موضعی و همگرایی به بهترین مقدار برازش زودتر از سایر الگوریتم و از تکرار به ترتیب ۵ و ۴۸ شده و برای داده‌های Derm و Derm-2 نیز الگوریتم GA موفق به همگرایی سریع‌تر بعد از به ترتیب ۵۰ و ۴۹ شده است. اما در خصوص داده‌های متوسط و بزرگ (شکل‌های ۶ تا ۱۰) عملکرد SLC تمایز آشکاری نسبت به LCA و GA داشته و این تمایز نسبت به PSO و AIS کم‌رنگ‌تر شده است به گونه‌ای که برای داده‌های Sponge (شکل ۷)، Urban (شکل ۸)، Musk (شکل ۹) و LSVT (شکل ۱۰) ضمن فرار از بهین‌های موضعی که سایر الگوریتم‌ها در آن متوقف شده‌اند، موفق به همگرایی سریع‌تر و دستیابی به بهترین جواب نیز شده است.

بنابراین حل مسئله کاهش ویژگی مفروض در این تحقیق بر اساس الگوریتم‌های بهینه‌سازی فراابتکاری در نظر گرفته شده دارای پیچیدگی زمانی محاسباتی  $O(2m^2 |U| \log(|U|))$  است. با عنایت به یکسان بودن پیچیدگی زمانی الگوریتم‌های مفروض، تفاوت در کمیت و چستی پارامترهای آنها در تحلیل حساسیت عملکرد این الگوریتم در حل مسئله کاهش ویژگی از اهمیت ویژه‌ای برخوردار می‌شود. الگوریتم SLC با دارا بودن تعداد پارامتر کمتر نسبت به چهار الگوریتم دیگر از این جهت ممتاز می‌باشد. این پارامترها نه بر روی نسبت جستجوی سراسری و موضوعی عوامل جستجوگر بلکه بر روی نحوه گروه‌بندی عامل‌ها در تیم‌ها و بازیکن ثابت یا ذخیره‌بودن تأثیر داشته که مزیت دوم پارامترهای SLC نسبت به پارامترهای چهار الگوریتم دیگر می‌باشد. همچنین به منظور انتخاب مناسب تعداد تیم‌ها و تعداد بازیکنان ثابت و ذخیره، با فرض ثابت‌بودن تعداد کل بازیکنان (عامل‌های جستجوگر) معادل ۲ برابر تعداد ویژگی‌ها، بر اساس جدول ۶ اجراهای آزمایشی صورت پذیرفت که در نهایت عملکرد الگوریتم SLC بر اساس  $m/4$  تیم، هر تیم ۴ بازیکن ثابت و ۴ بازیکن ذخیره از سایر گروه‌بندی‌ها بهتر بود.

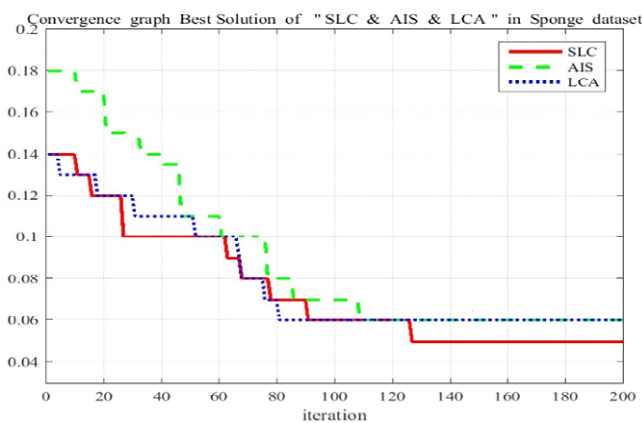
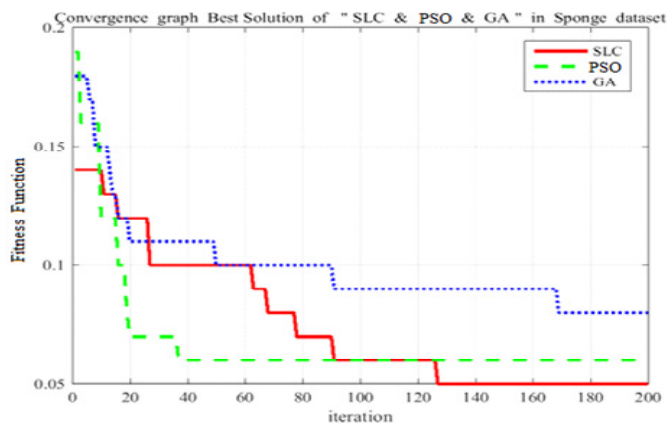
### ۳-۵ مقایسه سرعت همگرایی الگوریتم‌های پیاده‌سازی شده

به منظور مقایسه عملکرد سرعت همگرایی چهار الگوریتم PSO, GA,





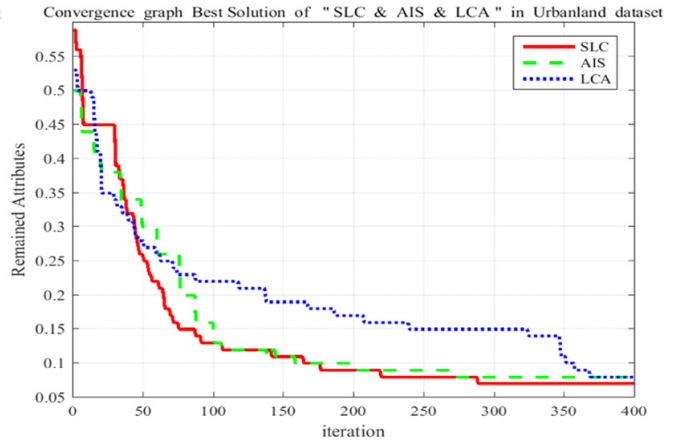
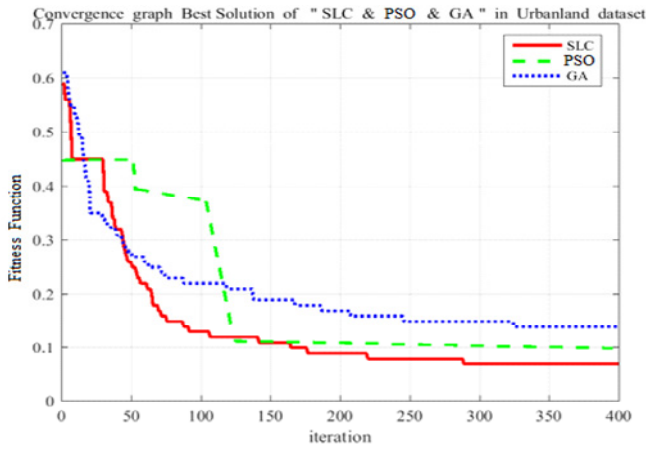
شکل ۶: نمودارهای همگرایی بهترین اجراها برای Lung.



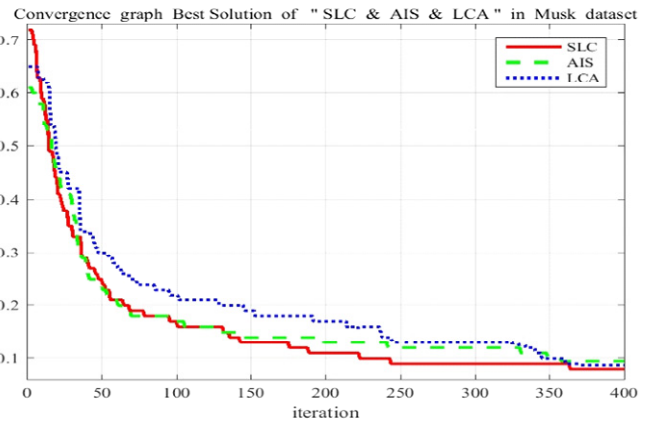
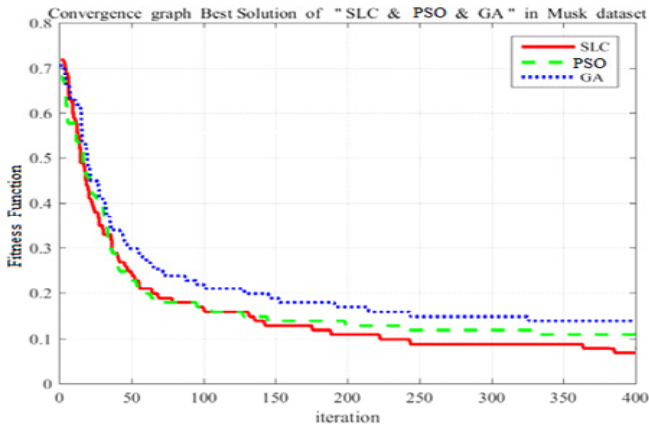
شکل ۷: نمودارهای همگرایی بهترین اجراها برای Sponge.

جدول ۴: جزئیات بهترین کاهش ویژگی یافته شده.

