

# ارائه روش جهش هوشمند مبتنی بر الگوریتم PSO برای حل مسئله انتخاب ویژگی

محمود پرنده، مینا زلفی لیقوان و جعفر تنها

یادگیری می‌شود. برای کاهش ابعاد داده از الگوریتم‌های انتخاب ویژگی استفاده می‌گردد. هدف الگوریتم‌های انتخاب ویژگی، حذف ویژگی‌های تکراری یا اضافی است. منظور از ویژگی‌های تکراری، ویژگی‌هایی است که در سایر ویژگی‌های داده اثر خود را نمایان ساخته‌اند و منظور از ویژگی‌های اضافی، ویژگی‌هایی هستند که در دقت الگوریتم یادگیری ماشین هیچ تأثیری ندارند [۲]. انتخاب ویژگی معمولاً باعث بهبود دقت الگوریتم یادگیری شده و منجر به ساخت مدلی ساده‌تر می‌گردد.

یکی از جنبه‌های مهم الگوریتم‌های انتخاب ویژگی، استفاده از معیارهای ارزیابی است که کیفیت ویژگی‌های انتخاب‌شده را بررسی نماید. به صورت کلی، ۳ نوع روش انتخاب ویژگی وجود دارد که شامل روش Filter [۳]، روش Wrapper [۴] و روش Embedded [۵] است. در روش‌های مبتنی بر Filter از معیارهای از پیش تعریف شده که مستقل از فاز یادگیری هستند، استفاده می‌شود. به همین دلیل هیچ تضمینی وجود ندارد که ویژگی‌های انتخاب‌شده در بهبود الگوریتم یادگیری تأثیرگذار باشند. با این که روش‌های مبتنی بر Filter کم‌هزینه هستند، اما در آنها به دلیل عدم استفاده از الگوریتم‌های یادگیری، احتمال انتخاب ویژگی‌های نامرتب با کلاس داده افزایش می‌یابد [۶]. روش‌های مبتنی بر Wrapper از روش‌های مبتنی بر Filter عملکرد بهتری دارند، اما از لحاظ محاسباتی پیچیده‌تر هستند. در این روش‌ها از الگوریتم‌های یادگیری متنوعی برای ارزیابی ویژگی‌های انتخاب‌شده استفاده می‌شود [۷]. سومین گروه از روش‌های انتخاب ویژگی، روش‌های مبتنی بر Embedded هستند که در آنها از ترکیب روش‌های مبتنی بر Filter و Wrapper استفاده شده است [۸].

اخیراً روش‌های الگوریتم‌های تکاملی مبتنی بر Wrapper مانند الگوریتم‌های ژنتیک [۹] و PSO [۱۰] در انتخاب ویژگی به دلیل قدرت بهینه‌سازی سراسری خود مورد توجه قرار گرفته‌اند. در این روش‌ها هیچ اطلاعاتی از فضای مسئله مورد نیاز نیست و به صورت فراکتشافی<sup>۳</sup> اقدام به حل مسئله می‌کنند. این روش‌ها به دو دسته تک‌هدفه یا چندهدفه تقسیم می‌شوند. از آنجایی که مسئله انتخاب ویژگی یک مسئله چندهدفه است، از الگوریتم‌های چندهدفه برای حل مسئله انتخاب ویژگی استفاده می‌شود که یکی از اهداف، کاهش تعداد ویژگی و هدف دیگر افزایش دقت الگوریتم یادگیری است [۱۱].

در این مقاله، از روش‌های مبتنی بر Wrapper به کمک الگوریتم PSO استفاده شده و همان طور که پیشتر اشاره شد، مسئله انتخاب ویژگی یک مسئله چندهدفه است. برای حل این گونه مسائل دو روش وجود دارد:

چکیده: امروزه با افزایش حجم تولید داده، توجه به الگوریتم‌های یادگیری ماشین جهت استخراج دانش از داده‌های خام افزایش یافته است. داده خام معمولاً دارای ویژگی‌های اضافی یا تکراری است که بر روی عملکرد الگوریتم‌های یادگیری تأثیر می‌گذارد. جهت افزایش کارایی و کاهش هزینه محاسباتی الگوریتم‌های یادگیری ماشین، از الگوریتم‌های انتخاب ویژگی استفاده می‌شود که روش‌های متنوعی برای انتخاب ویژگی ارائه شده است. از جمله روش‌های انتخاب ویژگی، الگوریتم‌های تکاملی هستند که به دلیل قدرت بهینه‌سازی سراسری خود مورد توجه قرار گرفته‌اند. الگوریتم‌های تکاملی بسیاری برای حل مسئله انتخاب ویژگی ارائه شده که بیشتر آنها روی فضای هدف تمرکز داشته‌اند. فضای مسئله نیز می‌تواند اطلاعات مهمی برای حل مسئله انتخاب ویژگی ارائه دهد. از آنجایی که الگوریتم‌های تکاملی از مشکل عدم خروج از نقطه بهینه محلی رنج می‌برند، ارائه یک مکانیزم مؤثر برای خروج از نقطه بهینه محلی ضروری است. در این مقاله از الگوریتم تکاملی PSO با تابع چندهدفه برای انتخاب ویژگی استفاده شده که در آن یک روش جدید جهش که از امتیاز ویژگی‌های ذرات استفاده می‌کند، به همراه نخبه‌گرایی برای خروج از نقاط بهینه محلی ارائه گردیده است. الگوریتم ارائه‌شده بر روی مجموعه داده‌های مختلف تست و با الگوریتم‌های موجود بررسی شده است. نتایج شبیه‌سازی‌ها نشان می‌دهند که روش پیشنهادی در مقایسه با روش جدید RFPFSOFS بهبود خطای ۲۰٪، ۱۱٪، ۸۵٪ و ۷٪ به ترتیب در دیتاست‌های Madelon، Musk و Isolet و Arrhythmia دارد.