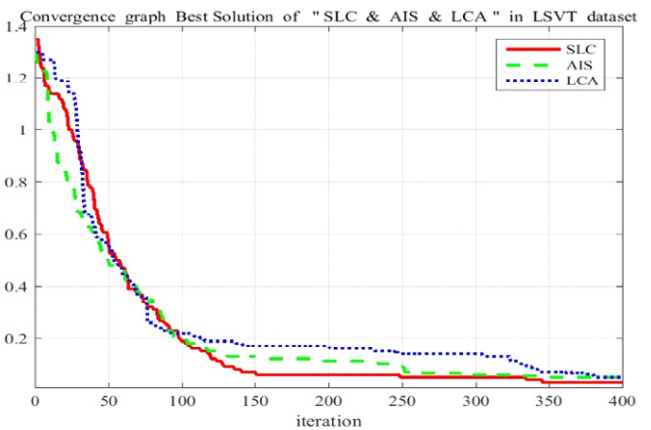
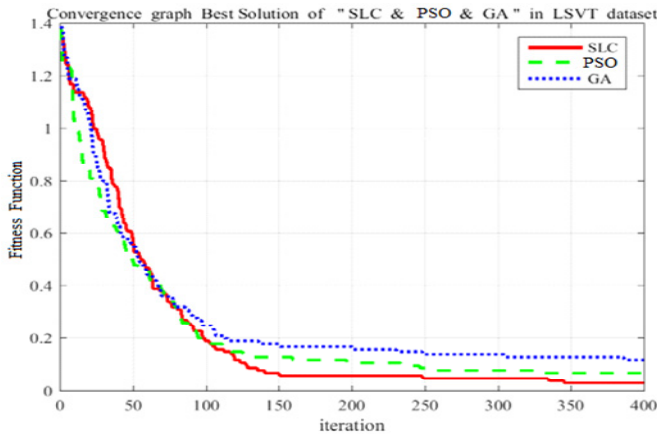
Data sets	SLC	PSO	GA	AIS	LCA
M-of-N	۱,۳,۵,۷,۹,۱۱	۱,۳,۵,۷,۸,۱۱	۱,۳,۵,۷,۸,۹,۱۱	۱,۳,۵,۶,۸,۱۲	۱,۳,۵,۸,۹,۱۱
Heart	۱,۲,۳,۴,۵,۷,۸,۱۰,۱۲	۱,۲,۳,۴,۵,۷,۸,۱۲	۱,۲,۳,۴,۵,۷,۸,۱۰,۱۲	۱,۲,۴,۵,۷,۹,۱۱,۱۲	۱,۲,۳,۴,۵,۶,۹,۱۰,۱۱
Mushroom	۴,۵,۱۱,۲۱	۵,۱۳,۱۴,۲۰,۲۱	۵,۱۳,۱۴,۲۰,۲۱	۴,۹,۱۰,۱۷,۲۰	۵,۷,۱۱,۱۴,۱۸
Derm	۱۴,۱۶,۱۹,۲۱,۲۸,۳۴	۱۷,۱۹,۲۸,۲۹,۳۳,۳۴	۴,۵,۱۷,۱۹,۲۸,۲۹,۳۲	۱۴,۱۷,۱۹,۲۴,۲۸,۳۰	۴,۵,۱۰,۱۱,۱۶,۲۶,۳۱
	۳,۴	۱,۴,۵	۱	۲,۳,۵	۱
Derm-۲	۱۰,۱۴,۲۱,۲۸,۲۹,۳۴	۱۴,۱۵,۲۵,۲۸,۳۳,۳۴	۱۴,۱۶,۲۱,۲۷,۳۲,۳۴	۵,۹,۱۲,۱۳,۲۰,۲۳,۲۷	۱,۳,۷,۸,۱۵,۲۴,۲۶,۳۳
	۳,۴,۵	۳,۵,۷	۲,۳	۴	
WQ	۲۵,۲۷,۳۰,۳۳,۳۴,۳۸	۲۹,۳۰,۳۱,۳۲,۳۳,۳۶	۲۲,۲۷,۳۰,۳۲,۳۴,۳۵	۲۴,۲۹,۳۰,۳۱,۳۳,۳۷	۲۱,۲۵,۳۱,۳۲,۳۳,۳۶
	۱,۵,۱۲,۱۵,۱۶,۲۳	۳,۴,۱۴,۱۶,۱۸,۱۹,۲۷	۱,۵,۶,۱۰,۱۶,۱۷,۲۰	۱,۲,۷,۸,۱۴,۱۹,۲۳	۲,۴,۶,۹,۱۳,۱۶,۲۰
		۲			
Lung	۹,۳۲,۳۵,۳۹	۱۴,۲۱,۳۵,۴۳	۱,۱۳,۱۴,۱۶,۴۲	۳,۱۴,۲۳,۳۰,۳۶	۶,۱۷,۲۹,۴۸
Sponge	۱,۱۹,۳۳,۳۶,۳۹	۱۰,۱۷,۱۸,۳۳,۳۶,۴۰	۲۰,۲۹,۳۲,۳۴,۳۶,۴۱	۴,۱۱,۱۵,۲۶,۳۲,۴۰	۲,۱۰,۱۷,۲۳,۳۱,۴۳
			۱۸,۱۹		
Urban	۶۷,۱۲۶,۱۲۸,۱۲۹,۱۳۵	۴۹,۶۹,۹۷,۱۰۵,۱۱۲	۹۶,۱۰۷,۱۲۲,۱۲۹,۱۴۷	۴۶,۵۹,۷۱,۷۸,۹۲,۱۲۵	۶۶,۷۴,۸۵,۱۱۴,۱۳۱
	۲۵,۶۰	۳,۲۳,۳۳,۳۵,۳۹	۲۳,۳۱,۳۶,۶۶,۷۵,۸۰	۹,۲۸	۲۳,۳۱,۴۷
			۸,۱۱		
Musk	۵۸,۱۳۹,۱۵۲,۱۵۵,۱۵۶	۸۴,۱۱۱,۱۲۴,۱۴۸,۱۵۶	۱۳۴,۱۴۱,۱۵۴,۱۶۱	۸۴,۹۹,۱۱۶,۱۴۱,۱۵۴	۴۴,۶۳,۹۰,۱۲۶,۱۵۳
	۱۱,۱۷	۴,۱۷,۲۲,۳۲,۳۴,۷۹	۳۵,۴۱,۶۳,۶۶,۹۱,۱۱۳	۱۴,۳۲,۵۶,۶۱	۱۱,۲۸,۳۷
			۶,۱۷,۳۲		
LSVT	۱۳,۵۸,۱۰۵,۲۰۰	۲۰۳,۲۳۵,۲۵۰,۲۸۵	۲۰۷,۲۳۰,۲۷۲,۳۰۷	۷,۲۲,۶۹,۱۴۶,۲۵۴	۲۶,۴۰,۸۲,۱۰۹,۲۴۲
		۳۷,۹۸,۱۳۳,۱۵۶,۱۷۲	۸۷,۹۵,۱۱۷,۱۳۳,۱۸۳		
			۱۲,۲۹,۵۶		



شکل ۸: نمودارهای همگرایی بهترین اجراها برای Urban.



شکل ۹: نمودارهای همگرایی بهترین اجراها برای Musk.



شکل ۱۰: نمودارهای همگرایی بهترین اجراها برای LSVT.

$\alpha = 0.05$  اجرا شد. نتایج به دست آمده در قالب  $[Z, P]$  و در جدول ۷ ارائه شده است. اگر مقدار  $P$  کوچکتر از  $0.05$  باشد، فرض صفر یا همان فرض توزیع یکسان داشتن دو نمونه رد خواهد شد و مقایسه آنها معنادار در سطح اطمینان ۹۵٪ خواهد بود. مقادیر ۱ برای  $P$  به معنای یکسان بودن کلیه مقادیر دو نمونه می‌باشد که برای SLC و PSO در داده Derm۲ اتفاق افتاده است. با بررسی جدول ۷ مشاهده می‌شود که غیر از Derm۲ مقایسه PSO با SLC برای داده‌های Heart، M-of-N و نیز Derm۲ مقایسه GA با SLC برای داده Derm۲، مقایسه سایر الگوریتم‌ها با اطمینان ۹۵٪ معنادار می‌باشد.

بر اساس نتایج آزمون ویلکاکسون به دست آمده در جدول ۷، مقایسه الگوریتم پیشنهادی SLC با الگوریتم‌های AIS و LCA به غیر از دو داده

### ۵-۴ تعیین سطح معناداری مقایسه الگوریتم‌ها

با توجه به این که الگوریتم‌های فراابتکاری دارای تکنیک‌ها و پارامترهای تصادفی در فرایند جستجوی خود هستند، نتایج به دست آمده از ۱۰ اجرای الگوریتم‌های یادشده در واقع یک نمونه کوچک از جامعه بسیار بزرگ آنها می‌باشند که به منظور تعیین معنادار بودن مقایسه این نمونه‌ها لازم است آزمون فرض مناسب بر روی نتایج به دست آمده اجرا شود. با عنایت به عدم توزیع نرمال و اندازه کوچک نمونه‌ها، آزمون ویلکاکسون<sup>۱</sup> توسط تحلیل‌گر SPSS بین هر دو نمونه با سطح معناداری

1. Wilcoxon Signed-Rank Test

جدول ۵: تعداد ویژگی‌های کاهش یافته توسط هر الگوریتم پس از فرایند کاهش (در بهترین و بدترین حالت).