کلیدواژه: انتخاب ویژگی، بهینه‌سازی چندهدفه، الگوریتم PSO، مجموع وزن دار تطبیق پذیر، جهش هوشمند، نخبه‌گرایی.

## ۱- مقدمه

امروزه با پیشرفت تکنولوژی، حجم تولید داده افزایش یافته است. بیشتر داده‌های تولیدشده دارای ابعاد بسیاری هستند که این مشکل با نام "نفرین ابعاد" شناخته می‌شود [۱]. از آنجایی که برای تبدیل داده به دانش از الگوریتم‌های یادگیری ماشین استفاده می‌شود، هرچه ابعاد داده کمتر باشد، باعث افزایش کارایی و کاهش هزینه محاسباتی الگوریتم

این مقاله در تاریخ ۱۵ مهر ماه ۱۴۰۰ دریافت و در تاریخ ۳۱ فروردین ماه ۱۴۰۱ بازنگری شد.

محمود پرنده، دانشکده مهندسی برق و کامپیوتر، دانشگاه تبریز، تبریز، ایران، (email: parandeh@tabrizu.ac.ir)

مینا زلفی لیقوان (نویسنده مسئول)، دانشکده مهندسی برق و کامپیوتر، دانشگاه تبریز، تبریز، ایران، (email: mzolfy@tabrizu.ac.ir)

جعفر تنها، دانشکده مهندسی برق و کامپیوتر، دانشگاه تبریز، تبریز، ایران، (email: tanha@tabrizu.ac.ir)

2. Particle Swarm Optimization

3. Meta-Heuristics

1. Curse of Deminisionality

$$x_{is}^t = x_{is}^{t-1} + v_{is}^t \quad (2)$$

که در آن  $t$  شماره تکرار،  $i$  شماره ذره،  $s$  بعد ذره و  $w$  ضریب اینرسی است که سرعت ذره را کنترل می‌کند و هرچه مقدار آن بیشتر باشد، سرعت ذره افزایش یافته و از بهینه محلی خارج می‌شود. مقادیر  $c_1$  و  $c_2$  برای کنترل این است که ذره به سمت بهینه محلی خود یا به سمت بهینه سراسری حرکت کند. مقادیر  $b_1$  و  $b_2$  نیز اعداد تصادفی با توزیع نرمال بین ۰ و ۱ هستند. الگوریتم PSO دارای نسخه‌های مختلفی است که پایه آنها مشابه یکدیگر است [۱۴].

## ۲-۲ بهینه‌سازی چندهدفه

بسیاری از مسائل وجود دارند که دارای چند هدف برای بهینه‌سازی هستند. این اهداف معمولاً در تضاد با یکدیگرند و به همین دلیل بهینه‌سازی چندهدفه برای حل این گونه مسائل ارائه شده است [۱۵]. از آنجایی که این اهداف با هم در یک راستا نیستند، بایستی از روش‌های فشرده‌سازی یا Pareto Front استفاده نمود که در ادامه به توضیح این دو روش می‌پردازیم.

### ۲-۱-۲ روش فشرده‌سازی

این روش یکی از ساده‌ترین روش‌های حل مسئله چندهدفه است که در آن برای هر هدف یک وزن در نظر گرفته می‌شود. سپس وزن‌ها در اهداف ضرب شده و با یکدیگر جمع می‌شوند و هدف جدیدی را ایجاد می‌کنند. برای استفاده از این روش باید تمامی اهداف در یک راستا باشند (به صورت کمینه‌سازی یا بیشینه‌سازی). در ادامه، (۳) نشان‌دهنده نحوه محاسبه تابع هدف در روش فشرده‌سازی است

$$newfitness = sw_1 \times fitness_1 + sw_2 \times fitness_2 \quad (3)$$

که در آن  $sw_1$  و  $sw_2$  وزن‌های اهداف بوده که مجموع آنها برابر یک است.

### ۲-۲-۲ روش Pareto Front

در این روش جواب‌هایی انتخاب می‌شوند که سایر جواب‌ها نتوانند بر آن غلبه کنند. فرض کنید مسئله دارای دو هدف  $g$  و  $f$  بوده که در آن  $g$  در راستای بیشینه‌سازی و  $f$  در راستای کمینه‌سازی است. برای این که یک جواب در Pareto Front قرار بگیرد، باید ۲ شرط زیر هم‌زمان در آن صادق باشد:

(۱) جوابی غالب است که هر دو تابع هدف  $g$  و  $f$  در آن بهتر یا برابر جواب‌های دیگر باشد.

(۲) جوابی غالب است که حتماً یکی از توابع  $g$  و  $f$  در آن بهتر از سایر جواب‌ها باشد.

و جواب‌هایی که شرایط بالا را داشته باشند در Pareto Front قرار می‌گیرند [۱۶].

### ۲-۳-۲ روش Wrapper

روش‌های مبتنی بر Wrapper برای ارزیابی ویژگی‌های انتخاب‌شده در الگوریتم‌های یادگیری ماشین استفاده می‌شوند. این روش‌ها دارای دو جزء الگوریتم یادگیری و استراتژی جستجو هستند. الگوریتم Wrapper دارای دو نوع استراتژی جستجو است که نوع اول آن انتخاب ترتیبی و نوع دوم آن الگوریتم‌های تکاملی هستند که بر اساس الگوریتم یادگیری استفاده‌شده، ویژگی‌های انتخاب‌شده ارزیابی می‌شوند [۱۷].

(۱) روش‌های فشرده‌سازی که در آن توابع هدف با هم ترکیب شده و یک تابع هدف جدید تولید می‌کنند و (۲) روش Pareto Front است که در آن هدف، یافتن یک مجموعه از جواب‌هایی است که دیگر جواب‌ها نتوانند بر آن غلبه کنند [۱۲]. در این مقاله از روش مجموع وزن‌دار تطبیق‌پذیر<sup>۱</sup> برای حل مسئله چندهدفه استفاده شده است. در این روش، در گام‌های ابتدایی، وزن دقت الگوریتم بیشتر است و سپس با افزایش گام‌های اجرای الگوریتم، وزن تعداد ویژگی‌ها افزایش یافته تا الگوریتم بتواند در کنار دقت بالا، ویژگی‌های کمتری را انتخاب کند. در انتهای گام‌های اجرای الگوریتم، مجدد وزن دقت الگوریتم افزایش یافته تا خطای الگوریتم کاهش یابد. همچنین از مکانیزم آرشو<sup>۲</sup> برای نگهداری بهترین جواب‌های یافت‌شده تا کنون استفاده می‌گردد که برای رتبه‌دهی ویژگی‌ها کاربرد دارد. علاوه بر این از یک روش چشم جدید به همراه نخبه‌گرایی با استفاده از رتبه‌های ویژگی در روش پیشنهادی استفاده شده است. نوآوری روش پیشنهادی به شرح زیر است:

- استفاده از روش مجموع وزن‌دار تطبیق‌پذیر
- استفاده از روش چشم هوشمند به همراه نخبه‌گرایی برای خروج از نقاط بهینه محلی

در ادامه مقاله و در بخش دوم به بررسی کارهای مرتبط و پیشینه الگوریتم PSO خواهیم پرداخت. در بخش سوم روش پیشنهادی توضیح داده خواهد شد. در بخش چهارم مجموعه داده و نتایج شبیه‌سازی آمده و در انتها، بخش پنجم شامل نتیجه‌گیری است.

## ۲- پیشینه و کارهای مرتبط

در این قسمت به بررسی پیشینه الگوریتم PSO، روش‌های مبتنی بر Wrapper و تعریف مسئله چندهدفه می‌پردازیم و سپس کارهای مرتبط با روش پیشنهادی بررسی خواهد شد.

### ۲-۱ الگوریتم PSO

PSO مبتنی بر روش‌های فرااکتشافی است که حرکت دسته‌ای از پرندگان را شبیه‌سازی می‌کند [۱۳]. هر ذره<sup>۳</sup> در این دسته<sup>۴</sup> نشان‌دهنده یک جواب است که در فضای چندبعدی حرکت می‌کند و از بهترین موقعیت خود و همچنین بهترین موقعیت کل دسته برای بهبود موقعیت خود استفاده می‌نماید. عملکرد هر ذره بر اساس تابع برازش از پیش تعریف شده محاسبه می‌گردد.

فرضاً فضای ما دارای  $D$  بعد است و تعداد  $m$  ذره در دسته وجود دارد. هر ذره یک موقعیت به صورت  $X_i = [x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD}]$  و یک سرعت به صورت  $V_i = [v_{i1}, v_{i2}, v_{i3}, \dots, v_{iD}]$  دارد که  $i = 1, 2, 3, \dots, m$  است. در الگوریتم PSO هر ذره در راستای بهترین موقعیت خود که به صورت  $Pbest_i = [Pbest_{i1}, Pbest_{i2}, \dots, Pbest_{iD}]$  و بهترین موقعیت دسته که به صورت  $Gbest = [Gbest_1, Gbest_2, \dots, Gbest_D]$  است، حرکت می‌کند. سرعت و مکان جدید هر ذره مطابق با روابط زیر محاسبه می‌شوند

$$v_{is}^t = wv_{is}^{t-1} + c_1 b_1 (pbest_{is}^{t-1} - x_{is}^{t-1}) + c_2 b_2 (gbest_s^{t-1} - x_{is}^{t-1}) \quad (1)$$

1. Adaptive Weighted Sum
2. Archive
3. Particle
4. Swarm

### ۲-۳-۱ روش انتخابی

بالایی هستند. برای ساخت یک جواب دیگر، این ویژگی‌ها بر اساس دقت الگوریتم مرتب شده و از آنها برای ایجاد جواب جدید استفاده می‌گردد. در روش حذف، الگوریتم شروع به حذف ویژگی از جواب ایجاد شده می‌کند و این کار را تا جایی ادامه می‌دهد که جواب ایجاد شده، دقت الگوریتم را افزایش دهد. در روش مبادله، برخلاف دو روش قبل که اقدام به حذف و اضافه کردن می‌کرد، اقدام به مبادله ویژگی‌ها در جواب ایجاد شده با ویژگی‌های دارای رتبه بالا در آرشیو می‌شود. یکی دیگر از روش‌های جدید برای انتخاب ویژگی <sup>۱۱</sup> RFPFSOFS [۱۰] نام دارد که در آن از آرشیو برای ذخیره کردن بهترین جواب‌ها استفاده شده و امتیاز هر ویژگی بر اساس تکرار آن ویژگی در آرشیو محاسبه می‌گردد. در این روش نیز ذرات به ۳ دسته تقسیم شده و بر روی دو دسته از آنها عملگر جهش به دو صورت یکنواخت و غیر یکنواخت اعمال می‌شود. در ادامه به توضیح روش پیشنهادی می‌پردازیم.