		SLC	PSO	GA	AIS	LCA
M-of-N	بهترین	۰/۰۶۰	۰/۰۶۰	۰/۰۷۰	۰/۰۶۰	۰/۰۶۰
	بدترین	۰/۰۷۰	۰/۰۷۰	۰/۰۷۰	۰/۰۷۰	۰/۰۷۰
	متوسط	۰/۰۶۲	۰/۰۶۴	۰/۰۷۰	۰/۰۶۶	۰/۰۶۹
Heart	بهترین	۰/۰۹۰	۰/۰۸۰	۰/۰۹۰	۰/۰۸۰	۰/۰۹۰
	بدترین	۰/۰۹۰	۰/۰۹۰	۰/۱۰۰	۰/۰۹۰	۰/۱۰۰
	متوسط	۰/۰۹۰	۰/۰۸۸	۰/۰۹۶	۰/۰۸۳	۰/۰۹۵
Mushroom	بهترین	۰/۰۴۰	۰/۰۵۰	۰/۰۵۰	۰/۰۵۰	۰/۰۵۰
	بدترین	۰/۰۵۰	۰/۰۶۰	۰/۰۶۰	۰/۰۶۰	۰/۰۵۰
	متوسط	۰/۰۴۲	۰/۰۵۶	۰/۰۵۲	۰/۰۵۱	۰/۰۵۰
Derm	بهترین	۰/۰۸۰	۰/۰۹۰	۰/۰۸۰	۰/۰۹۰	۰/۰۸۰
	بدترین	۰/۰۸۰	۰/۱۰۰	۰/۰۹۰	۰/۰۹۰	۰/۰۹۰
	متوسط	۰/۰۸۰	۰/۰۹۴	۰/۰۸۸	۰/۰۹۰	۰/۰۸۴
Derm-۲	بهترین	۰/۰۹۰	۰/۰۹۰	۰/۰۸۰	۰/۰۸۰	۰/۰۸۰
	بدترین	۰/۰۹۰	۰/۰۹۰	۰/۰۹۰	۰/۰۹۰	۰/۰۹۰
	متوسط	۰/۰۹۰	۰/۰۹۰	۰/۰۸۸	۰/۰۸۷	۰/۰۸۹
WQ	بهترین	۰/۱۲۰	۰/۱۴۰	۰/۱۳۰	۰/۱۳۰	۰/۱۳۰
	بدترین	۰/۱۳۰	۰/۱۴۰	۰/۱۵۰	۰/۱۴۰	۰/۱۴۰
	متوسط	۰/۱۲۴	۰/۱۴۰	۰/۱۴۰	۰/۱۳۴	۰/۱۳۲
Lung	بهترین	۰/۰۴۰	۰/۰۴۰	۰/۰۵۰	۰/۰۵۰	۰/۰۴۰
	بدترین	۰/۰۴۰	۰/۰۵۰	۰/۰۵۰	۰/۰۵۰	۰/۰۵۰
	متوسط	۰/۰۴۰	۰/۰۴۶	۰/۰۵۰	۰/۰۵۰	۰/۰۴۳
Sponge	بهترین	۰/۰۵۰	۰/۰۶۰	۰/۰۸۰	۰/۰۶۰	۰/۰۶۰
	بدترین	۰/۰۶۰	۰/۰۷۰	۰/۰۸۰	۰/۰۷۰	۰/۰۷۰
	متوسط	۰/۰۵۴	۰/۰۶۸	۰/۰۸۰	۰/۰۶۹	۰/۰۶۵
Urban	بهترین	۰/۰۷۰	۰/۱۱۰	۰/۱۳۰	۰/۰۸۰	۰/۰۸۰
	بدترین	۰/۰۸۰	۰/۱۲۰	۰/۱۴۰	۰/۰۹۰	۰/۰۹۰
	متوسط	۰/۰۷۴	۰/۱۱۲	۰/۱۳۲	۰/۰۸۵	۰/۰۸۷
Musk	بهترین	۰/۰۷۰	۰/۱۱۰	۰/۱۳۰	۰/۰۹۰	۰/۰۸۰
	بدترین	۰/۰۸۰	۰/۱۲۰	۰/۱۳۰	۰/۰۹۰	۰/۰۹۰
	متوسط	۰/۰۷۲	۰/۱۱۲	۰/۱۳۰	۰/۰۹۰	۰/۰۸۷
LSVT	بهترین	۰/۰۴۰	۰/۰۹۰	۰/۱۲۰	۰/۰۵۰	۰/۰۵۰
	بدترین	۰/۰۵۰	۰/۰۹۰	۰/۱۳۰	۰/۰۶۰	۰/۰۶۰
	متوسط	۰/۰۴۶	۰/۰۹۰	۰/۱۲۶	۰/۰۵۴	۰/۰۵۶

تقسیم‌بندی داده‌ها به دو بخش آموزش و آزمون از تقسیم‌بندی ۱۰-fold بر اساس اعتبارسنجی متقابل<sup>۲</sup> استفاده شده است. نتایج به دست آمده در جداول ۸ تا ۱۲ به ترتیب برای زیرمجموعه‌های کاهش یافته توسط الگوریتم‌های SLC، PSO، GA، AIS و LCA نمایش داده شده است. به منظور محاسبه دقت<sup>۳</sup>، صحت<sup>۴</sup> و حساسیت<sup>۵</sup> طبقه‌بندی به ترتیب از فرمول‌های ذیل که در منابع متعدد علم داده‌کاوی به آنها اشاره گردیده [۴۰]، استفاده شده است

$$Accuracy = \frac{TP + TN}{P + N} \quad (۲۱)$$

$$Precision = \frac{TP}{TP + FP} \quad (۲۲)$$

$$Sensitivity = \frac{TP}{P} \quad (۲۳)$$

$$Specificity = \frac{TN}{N} \quad (۲۴)$$

که در آنها  $P$ ،  $N$ ،  $TP$ ،  $TN$ ،  $FP$  و  $FN$  به ترتیب تعداد تاپل‌های مثبت، تاپل‌های منفی، مثبت‌های درست، منفی‌های درست، مثبت‌های نادرست و منفی‌های نادرست هستند. در جداول ۸ تا ۱۲، منظور از (O) اشاره به داده اصلی و (R) بهترین زیرمجموعه کاهش یافته توسط الگوریتم‌های یادشده می‌باشد. همان طور که از بررسی نتایج این جداول مستخرج می‌شود، میزان دقت، صحت، حساسیت و وضوح زیرمجموعه کمینه به دست آمده پس از انجام فرایند کاهش توسط الگوریتم SLC در اکثر موارد نسبت به چهار الگوریتم دیگر بهتر است ولی باید اشاره کرد که جنس مقایسه نتایج طبقه‌بندی برای داده‌های اصلی و کاهش یافته با آنچه تا کنون بر اساس آن عملکرد الگوریتم‌های SLC، PSO، GA، AIS و LCA بیان شده متفاوت است. بر اساس پدیده پیکینگ (شکل ۱) کاهش ویژگی ابتدا باعث بهبود کیفیت طبقه‌بندی می‌شود ولی از یک مقدار کاهش ویژگی بیشتر، این مهم باعث تنزل کیفیت طبقه‌بندی نسبت به داده اصلی خواهد شد. به عبارت دیگر پدیده پیکینگ و عدم در نظر گرفتن آن در مکانیزم‌های طراحی شده به منظور کاهش تعداد ویژگی‌ها یک چالش تحقیقاتی باز برای این حوزه تحقیقاتی می‌باشد. از طرفی رویکردهای طراحی شده مانند آنچه در این مقاله هدف تحقیق بود باعث افزایش سرعت پردازش داده‌های کاهش یافته نسبت به داده اصلی برای داده‌های ابعاد بزرگ است.

## ۶- نتیجه‌گیری

این مقاله به بررسی الگوریتم رقابت لیگ فوتبال (SLC) برای حل مسئله کاهش ویژگی مبتنی بر تئوری راف پرداخته است. در واقع قبل از انتخاب این الگوریتم برای حل مسئله کاهش ویژگی با توجه به مطالعه پژوهش‌های گذشته، به چالش‌های الگوریتم‌های فراابتکاری پیاده‌سازی شده برای مسئله تحقیق اشاره شد. از طرفی مزایای الگوریتم SLC از جمله پویایی، پارامتر کم و پتانسیل بالای آن در حل مسایل محاسباتی سنگین، انگیزه جدی در اصلاح و پیاده‌سازی آن به منظور حل مسئله کاهش ویژگی بود. در این پژوهش نه تنها برای اولین بار از الگوریتم

برای هر کدام، در سایر داده‌ها با سطح اطمینان ۹۵٪ معنادار می‌باشند. همچنین مقایسه الگوریتم‌های LCA و AIS در هشت داده همچون Musk, Urban, Sponge, WQ, Derm-۲, Mushroom, M-of-N و LSVT با توجه به مقدار  $\alpha > ۰/۰۵$  به دست آمده برای آنها معنادار نیست. نکته قابل توجه دیگر، تشابه عملکرد و عدم معناداری مقایسه الگوریتم‌های پیاده‌سازی شده برای داده Derm-۲ می‌باشد.