### ۳- روش پیشنهادی

همان طور که بحث شد، انتخاب ویژگی یک مسئله چندهدفه است که در آن هدف، کاهش تعداد ویژگی‌های انتخاب شده و نیز افزایش دقت الگوریتم یادگیری است. در روش پیشنهادی با نام <sup>۱۲</sup> AWMOPFSOFS از فشرده‌سازی برای تبدیل تابع چندهدفه به تک‌هدفه استفاده شده است. همچنین به دلیل سادگی و هزینه محاسباتی پایین الگوریتم PSO، این الگوریتم به عنوان الگوریتم تکاملی مبتنی بر Wrapper انتخاب شده است. در روش پیشنهادی نیز از آرشیو به عنوان فضایی جهت ذخیره‌سازی بهترین جواب‌هایی که تا کنون الگوریتم یافته است، استفاده شده است. امتیازدهی به ویژگی‌های موجود در آرشیو نیز از فرایندهای مهم در روش پیشنهادی است. در ادامه، گام‌های روش پیشنهادی به صورت کلی در شکل‌های ۱ و ۲ آمده و نوآوری‌های روش پیشنهادی در بخش‌های ۳-۶ تا ۳-۸ به تفصیل توضیح داده شده است.

### ۳-۱ کدگذاری ذرات

در روش پیشنهادی، ذرات که نمایانگر ویژگی‌های مسئله هستند به صورت (۴) کدگذاری شده‌اند

$$X_i(t) = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD}) \quad (4)$$

$$, x_{ij} \in [0, 1], j = 1, 2, 3, \dots, D$$

که در آن  $X_i(t)$  ذره در تکرار  $t$  و ویژگی  $j$  ام ذره  $i$  ام که مقدار آن بین صفر و یک است، می‌باشد و  $D$  تعداد ویژگی‌های مسئله است. از آنجایی که در مسئله انتخاب ویژگی نیاز به کدگذاری به صورت صفر و یک داریم، از یک مقدار آستانه با نام  $\theta$  برای این کار استفاده می‌کنیم. مقدار  $\theta$  با توجه به [۱۰] برابر  $0.6$  در نظر گرفته شده است. بنابراین  $x_{ij}$ ‌هایی که مقدارشان از  $\theta$  بزرگ‌تر باشد، برابر یک و در غیر این صورت برابر صفر در نظر گرفته خواهند شد. نسخه دودویی  $X_i(t)$  در (۵) آمده است

$$Z_i(t) = (z_{i1}, z_{i2}, z_{i3}, \dots, z_{iD}) \quad (5)$$

$$, z_{ij} \in \{0, 1\}, j = 1, 2, 3, \dots, D$$

که در آن  $Z_i(t)$  نسخه دودویی  $X_i(t)$ ،  $z_{ij}$  نشان‌دهنده انتخاب یا عدم انتخاب ویژگی  $j$  ام ذره  $i$  ام و  $D$  تعداد ویژگی‌های مسئله است.

روش‌های انتخابی <sup>۱</sup> ۳ نوع هستند: (۱) روش انتخاب رو به جلو، (۲) روش انتخاب رو به عقب و (۳) روش انتخاب دوطرفه. در روش انتخاب رو به جلو، ویژگی‌ها اضافه می‌شوند تا تابع هدف بهینه شود. برعکس، در روش انتخاب رو به عقب، ویژگی‌ها حذف می‌شوند. در روش انتخاب دوطرفه، هر دو حالت افزودن/حذف کردن ویژگی امکان‌پذیر است. این روش‌ها از مشکل "اثر تودرتو" <sup>۲</sup> رنج می‌برند. در این اثر در روش رو به جلو، ویژگی‌ای که انتخاب نشده است، دیگر شانس انتخاب مجدد ندارد. همچنین در روش رو به عقب، ویژگی‌ای که انتخاب شده دیگر کنار گذاشته نمی‌شود [۱۸]. برای رفع این مشکل دو روش <sup>۳</sup> SBFS و <sup>۴</sup> SFFS ارائه شده‌اند [۱۹].

### ۲-۳-۲ روش الگوریتم‌های تکاملی

الگوریتم‌های تکاملی سعی در بهینه‌سازی تابع هدف دارند، به دلیل این که به مسائل به صورت یک جعبه سیاه و ناشناخته نگاه می‌کنند، به راحتی برای مسائل مختلف قابل استفاده هستند و به شکل یک‌هدفه یا چندهدفه مورد استفاده قرار می‌گیرند. الگوریتم‌های تکاملی مبتنی بر روش Wrapper به دلیل درگیر شدن فرایند یادگیری، زمان بسیاری را صرف محاسبات و یافتن مجموعه‌های ویژگی می‌نمایند.

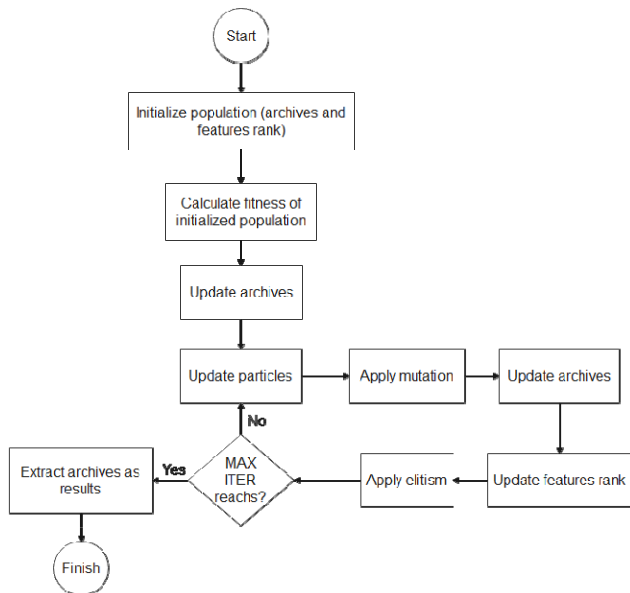
### ۲-۴ کارهای مرتبط

الگوریتم‌های بسیاری مبتنی بر روش Wrapper و با استفاده از الگوریتم‌های تکاملی ارائه گردیده است که در این قسمت به بررسی برخی از این روش‌ها می‌پردازیم. دو الگوریتم <sup>۵</sup> CMDPSOFS [۲۰] و <sup>۶</sup> HMDPSOFS [۲۱] برای انتخاب ویژگی ارائه شده‌اند. در روش CMDPSOFS از عملگر جهش به صورت یکنواخت و غیر یکنواخت استفاده شده و از این دو روش جهت حفظ تنوع ذرات و بهبود توانایی جستجوی الگوریتم استفاده گردیده است. برای رسیدن به این هدف مجموعه ذرات را به ۳ قسمت مساوی تقسیم کرده و اعمال جهش مربوط را بر روی ۲ قسمت از آن اعمال می‌کند. در روش HMDPSOFS از یک جهش ترکیبی استفاده شده که در روش اول آن در هر تکرار، ۱۰٪ از ذرات اجازه دارند که مقدار سرعت خود را با مقدار اولیه تنظیم نمایند. در روش دوم نیز از جهش پرشی <sup>۷</sup> استفاده شده که در آن ذرات با احتمال مشخص شده امکان جهش به صورت یکنواخت را دارند. این کار باعث افزایش قدرت جستجوی سراسری الگوریتم شده است. روش دیگر ISRPSO [۲۲] نام دارد که در آن از تکنیک‌های جستجوی محلی برای بهبود آرشیو (که در آن بهترین جواب‌ها نگهداری می‌شوند) استفاده می‌شود. در تکنیک جستجوی محلی از ۳ روش درج‌کردن <sup>۸</sup>، مبادله‌کردن <sup>۹</sup> مبادله‌کردن <sup>۹</sup> و حذف‌کردن <sup>۱۰</sup> برای بهبود آرشیو استفاده شده است. در روش درج‌کردن، ویژگی‌هایی که در آرشیو وجود دارند دارای اهمیت

1. Selection
2. Nested Effects
3. Sequential Backward Floating Selection
4. Sequential Forward Floating Selection
5. Crowding Mutation Dominance PSO Feature Selection
6. Hybrid Mutation Dominance PSO Feature Selection
7. Jumping Mutation
8. Inserting
9. Swapping
10. Removing

11. Ranked Feature PSO Feature Selection

12. Adaptive Weight Multi Objective PSO Feature Selection



شکل ۲: روندنمای روش پیشنهادی.

که اندازه آرشیو به مقدار بیشینه خود رسیده بود، هنگام به روز رسانی آرشیو، اعضای که مقدار تابع برازش آنها کمتر از کاندیدای جدید بود، با کاندیدای جدید عوض می‌شود. این فرایند باعث می‌شود که همواره بهترین جواب‌های الگوریتم در طول اجرا در آرشیو نگهداری شوند.

### ۳-۴ امتیازدهی به ویژگی‌ها

هدف اصلی انتخاب ویژگی، یافتن بهترین ویژگی‌ها و حذف ویژگی‌های اضافی است. برای رسیدن به این هدف، ویژگی‌ها بر اساس تعداد مشاهده در آرشیو امتیازدهی می‌شوند. دلیل این امر آن است که آرشیو، بهترین جواب‌ها را نگه می‌دارد. معادله (۹) نمایش‌دهنده نحوه محاسبه امتیاز ویژگی است

$$FR = \sum_{k=1}^{|A|} Z_k(t), Z_k(t) \in A \quad (9)$$

که  $FR$  یک بردار با  $D$  بعد و به صورت  $FR = [r_1, r_2, r_3, \dots, r_D]$  است که در آن مقدار امتیاز هر ویژگی می‌باشد.  $A$  نمایانگر آرشیو و  $Z_k(t)$  مقدار دودویی ویژگی در تکرار  $t$  است که در (۵) نشان داده شده بود. اگر ویژگی  $Z_k$  در هیچ یک از اعضای آرشیو انتخاب نشده باشد، مقدار آن برابر صفر خواهد بود و بیشینه مقدار امتیاز ویژگی‌ها برابر اندازه آرشیو است.