## ۵- ارزیابی اعتبار بهترین زیرمجموعه‌های کاهش یافته

در این قسمت بر روی مجموعه اصلی و نیز بهترین زیرمجموعه کاهش یافته شده توسط هر یک از الگوریتم‌های یادشده طبقه‌بندی بیزین<sup>۱</sup> را به منظور مقایسه دقت، حساسیت، وضوح و صحت طبقه‌بندی داده اصلی با زیرمجموعه کاهش یافته پیاده‌سازی شده است. در

2. Cross Validation
3. Accuracy
4. Precision
5. Sensitivity

جدول ۶: مقادیر مختلف تیمها و بازیکنان ثابت و ذخیره.

تعداد تیمها	m/۲	m/۴	m/۴	m/۶	m/۶	m/۸	m/۱۰
تعداد بازیکنان ثابت	۳	۴	۶	۸	۱۰	۱۰	۱۴
تعداد بازیکنان ذخیره	۱	۴	۲	۴	۲	۶	۶
تعداد کل بازیکنان	m×۲	m×۲	m×۲	m×۲	m×۲	m×۲	m×۲

جدول ۷: نتایج آزمون فرض ویلکاکسون در قالب [Z,P].

	GA vs PSO	PSO vs SLC	GA vs SLC	LCA vs SLC	AIS vs SLC	LCA vs AIS	PSO vs LCA	PSO vs AIS	GA vs AIS	GA vs LCA
M-of-N	[۰,۰۱۴]	[۰,۳۱۷]	[۰,۰۰۵]	[۰,۰۰۸]	[۰,۱۵۷]	[۰,۸۳۰]	[۰,۰۲۵]	[۰,۴۱۴]	[۰,۴۶۰]	[۰,۳۱۷]
	[-۲,۴۴۰]	[-۱,۰۰۰]	[-۲,۸۲۸]	[-۲,۶۴۶]	[-۱,۴۱۴]	[-۱,۷۳۲]	[-۲,۲۳۶]	[-۰,۸۱۶]	[-۲,۰۰۰]	[-۱,۰۰۰]
Heart	[۰,۰۱۱]	[۰,۱۵۷]	[۰,۰۱۴]	[۰,۰۲۵]	[۰,۰۰۸]	[۰,۰۰۳]	[۰,۰۰۸]	[۰,۰۲۵]	[۰,۰۰۹]	[۰,۰۷۰۵]
	[-۲,۵۳۰]	[-۱,۴۱۴]	[-۲,۴۴۹]	[-۲,۲۳۶]	[-۲,۶۴۶]	[-۲,۹۷۲]	[-۲,۶۴۶]	[-۲,۲۳۶]	[-۲,۵۹۸]	[-۰,۳۷۸]
Mushroom	[۰,۰۴۶]	[۰,۰۰۶]	[۰,۰۰۸]	[۰,۰۰۵]	[۰,۰۰۷]	[۰,۳۱۷]	[۰,۰۱۴]	[۰,۰۲۵]	[۰,۳۱۷]	[۰,۱۵۷]
	[-۲,۰۰۰]	[-۲,۷۳۹]	[-۲,۶۴۰]	[-۲,۸۲۵]	[-۲,۷۱۴]	[-۱,۰۰۰]	[-۲,۴۴۹]	[-۲,۲۳۶]	[-۱,۰۰۰]	[-۱,۴۱۴]
Derm	[۰,۰۱۴]	[۰,۰۰۴]	[۰,۰۰۵]	[۰,۰۰۶]	[۰,۰۰۲]	[۰,۰۱۴]	[۰,۰۰۸]	[۰,۰۴۶]	[۰,۱۵۷]	[۰,۰۴۶]
	[-۲,۴۴۹]	[-۲,۸۸۹]	[-۲,۸۲۸]	[-۲,۰۰۰]	[-۳,۱۶۲]	[-۲,۴۴۹]	[-۲,۶۴۰]	[-۲,۰۰۰]	[-۱,۴۱۴]	[-۲,۰۰۰]
Derm-۲	[۰,۱۵۷]	[۱,۰۰۰]	[۰,۱۵۷]	[۰,۳۱۷]	[۰,۸۳۰]	[۰,۱۵۷]	[۰,۳۱۷]	[۰,۰۸۳]	[۰,۳۱۷]	[۰,۳۱۷]
	[-۱,۴۱۴]	[۰,۰۰۰]	[-۱,۴۱۴]	[-۱,۰۰۰]	[-۱,۷۳۲]	[-۱,۴۱۴]	[-۱,۰۰۰]	[-۱,۷۳۲]	[-۱,۰۰۰]	[-۱,۰۰۰]
WQ	[۰,۰۰۴]	[۰,۰۰۴]	[۰,۰۰۴]	[۰,۰۱۱]	[۰,۰۱۵]	[۰,۴۱۴]	[۰,۰۰۵]	[۰,۰۱۴]	[۰,۱۰۷]	[۰,۰۰۵]
	[-۲,۸۴۸]	[-۲,۸۸۹]	[-۲,۸۵۹]	[-۲,۵۳۰]	[-۲,۴۲۸]	[-۰,۸۱۶]	[-۲,۸۲۸]	[-۲,۴۴۹]	[-۱,۶۱۳]	[-۲,۸۲۸]
Lung	[۰,۰۴۶]	[۰,۰۱۴]	[۰,۰۰۲]	[۰,۰۸۳]	[۰,۰۰۲]	[۰,۰۰۸]	[۰,۱۸۰]	[۰,۰۴۶]	[۱,۰۰۰]	[۰,۰۰۸]
	[-۲,۰۰۰]	[-۲,۴۴۹]	[-۳,۱۶۲]	[-۱,۷۳۲]	[-۳,۱۶۲]	[-۲,۶۴۶]	[-۱,۳۴۲]	[-۲,۰۰۰]	[۰,۰۰۰]	[-۲,۶۴۶]
Sponge	[۰,۰۰۳]	[۰,۰۰۴]	[۰,۰۰۴]	[۰,۰۰۵]	[۰,۰۰۶]	[۰,۱۰۲]	[۰,۰۸۳]	[۰,۵۶۴]	[۰,۰۰۲]	[۰,۰۰۴]
	[-۲,۹۷۲]	[-۲,۸۸۹]	[-۲,۸۸۹]	[-۲,۸۱۰]	[-۲,۷۶۲]	[-۱,۶۳۳]	[-۱,۷۳۲]	[-۰,۵۷۷]	[-۳,۰۵۱]	[-۲,۸۷۹]
Urban	[۰,۰۰۴]	[۰,۰۰۵]	[۰,۰۰۴]	[۰,۰۰۶]	[۰,۰۰۲]	[۰,۳۱۷]	[۰,۰۰۵]	[۰,۰۰۵]	[۰,۰۰۴]	[۰,۰۰۴]
	[-۲,۸۴۸]	[-۲,۸۳۱]	[-۲,۸۷۷]	[-۲,۷۳۹]	[-۳,۰۵۱]	[-۱,۰۰۰]	[-۲,۸۳۱]	[-۲,۸۳۱]	[-۲,۸۵۹]	[-۲,۸۷۹]
Musk	[۰,۰۰۳]	[۰,۰۰۲]	[۰,۰۰۳]	[۰,۰۰۶]	[۰,۰۰۳]	[۰,۰۸۳]	[۰,۰۰۴]	[۰,۰۰۳]	[۰,۰۰۲]	[۰,۰۰۴]
	[-۲,۹۷۲]	[-۳,۱۶۲]	[-۲,۹۷۲]	[-۲,۷۶۲]	[-۲,۹۷۲]	[-۱,۷۳۲]	[-۲,۸۸۷]	[-۲,۹۷۲]	[-۳,۱۶۲]	[-۲,۹۱۹]
LSVT	[۰,۰۰۴]	[۰,۰۰۴]	[۰,۰۰۳]	[۰,۰۱۵]	[۰,۰۱۱]	[۰,۴۱۴]	[۰,۰۰۴]	[۰,۰۰۴]	[۰,۰۰۴]	[۰,۰۰۴]
	[-۲,۸۸۹]	[-۲,۸۸۹]	[-۲,۹۷۰]	[-۲,۴۲۸]	[-۲,۵۳۰]	[-۰,۸۱۶]	[-۲,۸۸۹]	[-۲,۸۸۹]	[-۲,۸۷۷]	[-۲,۸۷۳]

آنها در پیاده‌سازی برای این مسئله (جدول ۲) صورت پذیرفت. در بخش ۵ ضمن نمایش نتایج پیاده‌سازی در جداول ۳ تا ۵، سرعت همگرایی این الگوریتمها در شکل‌های ۶ تا ۱۰ مقایسه گردید. خروجی آشکار این جداول و شکل‌های عملکرد قابل رقابت SLC در رسیدن به زیرمجموعه کاهش یافته کوچک‌تر با سرعت همگرایی بالاتر بود. تحلیل حساسیت الگوریتم SLC بر اساس پارامترهای تأثیرگذار در حل مسئله تحقیق، اقدام بعدی در ارزیابی و مقایسه عملکرد این الگوریتم با سایر الگوریتم‌های پیاده‌سازی شده بود. تعداد پارامترهای کمتر SLC و ایجاد هسته‌های جستجوی (تیمها) متمایز ولی مرتبط مزیت رقابتی این الگوریتم در زمانی که دارای پیچیدگی زمان محاسباتی یکسان با چهار الگوریتم دیگر بود، ارزیابی شد. از طرفی ماهیت تصادفی الگوریتم‌های فراابتکاری لزوم اجرای آزمون فرض به منظور محاسبه معناداری مقایسه نتایج به دست آمده را ایجاد می‌نمود که این مهم توسط آزمون فرض ویلکاکسون (جدول ۷) صورت پذیرفت که به غیر از موارد پررنگ‌شده، مقایسه سایر نتایج در سطح قابل قبول ۹۵٪ قرار گرفتند.

آخرین مرحله آزمایش‌ها، پیاده‌سازی طبقه‌بند Naïve Bayes بر روی داده‌های اصلی و بهترین زیرمجموعه‌های کاهش یافته به دست آمده توسط الگوریتم‌های یادشده بود. در این مرحله ضمن برتری نسبی نتایج به دست آمده برای خروجی‌های SLC، آنچه به عنوان یک چالش جدی

رقابت لیگ فوتبال در راستای حل مسئله کاهش ویژگی بهره گرفته شد، بلکه اصلاحات در ساختار الگوریتم اصلی و ایده‌هایی به منظور سازگاری این الگوریتم پیوسته‌رفتار با مسئله گسسته‌رفتار هدف تحقیق ارائه گردید. بهره‌گیری از مجموع توان بازیکنان ثابت و ذخیره در محاسبه توان هر تیم، در نظر گرفتن ترکیب ساختارهای پیوسته و گسسته برای هر بازیکن، ارائه یک مکانیزم ابتکاری گسسته‌سازی جدید، ارائه تحلیل هیدرولیک متناسب با مسئله تحقیق برای فرایند برازش هر بازیکن، ارائه اصلاح در عملگرهای تقلید و تهییج بر اساس چالش‌های نسخه اصلی از جمله این تغییرات نسخه اصلاح‌شده می‌باشند.

همچنین علاوه بر داده‌های محک معروف مورد استفاده در اکثر مقالات گذشته، از داده‌هایی با ابعاد متوسط و بزرگ (جدول ۱) نیز به منظور تست عملکرد ایده پیشنهادی و مقایسه با دو الگوریتم بهینه‌سازی پیوسته‌رفتار PSO و LCA و دو الگوریتم گسسته‌مقدار همچون GA و AIS در نظر گرفته شد. شایان ذکر است که الگوریتم لیگ قهرمانان (LCA) برای اولین بار برای مسئله تحقیق پیاده‌سازی و سازگار شد و علت انتخاب نزدیکی ایده الهام شکل‌دهنده آن به ایده فرم‌دهنده الگوریتم رقابت لیگ فوتبال بود. نحوه تنظیم پارامترهای این الگوریتمها برای SLC و LCA بر اساس آزمون و خطا در پیاده‌سازی بر روی داده Heart و برای سه الگوریتم دیگر بر اساس تنظیمات معرفی‌شده در منابع گذشته

جدول ۸: نتایج طبقه‌بندی داده‌های اصلی و زیرمجموعه کاهش یافته با SLC توسط طبقه‌بند NAIVE BAYES.

روش	صحت		حساسیت		دقت	
	آموزش	تست	آموزش	تست	آموزش	تست
M-of-N (O)	۰.۷۴۹	۰.۷۴۷	۱	۱	۰.۷۱۵	۰.۷۱۳
M-of-N (R)	۰.۷۹۹	۰.۷۹۸	۱	۱	۰.۷۵۹	۰.۷۵۷
Heart (O)	۰.۸۱۸	۰.۸۲۰	۰.۶۲۷	۰.۶۵۲	۰.۸۲۷	۰.۸۱۱
Heart (R)	۰.۷۷۱	۰.۷۶۹	۰.۵۶۶	۰.۵۶۵	۰.۷۹۶	۰.۷۹۲
Mushroom (O)	۰.۹۲۰	۰.۹۱۹	۰.۹۶۸	۰.۹۶۹	۰.۸۸۸	۰.۸۸۷
Mushroom (R)	۰.۸۹۶	۰.۸۹۴	۰.۹۵۳	۰.۹۵۲	۰.۸۶۰	۰.۸۶۰
Derm (O)	۰.۹۴۴	۰.۹۲۹	۱	۰.۹۹۱	۰.۸۴۸	۰.۸۱۸
Derm (R)	۰.۷۹۱	۰.۷۴۹	۰.۹۷۵	۰.۹۵۵	۰.۵۹۸	۰.۵۵۴
Derm <sub>2</sub> (O)	۰.۹۴۹	۰.۹۴۷	۱	۰.۹۹۱	۰.۸۶۳	۰.۸۶۴
Derm <sub>2</sub> (R)	۰.۸۴۴	۰.۸۲۷	۰.۹۶۶	۰.۹۳۶	۰.۶۷۴	۰.۶۵۴
WQ (O)	۰.۶۴۶	۰.۶۳۱	۰.۹۹۸	۱	۰.۵۹۹	۰.۵۸۹
WQ (R)	۰.۵۷۶	۰.۵۶۸	۰.۹۸۲	۰.۹۷۸	۰.۵۵۶	۰.۵۵۱
Lung (O)	۰.۸۵۵	۰.۷۷۵	۰.۶۲۱	۰.۳۰۰	۰.۹۷۱	۰.۳۰۰
Lung (R)	۰.۸۷۸	۰.۸۷۵	۰.۵۶۸	۰.۵۰۰	۱	۰.۵۰۰
Spong (O)	۰.۹۴۹	۰.۷۸۷	۱	۰.۶۰۰	۰.۶۵۷	۰.۲۹۲
Spong (R)	۰.۶۶۴	۰.۴۷۷	۰.۶۸۶	۰.۳۰۰	۰.۱۶۹	۰.۰۷۲
Urban (O)	۰.۹۰۵	۰.۸۱۶	۰.۹۳۳	۰.۸۶۱	۰.۶۷۶	۰.۵۰۷
Urban (R)	۰.۷۵۰	۰.۷۰۰	۰.۸۲۴	۰.۷۶۲	۰.۴۰۹	۰.۳۵۵
Musk (O)	۰.۹۰۴	۰.۸۳۲	۰.۹۸۹	۰.۹۱۳	۰.۸۲۶	۰.۷۵۸
Musk (R)	۰.۹۰۴	۰.۸۳۰	۰.۹۸۹	۰.۸۹۴	۰.۸۲۴	۰.۷۶۰
LSVT (O)	۰.۸۷۶	۰.۷۲۳	۱	۰.۶۴۰	۰.۷۳۰	۰.۵۶۹
LSVT (R)	۰.۷۸۶	۰.۷۷۰	۰.۷۰۶	۰.۶۵۵	۰.۶۹۹	۰.۷۰۹

جدول ۹: نتایج طبقه‌بندی داده‌های اصلی و زیرمجموعه کاهش یافته با PSO توسط طبقه‌بند NAIVE BAYES.