### ۳-۵ به روز رسانی موقعیت ذره

در مسئله انتخاب ویژگی، المان‌های ذرات همان ویژگی‌ها هستند که با تغییر مقادیر ویژگی ذرات حرکت می‌کنند. پس از به روز رسانی آرشیو، مجموعه ذرات دوباره با هم مقایسه می‌شوند. ذره‌ای که برازش بهتری داشت، به عنوان ذره برنده انتخاب شده و به گروه برنده‌ها تعلق می‌گیرد و ذره بازنده نیز به گروه بازنده‌ها اضافه می‌شود. ذراتی که در گروه برنده هستند، به طور مستقیم به تکرار بعدی منتقل می‌شوند و ذراتی که در گروه بازنده‌ها هستند به کمک (۱۰) تا (۱۲) [۲۴] به روز رسانی می‌شوند

$$\bar{x}^t = avg(x_w^t), x_w^t \in winner_{group} \quad (10)$$

$$v_i^{t+1} = R_1^t v_i^t + R_2^t (x_w^t - x_i^t) + \phi R_3^t (\bar{x}^t - x_i^t) \quad (11)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (12)$$

تقسیم داده‌های مجموعه داده به آموزشی و تست و سپس اعمال گام‌های زیر بر روی داده‌های آموزشی:

- گام ۱: مقداردهی اولیه جمعیت بر اساس قسمت ۳-۱
- گام ۲: ارزیابی مقدار برازش جمعیت بر اساس روش مجموع وزن‌دار تطبیق‌پذیر
- گام ۳: به روز رسانی آرشیو بر اساس مقدار برازش جمعیت
- گام ۴: به روز رسانی جمعیت بر اساس (۱۱) و (۱۲)
- گام ۵: اعمال جهش هوشمند و به روز رسانی ذرات
- گام ۶: به روز رسانی آرشیو پس از اعمال جهش هوشمند
- گام ۷: محاسبه و به روز رسانی امتیاز هر ذره بر اساس آرشیو
- گام ۸: اعمال عملگر نخبه‌گرایی بر روی جمعیت آرشیو
- گام ۹: اگر الگوریتم به بیشینه مقدار تکرار رسیده بود، به گام ۱۰ برو و در غیر این صورت به گام ۴ برگرد.
- گام ۱۰: استخراج اعضای موجود در آرشیو به عنوان بهترین جواب‌های روش پیشنهادی

شکل ۱: گام‌های الگوریتم پیشنهادی.

### ۳-۲ ارزیابی برازش ذرات

همان طور که پیشتر اشاره شد، در روش‌های مبتنی بر Wrapper، الگوریتم یادگیری نیز در انتخاب ویژگی دخالت دارد. به همین دلیل برای ارزیابی ویژگی‌های انتخاب‌شده از الگوریتم یادگیری KNN [۲۳] استفاده گردیده است. در این مقاله دو هدف تعداد ویژگی انتخاب‌شده و دقت الگوریتم به عنوان اهداف تابع برازش انتخاب شده‌اند که با هم در تضاد هستند. به همین دلیل به جای دقت الگوریتم از خطای الگوریتم استفاده می‌کنیم تا اهداف هر دو در یک راستا شوند. در (۶) نحوه محاسبه خطای الگوریتم آمده است

$$Error_{rate}(i) = 1 - \frac{FP + FN}{FP + FN + TP + TN} \quad (6)$$

که در آن  $TP$ ،  $TN$ ،  $FP$  و  $FN$  به ترتیب برابر مثبت واقعی، مثبت کاذب، منفی واقعی و منفی کاذب هستند. در (۷) نیز نحوه محاسبه تعداد ویژگی‌های انتخاب‌شده نشان داده شده است

$$Feature_{rate}(i) = \frac{f_i}{D} \quad (7)$$

$$f_i = \sum_{j=1}^D z_{ij}$$

که در آن  $f_i$  تعداد ویژگی‌های انتخاب‌شده و  $D$  تعداد کل ویژگی‌های مسئله است.

هر دو تابع هدف به صورت نرمال‌شده در نظر گرفته شده‌اند که مقدار هر کدام بین صفر و یک است. اکنون به کمک روش مجموع وزن‌دار تطبیق‌پذیر، تابع هدف نهایی به کمک (۸) محاسبه می‌شود

$$Fitness(i) = fw_1 \times Error_{rate}(i) + fw_2 \times Feature_{rate}(i) \quad (8)$$

که در آن  $fw_1$  و  $fw_2$  به ترتیب وزن‌های خطای الگوریتم و تعداد ویژگی‌های انتخاب‌شده هستند.

### ۳-۳ آرشیو

برای ذخیره‌کردن بهترین ذرات یافته‌شده در تکرارهای الگوریتم از مکانیزم آرشیو استفاده شده است. پس از هر تکرار، مقدار برازش هر ذره محاسبه می‌گردد و ذراتی که مقدار برازش آنها از بقیه ذرات بهتر باشد در آرشیو ذخیره می‌شوند. از آنجایی که اندازه آرشیو محدود است، در صورتی

**Input:** archive, features rank  
**Output:** updated archive  
**Begin**

$bestrank_{index} \leftarrow \text{select } fr_i \text{ where } fr_i \text{ greater than } \max(\frac{FR}{2}),$

$fr_i$  is feature rank of  $j$  th feature

$x'_{ij} \leftarrow \text{copy of } x_{ij}, x_{ij}$  is  $j$  th feature of particle  $i$  in archive

**for**  $i \leftarrow 1$  **to** size of archive **do**

**begin**

$ones_{index} \leftarrow x'_{ij}$  where  $x'_{ij}$  greater than  $\theta$

$ones_{length} \leftarrow \text{sum of } x'_{ij}$  where  $x'_{ij}$  greater than  $\theta$

$randoms_{ones} \leftarrow \text{select } \frac{1}{4}$  of  $ones_{index}$  randomly

**for**  $k \leftarrow 1$  **to** size of  $randoms_{ones}$  **do**

$x'_{i,randoms_{ones}(k)} \leftarrow 0$

**for**  $m \leftarrow 1$  **to** size of  $\frac{1}{4}ones_{index}$  **do**

$x'_{i,ones_{index}(m)} \leftarrow 1$

calculate fitness of  $x'_i$

**if** fitness of  $x'_i$  is better than  $x_i$  **then**

$x_i \leftarrow x'_i$

**end**

**end**

شکل ۵: روش نخبه‌گرایی هوشمند.