روش	صحت		حساسیت		دقت	
	آموزش	تست	آموزش	تست	آموزش	تست
M-of-N (O)	۰.۷۴۹	۰.۷۴۷	۱	۱	۰.۷۱۵	۰.۷۱۴
M-of-N (R)	۰.۸۳۸	۰.۸۳۸	۱	۱	۰.۷۹۶	۰.۷۹۶
Heart (O)	۰.۸۱۸	۰.۸۲۰	۰.۶۲۷	۰.۶۵۱	۰.۸۲۷	۰.۸۱۱
Heart (R)	۰.۷۶۳	۰.۷۴۸	۰.۴۷۷	۰.۴۵۴	۰.۷۸۷	۰.۷۷۴
Mushroom (O)	۰.۹۲۰	۰.۹۱۹	۰.۹۶۸	۰.۹۶۸	۰.۸۸۸	۰.۸۸۷
Mushroom (R)	۰.۹۱۵	۰.۹۱۵	۰.۹۲۸	۰.۹۲۸	۰.۹۱۰	۰.۹۱۰
Derm (O)	۰.۹۴۴	۰.۹۲۹	۱	۰.۹۹۱	۰.۸۴۸	۰.۸۱۸
Derm (R)	۰.۸۱۳	۰.۷۴۰	۰.۹۲۴	۰.۸۵۶	۰.۶۳۲	۰.۵۵۲
Derm <sub>2</sub> (O)	۰.۹۴۹	۰.۹۴۷	۱	۰.۹۹۰	۰.۸۶۳	۰.۸۶۴
Derm <sub>2</sub> (R)	۰.۸۹۱	۰.۸۷۶	۰.۹۹۳	۰.۹۵۴	۰.۷۴۳	۰.۷۰۴
WQ (O)	۰.۶۴۶	۰.۶۳۱	۰.۹۹۸	۱	۰.۵۹۹	۰.۵۸۹
WQ (R)	۰.۶۳۰	۰.۶۱۰	۰.۹۷۵	۰.۹۷۴	۰.۵۹۰	۰.۵۷۸
Lung (O)	۰.۸۵۵	۰.۷۵۵	۰.۶۲۰	۰.۳۰۰	۰.۹۷۱	۰.۳۰۰
Lung (R)	۰.۸۰۵	۰.۷۵۸	۰.۳۸۷	۰.۲۰۰	۰.۷۸۰	۰.۲۰۰
Spong (O)	۰.۹۴۹	۰.۷۸۷	۱	۰.۶۰۰	۰.۶۵۷	۰.۲۹۲
Spong (R)	۰.۷۸۱	۰.۵۸۷	۰.۷۴۵	۰.۴۰۰	۰.۲۶۱	۰.۱۶۵
Urban (O)	۰.۹۰۵	۰.۸۱۶	۰.۹۳۳	۰.۸۶۱	۰.۶۷۶	۰.۵۰۷
Urban (R)	۰.۷۹۹	۰.۶۹۹	۰.۸۱۴	۰.۷۴۰	۰.۴۷۲	۰.۳۲۷
Musk (O)	۰.۹۰۴	۰.۸۳۲	۰.۹۸۹	۰.۹۱۳	۰.۸۲۶	۰.۷۵۸
Musk (R)	۰.۷۴۲	۰.۷۱۲	۰.۹۶۴	۰.۹۰۳	۰.۶۳۴	۰.۶۱۶
LSVT (O)	۰.۸۷۶	۰.۷۲۳	۱	۰.۶۴۰	۰.۷۳۰	۰.۵۶۹
LSVT (R)	۰.۸۲۷	۰.۷۱۵	۰.۹۷۶	۰.۷۴۰	۰.۶۶۴	۰.۵۵۰

جدول ۱۰: نتایج طبقه‌بندی داده‌های اصلی و زیرمجموعه کاهش یافته با GA توسط طبقه‌بند NAIVE BAYES.

روش	صحت		حساسیت		دقت	
	آموزش	تست	آموزش	تست	آموزش	تست
M-of-N (O)	۰.۷۴۹	۰.۷۴۷	۱	۱	۰.۷۱۵	۰.۷۱۴
M-of-N (R)	۰.۷۷۸	۰.۷۷۶	۱	۱	۰.۷۴۰	۰.۷۳۸
Heart (O)	۰.۸۱۸	۰.۸۱۹	۰.۶۲۷	۰.۶۵۲	۰.۸۲۷	۰.۸۱۱
Heart (R)	۰.۷۷۱	۰.۷۶۹	۰.۵۶۶	۰.۵۶۵	۰.۷۹۶	۰.۷۹۲
Mushroom (O)	۰.۹۲۰	۰.۹۱۹	۰.۹۶۸	۰.۹۶۸	۰.۸۸۸	۰.۸۸۷
Mushroom (R)	۰.۹۱۵	۰.۹۱۵	۰.۹۲۸	۰.۹۲۸	۰.۹۱۰	۰.۹۱۰
Derm (O)	۰.۹۴۴	۰.۹۲۹	۱	۰.۹۹۱	۰.۸۴۸	۰.۸۱۸
Derm (R)	۰.۸۲۸	۰.۸۰۹	۰.۹۵۰	۰.۹۵۴	۰.۶۵۱	۰.۶۲۹
Derm۲ (O)	۰.۹۴۹	۰.۹۴۷	۱	۰.۹۹۱	۰.۸۶۳	۰.۸۶۴
Derm۲ (R)	۰.۸۳۲	۰.۸۰۲	۰.۹۸۲	۰.۹۸۲	۰.۶۵۳	۰.۶۱۶
WQ (O)	۰.۶۴۶	۰.۶۳۱	۰.۹۹۸	۱	۰.۵۹۱	۰.۵۸۹
WQ (R)	۰.۶۴۶	۰.۶۲۲	۰.۹۷۸	۰.۹۶۸	۰.۶۰۱	۰.۵۸۷
Lung (O)	۰.۸۸۵	۰.۷۷۵	۰.۶۲۰	۰.۳۰۰	۰.۹۷۱	۰.۳۰۰
Lung (R)	۰.۸۵۸	۰.۷۲۵	۰.۷۸۲	۰.۶۰۰	۰.۷۵۱	۰.۳۸۳
Spong (O)	۰.۹۴۹	۰.۷۸۷	۱	۰.۶۰۰	۰.۶۵۷	۰.۲۹۲
Spong (R)	۰.۶۸۱	۰.۵۳۶	۰.۹۸۶	۰.۶۰۰	۰.۲۲۵	۰.۱۵۰
Urban (O)	۰.۹۰۵	۰.۸۱۶	۰.۹۳۳	۰.۸۶۱	۰.۶۷۵	۰.۵۰۷
Urban (R)	۰.۸۰۹	۰.۷۲۲	۰.۸۱۲	۰.۷۷۳	۰.۴۸۸	۰.۳۷۸
Musk (O)	۰.۹۰۴	۰.۸۳۲	۰.۹۸۹	۰.۹۱۳	۰.۸۲۶	۰.۷۵۸
Musk (R)	۰.۷۲۵	۰.۶۷۹	۰.۹۳۷	۰.۸۴۹	۰.۶۲۲	۰.۵۹۵
LSVT (O)	۰.۸۷۶	۰.۷۲۳	۱	۰.۶۴۰	۰.۷۳۰	۰.۵۶۹
LSVT (R)	۰.۸۵۹	۰.۷۳۰	۰.۸۸۱	۰.۶۰۰	۰.۷۵۱	۰.۶۰۴

جدول ۱۱: نتایج طبقه‌بندی داده‌های اصلی و زیرمجموعه کاهش یافته با AIS توسط طبقه‌بند NAIVE BAYES.

روش	صحت		حساسیت		دقت	
	آموزش	تست	آموزش	تست	آموزش	تست
M-of-N (O)	۰.۷۴۹	۰.۷۴۷	۱	۱	۰.۷۱۵	۰.۷۱۴
M-of-N (R)	۰.۷۵۷	۰.۷۸۰	۱	۱	۰.۷۳۰	۰.۷۵۴
Heart (O)	۰.۸۱۸	۰.۸۲۰	۰.۶۲۷	۰.۶۵۲	۰.۸۲۷	۰.۸۱۱
Heart (R)	۰.۷۰۱	۰.۷۱۴	۰.۵۶۱	۰.۵۵۲	۰.۷۵۲	۰.۷۳۴
Mushroom (O)	۰.۹۲۰	۰.۹۱۹	۰.۹۶۸	۰.۹۶۹	۰.۸۸۸	۰.۸۸۷
Mushroom (R)	۰.۹۲۱	۰.۹۲۹	۰.۹۰۲	۰.۹۱۴	۰.۸۵۱	۰.۸۴۳
Derm (O)	۰.۹۴۴	۰.۹۲۹	۱	۰.۹۹۱	۰.۸۴۸	۰.۸۱۸
Derm (R)	۰.۸۵۷	۰.۸۹۱	۰.۹۷۷	۰.۹۷۹	۰.۷۵۲	۰.۶۳۷
Derm۲ (O)	۰.۹۴۹	۰.۹۴۷	۱	۰.۹۹۱	۰.۸۶۳	۰.۸۶۴
Derm۲ (R)	۰.۹۰۶	۰.۸۸۱	۰.۹۷۱	۰.۹۰۲	۰.۸۶۹	۰.۸۶۷
WQ (O)	۰.۶۴۶	۰.۶۳۱	۰.۹۹۸	۱	۰.۵۹۹	۰.۵۸۹
WQ (R)	۰.۶۰۱	۰.۵۹۹	۰.۹۳۱	۰.۹۴۹	۰.۵۶۷	۰.۵۵۲
Lung (O)	۰.۸۵۵	۰.۷۷۵	۰.۶۲۱	۰.۳۰۰	۰.۹۷۱	۰.۳۰۰
Lung (R)	۰.۸۳۲	۰.۷۶۲	۰.۶۱۰	۰.۳۰۰	۰.۹۸۰	۰.۳۰۹
Spong (O)	۰.۹۴۹	۰.۷۸۷	۱	۰.۶۰۰	۰.۶۵۷	۰.۲۹۲
Spong (R)	۰.۸۰۳	۰.۶۶۷	۰.۸۹۶	۰.۵۰۸	۰.۴۰۱	۰.۲۰۱
Urban (O)	۰.۹۰۵	۰.۸۱۶	۰.۹۳۳	۰.۸۶۱	۰.۶۷۶	۰.۵۰۷
Urban (R)	۰.۸۴۲	۰.۷۰۶	۰.۸۲۹	۰.۷۳۵	۰.۵۰۹	۰.۴۴۷
Musk (O)	۰.۹۰۴	۰.۸۳۲	۰.۹۸۹	۰.۹۱۳	۰.۸۲۶	۰.۷۵۸
Musk (R)	۰.۸۰۸	۰.۷۳۱	۰.۹۲۳	۰.۸۵۳	۰.۷۱۲	۰.۶۷۸
LSVT (O)	۰.۸۷۶	۰.۷۲۳	۱	۰.۶۴۰	۰.۷۳۰	۰.۵۶۹
LSVT (R)	۰.۸۰۶	۰.۷۲۹	۰.۹۳۱	۰.۶۱۰	۰.۶۸۹	۰.۵۱۹

جدول ۱۲: نتایج طبقه‌بندی داده‌های اصلی و زیرمجموعه کاهش یافته با LCA توسط طبقه‌بند NAIVE BAYES.

روش	صحت		حساسیت		دقت	
	تست	آموزش	تست	آموزش	تست	آموزش
M-of-N (O)	۰٫۷۴۷	۰٫۷۴۹	۱	۱	۰٫۷۱۴	۰٫۷۱۵
M-of-N (R)	۰٫۷۶۸	۰٫۷۵۵	۱	۱	۰٫۷۲۶	۰٫۷۲۹
Heart (O)	۰٫۸۲۰	۰٫۸۱۸	۰٫۶۵۲	۰٫۶۲۷	۰٫۸۱۱	۰٫۸۲۷
Heart (R)	۰٫۶۶۴	۰٫۶۸۱	۰٫۵۱۰	۰٫۵۱۰	۰٫۷۱۳	۰٫۷۰۲
Mushroom (O)	۰٫۹۱۹	۰٫۹۲۰	۰٫۹۶۹	۰٫۹۶۸	۰٫۸۸۷	۰٫۸۸۸
Mushroom (R)	۰٫۸۶۴	۰٫۸۵۱	۰٫۹۱۴	۰٫۹۰۹	۰٫۸۵۹	۰٫۸۴۱
Derm (O)	۰٫۹۲۹	۰٫۹۴۴	۱	۱	۰٫۸۱۸	۰٫۸۴۹
Derm (R)	۰٫۹۲۹	۰٫۹۳۰	۰٫۹۹۸	۰٫۹۸۷	۰٫۷۷۱	۰٫۸۲۱
Derm۲ (O)	۰٫۹۴۷	۰٫۹۴۹	۱	۱	۰٫۸۶۴	۰٫۸۶۲
Derm۲ (R)	۰٫۸۸۷	۰٫۸۱۰	۰٫۹۹۹	۰٫۹۸۹	۰٫۷۱۱	۰٫۷۴۲
WQ (O)	۰٫۶۳۱	۰٫۶۴۶	۱	۰٫۹۹۸	۰٫۵۸۹	۰٫۵۹۹
WQ (R)	۰٫۶۱۴	۰٫۶۰۷	۰٫۹۱۰	۰٫۹۰۳	۰٫۵۴۴	۰٫۵۵۳
Lung (O)	۰٫۷۵۵	۰٫۸۸۵	۰٫۳۰۰	۰٫۶۲۱	۰٫۳۰۰	۰٫۹۷۱
Lung (R)	۰٫۷۱۴	۰٫۸۰۸	۰٫۶۰۹	۰٫۶۰۹	۰٫۲۷۰	۰٫۸۹۱
Spong (O)	۰٫۷۸۷	۰٫۹۴۹	۱	۱	۰٫۲۹۲	۰٫۶۵۷
Spong (R)	۰٫۶۶۵	۰٫۷۵۶	۰٫۶۰۰	۰٫۹۹۰	۰٫۲۹۹	۰٫۵۶۴
Urban (O)	۰٫۸۱۶	۰٫۹۰۵	۰٫۸۶۱	۰٫۹۳۳	۰٫۵۰۷	۰٫۶۷۶
Urban (R)	۰٫۷۱۱	۰٫۸۱۰	۰٫۷۳۵	۰٫۸۴۵	۰٫۳۷۶	۰٫۵۳۴
Musk (O)	۰٫۸۳۲	۰٫۹۰۴	۰٫۹۱۳	۰٫۹۸۹	۰٫۷۵۸	۰٫۸۲۶
Musk (R)	۰٫۶۴۳	۰٫۸۱۲	۰٫۷۹۹	۰٫۸۶۷	۰٫۷۵۹	۰٫۸۲۷
LSVT (O)	۰٫۷۲۳	۰٫۸۷۶	۱	۱	۰٫۵۶۹	۰٫۷۳۰
LSVT (R)	۰٫۶۵۴	۰٫۷۷۶	۰٫۶۱۱	۰٫۹۴۵	۰٫۵۳۷	۰٫۷۰۲

*J. of Advanced Manufacturing Technology*, vol. 27, no. 1/2, pp. 119-127, Jan. 2005.

- [7] Y. G. Ren, Y. Wang, and D. Q. Yan, "Rough set attribute reduction algorithm based on GA," *Mini-Micro System*, vol. 27, no. 5, pp. 862-865, Mar. 2006.
- [8] N. S. Jaddi and S. Abdullah, "Hybrid of genetic algorithm and great deluge algorithm for rough set attribute reduction," *Turkish J. of Electrical Engineering & Computer Sciences*, vol. 21, no. 6, pp. 1737-1750, 2013.
- [9] S. Y. Jing, "A hybrid genetic algorithm for feature subset selection in rough set theory," *Soft Computing*, vol. 18, no. 7, pp. 1373-1382, Oct. 2014.
- [10] Z. Xu, D. Gu, and B. Yang, *Attribute Reduction Algorithm Based on Genetic Algorithm*, 2009.
- [11] A. K. Das, S. Sengupta, and S. Bhattacharyya, "A group incremental feature selection for classification using rough set theory based genetic algorithm," *Applied Soft Computing*, vol. 65, pp. 400-411, Apr. 2018.
- [12] R. Jensen and Q. Shen, "Finding rough set reducts with ant colony optimization," in *Proc. UK Workshop on Computational Intelligence*, 8 pp., 2003.
- [13] L. Ke, Z. Feng, and Z. Ren, "An efficient ant colony optimization approach to attribute reduction in rough set theory," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1351-1357, Jul. 2008.
- [14] T. Q. Deng, et al., "An improved ant colony optimization applied to attributes reduction," in *Proc. Fuzzy Information and Engineering*, vol. 54, 6 pp., Springer, Berlin, 2009.
- [15] Y. Chen, D. Miao, and R. Wang, "A rough set approach to feature selection based on ant colony optimization," *Pattern Recognition Letters*, vol. 31, no. 3, pp. 226-233, Feb. 2010.
- [16] M. M. Kabir, M. Shahjahan, and K. Murase, "A new hybrid ant colony optimization algorithm for feature selection," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3747-3763, Feb. 2012.
- [17] X. Wang, et al., "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 459-471, Mar. 2007.
- [18] D. Ye, Z. Chen, and J. Liao, "A new algorithm for minimum attribute reduction based on binary particle swarm optimization with vaccination," in *Pacific-Asia Conf. on Knowledge Discovery and Data Mining, PAKDD'07*, pp. 1029-1036, May 2007.

آشکار شد، پدیده پیکینگ بود که کاهش حداکثری اندازه زیرمجموعه‌های انتخابی، کیفیت طبقه‌بندی را تنزل داده بود. کاهش حداکثری ابعاد داده‌ها در موارد مواجهه با داده‌های ابعاد بزرگ و اهمیت هزینه محاسباتی بالای این داده‌ها بدون شک مفید خواهد بود ولی در مواردی که کیفیت الگوهای پنهان درون داده از هزینه محاسباتی بالاتر باشد، لازم است با توجه به پدیده پیکینگ، مکانیزمی به منظور کشف بهترین زیرمجموعه با بالاترین کیفیت الگوی پنهان طراحی گردد. با توجه به ساختار تئوری مجموعه راف، طراحی روش‌های ترکیبی به منظور توجه بیشتر به پدیده پیکینگ، موازی‌سازی الگوریتم‌های فراابتکاری، حافظه‌مندی و فازی‌سازی ساختار الگوریتم SLC از جمله جهت‌های تحقیقاتی آتی هستند که این مقاله آنها را برای محققین باز می‌نماید.