### ۳-۷ نخبه‌گرایی

ایده اصلی نخبه‌گرایی، ایجاد جواب بهتر در آرشیو می‌باشد و برای این منظور یک مکانیزم هوشمند ارائه شده است. در این روش ابتدا ویژگی‌ها بر اساس امتیازشان رتبه‌بندی می‌شوند، سپس بر اساس (۱۳) مجموعه‌ای از ویژگی‌های برتر انتخاب می‌گردند

$$BFR = FR(i), FR(i) \geq \max(\frac{FR}{\gamma}) \quad (13)$$

که  $BFR$  لیست بهترین ویژگی‌ها و  $FR(i)$  مقدار امتیاز ویژگی  $i$  ام است. سپس این ویژگی‌ها به صورت تصادفی با یک‌چهارم ویژگی‌هایی که مقدار آنها برابر یک است، جایگزین می‌شوند. در صورتی که مقدار برازش ذره جدید ایجاد شده از ذره قبلی بهتر بود، با آن ذره تعویض می‌شود. این فرایند بر روی تمامی اعضای آرشیو اجرا می‌شود. شبه‌کد نخبه‌گرایی در شکل ۵ آمده است.

### ۳-۸ مجموع وزن دار تطبیق‌پذیر

همان طور که پیشتر اشاره شد، از روش‌های حل مسائل بهینه‌سازی چندهدفه، استفاده از روش‌های مبتنی بر فشرده‌سازی است و در این مقاله از روش مجموع وزن دار تطبیق‌پذیر استفاده شده است. در این روش ابتدا وزن‌های خطای الگوریتم و تعداد ویژگی‌های انتخاب شده که به ترتیب برابر  $f_{w1}$  و  $f_{w2}$  هستند، مقداردهی اولیه می‌شوند. سپس از گام ۳۰ به بعد با افزایش مضرب ۱۰ از گام‌های تکرار الگوریتم، از اهمیت وزن  $f_{w1}$  کم شده و به وزن  $f_{w2}$  اضافه می‌شود. این فرایند تا گام ۷۰ ادامه می‌یابد. سپس از گام ۷۰ به بعد با افزایش مضرب ۱۰ از گام‌های تکرار به اهمیت وزن  $f_{w1}$  افزوده شده و از اهمیت  $f_{w2}$  کاسته می‌شود. در ابتدای الگوریتم، اهمیت خطا بیشتر از تعداد ویژگی‌های انتخاب شده است. این کار باعث می‌شود که الگوریتم سعی در یافتن جواب‌هایی نماید که خطای کمتری دارند. سپس با افزایش گام تکرار، الگوریتم سعی در یافتن

**Input:** population, archive, features rank  
**Output:** updated population  
**Begin**

**for**  $i \leftarrow 1$  **to**  $\frac{N}{3}$ ,  $N$  is size of population **do**

**if** generated uniform random less than  $p_{uniform}$  **then**

**for**  $j \leftarrow 1$  **to**  $D$ ,  $D$  is total number of features **do**

**if**  $x_{ij}$  is less than  $\theta$  **then**

$x_{ij} \leftarrow x_{ij} + \theta$

**else**

$x_{ij} \leftarrow x_{ij} - \theta$

**end**

شکل ۳: روش جهش هوشمند یکنواخت. (endهای اضافی حذف شد)

**Input:** population, archive, features rank  
**Output:** updated population  
**Begin**

**for**  $i \leftarrow 1$  **to**  $\frac{N}{3}$ ,  $N$  is size of population **do**

**if** generated uniform random less than  $p_{non-uniform}$  **then**

**for**  $j \leftarrow 1$  **to**  $D$ ,  $D$  is total number of features **do**

**if** generated uniform random less than 0.5 **then**

**if**  $x_{ij}$  is less than  $\theta$  and  $featurerank_j$  less than  $\mu$  **then**

$x_{ij} \leftarrow x_{ij} + \theta$

**else**

$x_{ij} \leftarrow x_{ij} - \theta$

**end**

شکل ۴: روش جهش هوشمند غیر یکنواخت.

که در آن  $\bar{x}^t$  میانگین ذرات موجود در گروه برنده در تکرار  $t$ ,  $R^t$ ,  $R^t$  و  $R^t$  اعداد تصادفی بین صفر و یک و  $\phi$  یک ضریب از پیش تعریف شده است.  $x_w^t$  موقعیت ذره برنده در تکرار  $t$ ,  $x_i^t$  موقعیت ذره بازنده و  $v_i^t$  نیز سرعت ذره بازنده در تکرار  $t$  در نظر گرفته شده است.