## مراجع

- [1] C. Sima and E. R. Dougherty, "The peaking phenomenon in the presence of feature-selection," *Pattern Recognition Letters*, vol. 29, no. 11, pp. 1667-1674, Aug. 2008.
- [2] Z. Pawlak, "Rough sets," *International J. of Computer & Information Sciences*, vol. 11, no. 5, pp. 341-356, Oct. 1982.
- [3] M. Abdolrazzagah-Nezhad, H. Radgohar, and S. N. Salimian, "Enhanced cultural algorithm to solve multi-objective attribute reduction based on rough set theory," *Mathematics and Computers in Simulation*, vol. 170, pp. 332-350, Apr. 2020.
- [4] M. Ghanei Ostad, H. Khosravi Mahmoee, and M. Abdolrazzagah Nezhad, "Attribute reduction in incomplete information system based on rough set theory using fuzzy imperialist competitive algorithm," *J. of Information Technology Management*, vol. 9, no. 1, pp. 123-142, Spring 2017.
- [5] J. H. Dai and Y. X. Li, *Heuristic Genetic Algorithm for Minimal Reduction Decision System Based on Rough Set Theory*, 2002.
- [6] C. Huang, T. Li, and T. Peng, "A hybrid approach of rough set theory and genetic algorithm for fault diagnosis," *The International*

- [35] N. Moosavian and B. K. Roodsari, "Soccer league competition algorithm: a novel meta-heuristic algorithm for optimal design of water distribution networks," *Swarm and Evolutionary Computation*, vol. 17, pp. 14-24, Aug. 2014.
- [36] A. H. Kashan, "League Championship Algorithm (LCA): an algorithm for global optimization inspired by sport championships," *Applied Soft Computing*, vol. 16, pp. 171-200, Mar. 2014.
- [37] G. Dudek, "An artificial immune system for classification with local feature selection," *IEEE Trans. on Evolutionary Computation*, vol. 16, no. 6, pp. 847-860, Dec. 2012.
- [38] N. S. Hoa and N. H. Son, "Some efficient algorithms for rough set methods," in *Proc. IPMU*, pp. 1451-1456, Granada, Spain, 1-5 Jul. 1996.
- [39] K. Deb, *Multi-Objective Optimization*, in Search Methodologies, Springer, pp. 403-449, 2014.
- [40] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*, Elsevier, 2011.
- [19] L. Cervante, B. Xue, L. Shang, and M. Zhang, "Binary particle swarm optimisation and rough set theory for dimension reduction in classification," in *Proc. IEEE Congress on Evolutionary Computation, CEC'13*, pp. 2428-2435, Cancun, Mexico, 20-23 Jun. 2013.
- [20] J. Dai, et al., "Discrete particle swarm optimization approach for cost sensitive attribute reduction," *Knowledge-Based Systems*, vol. 102, pp. 116-126, Jun. 2016.
- [21] N. Suguna and K. Thanushkodi, "A Novel Rough Set Reduct Algorithm for Medical Domain Based on Bee Colony Optimization," arXiv preprint arXiv:1006.4540, Jun. 2010.
- [22] D. Ye and Z. Chen, "A new approach to minimum attribute reduction based on discrete artificial bee colony," *Soft Computing*, vol. 19, no. 7, pp. 1893-1903, Jul. 2015.
- [23] J. Cai, et al., "An improved artificial bee colony algorithm for minimal time cost reduction," *International J. of Machine Learning and Cybernetics*, vol. 5, no. 5, pp. 743-752, Dec. 2014.
- [24] J. Ganesan, et al., "Tolerance rough set firefly-based quick reduct," *Neural Computing and Applications*, vol. 28, no. 10, pp. 2995-3008, Aug. 2017.
- [25] X. Y. Luan, Z. P. Li, and T. Z. Liu, "A novel attribute reduction algorithm based on rough set and improved artificial fish swarm algorithm," *Neurocomputing*, vol. 174, pt. A, pp. 522-529, Jan. 2016.
- [26] R. Cheruku, et al., "Rst-batminer: a fuzzy rule miner integrating rough set feature selection and bat optimization for detection of diabetes disease," *Applied Soft Computing*, vol. 67, pp. 764-780, Jun. 2018.
- [27] W. Ding, et al., "Enhanced minimum attribute reduction based on quantum-inspired shuffled frog leaping algorithm," *J. of Systems Engineering and Electronics*, vol. 24, no. 3, pp. 426-434, Jun. 2013.
- [28] M. A. El Aziz and A. E. Hassanien, "An improved social spider optimization algorithm based on rough sets for solving minimum number attribute reduction problem," *Neural Computing and Applications*, vol. 30, no. 8, pp. 2441-2452, Jan. 2018.
- [29] A. E. Hassanien, et al., "An improved moth flame optimization algorithm based on rough sets for tomato diseases detection," *Computers and Electronics in Agriculture*, vol. 136, pp. 86-96, Apr. 2017.
- [30] A. E. Hassanien, R. M. Rizk-Allah, and M. Elhoseny, "A hybrid crow search algorithm based on rough searching scheme for solving engineering optimization problems," *J. of Ambient Intelligence and Humanized Computing*, 25 pp., 2018.
- [31] S. Abdullah, L. Golafshan, and M. Z. A. Nazri, "Re-heat simulated annealing algorithm for rough set attribute reduction," *International J. of Physical Sciences*, vol. 6, no. 8, pp. 2083-2089, 2011.
- [32] M. Abdolrazzagh-Nezhad and S. Izadpanah, "A modified electromagnetic-like mechanism for rough set attribute reduction," in *Proc. Int. Conf. on Information and Software Technologies, ICIST'16*, pp. 356-368, Sept. 2016.
- [33] A. R. Hedar, J. Wang, and M. Fukushima, "Tabu search for attribute reduction in rough set theory," *Soft Computing*, vol. 12, no. 9, pp. 909-918, Nov. 2008.
- [34] Y. Chen, D. Miao, and R. Wang, "A rough set approach to feature selection based on ant colony optimization," *Pattern Recognition Letters*, vol. 31, no. 3, pp. 226-233, Feb. 2010.

**مجید عبدالرزاق نژاد** در سال ۱۳۸۲ مدرک کارشناسی ریاضیات کاربردی خود را از دانشگاه بیرجند و در سال ۱۳۸۴ مدرک کارشناسی ارشد ریاضیات کاربردی خود را از دانشگاه سیستان و بلوچستان با کار بر روی حوزه محاسبات نرم و شبکه‌های عصبی دریافت نمود. از سال ۱۳۸۴ الی ۱۳۸۶ نام‌برده به عنوان مربی در دانشگاه صنایع و معادن شهرستان بیرجند و پس از آن تا سال ۱۳۸۸ در جهاد دانشگاهی مشهد بعنوان مدرس مدعو خدمت کرده و پس از آن به دوره دکترای علوم کامپیوتر گرایش محاسبات هوشمند در دانشگاه ملی مالزی وارد گردید. در سال ۱۳۹۲ موفق به اخذ درجه دکترا در علوم کامپیوتر از دانشگاه مذکور گردید. دکتر عبدالرزاق نژاد از سال ۱۳۹۲ در دانشکده فنی مهندسی قائن دانشگاه بیرجند به عضویت هیأت علمی درآمد و پس از استقلال این دانشکده از دانشگاه بیرجند تحت عنوان دانشگاه بزرگمهر قائنات تا کنون به‌عنوان عضو هیأت علمی دانشکده فنی و مهندسی این دانشگاه مشغول به فعالیت می‌باشد. او مسئولیت‌هایی همچون رئیس دانشکده علوم پایه، سرپرست معاونت دانشجویی-فرهنگی و مدیر گروه مهندسی کامپیوتر دانشگاه بزرگمهر قائنات را بر عهده داشته است. زمینه‌های علمی مورد علاقه نام‌برده، انواع مسائل بهینه، مسائل زمانبندی، سیستم‌های نادقیق و منطق فازی، شبکه‌های عصبی، داده‌کاوی و کاربردهای آن، الگوریتم‌های فوق ابتکاری و ابرابتکاری می‌باشد.

**علی ادیبیان** در سال ۹۶ مدرک کارشناسی ارشد مهندسی کامپیوتر (گرایش فزار) از دانشگاه علوم تحقیقات (واحد تهران) و مدرک کارشناسی مهندسی کامپیوتر (گرایش نرم‌افزار) در سال ۱۳۹۲ از دانشگاه آزاد بیرجند دریافت نمود. حد فاصل سال‌های ۹۷ تا ۹۹ به‌عنوان عضو فعال باشگاه پژوهشگران جوان و نخبگان استان خراسان جنوبی فعالیت داشته است. در حال حاضر مشغول کار در حوزه برنامه‌نویسی بر روی پروژه‌های علوم داده و امنیت شبکه می‌باشد. ایشان مسلط به زبان‌های برنامه‌نویسی Matlab و Kotlin است و زمینه‌های علمی مورد علاقه نام‌برده شامل موضوعاتی مانند محاسبات نرم، داده‌کاوی و تشخیص الگو می‌باشد.