### ۳-۶ جهش هوشمند

در گام قبل ممکن است که ذرات پس از چندین تکرار در نقطه بهینه محلی گرفتار شوند و به همین دلیل نیاز به مکانیزمی است تا ذرات بتوانند از نقطه بهینه محلی خارج شوند. برای حل این مشکل مکانیزم جهش هوشمند ارائه گردیده است. در این روش، ابتدا جمعیت به سه دسته مساوی تقسیم می‌شود که دلیل این امر، حفظ جستجوی محلی در کنار جستجوی جهانی فضای حالت است. دسته اول بدون تغییر به نسل بعدی منتقل می‌شوند. در دسته دوم، یک عدد تصادفی تولید شده و اگر از یک مقدار پیش فرض  $p_{uniform}$  کمتر باشد، مقادیر ویژگی‌ها عوض می‌شوند. به این صورت که در آن اگر ویژگی انتخاب شده بود، به انتخاب نشده تغییر می‌کند و اگر انتخاب نشده بود، به انتخاب شده تغییر خواهد کرد. این عمل برای تمامی ویژگی‌های ذره انجام خواهد گردید. در دسته سوم نیز یک عدد تصادفی تولید می‌شود و اگر از مقدار پیش فرض  $p_{non-uniform}$  کمتر بود، این ذره برای جهش انتخاب شده و برای هر المان ذره یک عدد تصادفی دیگر تولید می‌شود. اگر این عدد تصادفی از ۰/۵ کمتر بود، بر روی آن المان هیچ گونه جهشی انجام نمی‌شود و در غیر این صورت، اگر عدد تصادفی تولید شده بیشتر از ۰/۵ بود، بر روی آن المان در صورتی که امتیاز آن کمتر از مقدار پیش فرض  $\mu$  باشد، عمل جهش انجام می‌شود. شبه‌کد مربوط به دو روش جهش پیشنهادی در شکل‌های ۳ و ۴ آورده شده است.

جدول ۱: توضیحات مجموعه داده.

نام مجموعه داده	تعداد ویژگی‌ها	تعداد داده‌ها	ویژگی نامشخص	تعداد کلاس
Isolet	۶۱۷	۷۷۹۷	خیر	۲۶
Musk	۱۶۷	۶۵۹۸	خیر	۲
Madelon	۵۰۰	۴۴۰۰	نامشخص	۲
Arrhythmia	۲۷۹	۴۲۵	بله	۱۶
Cnae	۸۵۷	۱۰۸۰	نامشخص	۹

جدول ۲: آزمایش یافتن مقدار بهینه  $P_{uniform}$ .

مقدار $P_{uniform}$	بهترین جواب ۱		بهترین جواب ۲		بهترین جواب ۳	
	تعداد	خطا	تعداد	خطا	تعداد	خطا
۰٫۱۰	۷	۰٫۳۴	۶	۰٫۳۶	۵	۰٫۳۷
۰٫۱۵	۱۰	۰٫۳۳	۶	۰٫۳۵	۷	۰٫۳۴
۰٫۲۰	۶	۰٫۳۵	۱۲	۰٫۳۵	۴	۰٫۳۷
۰٫۲۵	۶	۰٫۳۱	۸	۰٫۳۰	۶	۰٫۳۳
۰٫۳۰	۹	۰٫۳۹	۹	۰٫۲۹	۸	۰٫۳۱
۰٫۳۵	۵	۰٫۲۹	۵	۰٫۳۰	۵	۰٫۳۰
۰٫۴۰	۵	۰٫۳۲	۶	۰٫۳۳	۶	۰٫۳۳
۰٫۴۵	۵	۰٫۳۰	۵	۰٫۳۰	۷	۰٫۳۰
۰٫۵۰	۵	۰٫۳۱	۵	۰٫۳۱	۶	۰٫۳۱

جدول ۳: آزمایش یافتن مقدار بهینه  $P_{non-uniform}$ .

مقدار $P_{non-uniform}$	بهترین جواب ۱		بهترین جواب ۲		بهترین جواب ۳	
	تعداد	خطا	تعداد	خطا	تعداد	خطا
۰٫۱۰	۸	۰٫۳۰	۸	۰٫۳۳	۷	۰٫۳۳
۰٫۱۵	۶	۰٫۳۱	۶	۰٫۳۲	۵	۰٫۳۳
۰٫۲۰	۸	۰٫۳۴	۶	۰٫۳۴	۵	۰٫۳۸
۰٫۲۵	۶	۰٫۳۳	۶	۰٫۳۳	۸	۰٫۳۲
۰٫۳۰	۷	۰٫۳۳	۷	۰٫۳۳	۸	۰٫۳۲
۰٫۳۵	۵	۰٫۳۳	۵	۰٫۳۳	۶	۰٫۳۲
۰٫۴۰	۶	۰٫۳۲	۶	۰٫۳۲	۷	۰٫۳۳
۰٫۴۵	۸	۰٫۳۱	۸	۰٫۳۱	۹	۰٫۳۰
۰٫۵۰	۵	۰٫۳۳	۵	۰٫۳۲	۵	۰٫۳۳

جدول ۴: مقداردهی پارامترهای الگوریتم پیشنهادی.

نام پارامتر	مقدار پارامتر
$\theta$	۰٫۶
$fiw1$	۰٫۷۵
$fiw2$	۰٫۲۵
$\phi$	۰٫۱
$\mu$	۰٫۳
$P_{uniform}$	۰٫۳۵
$P_{non-uniform}$	۰٫۴

برابر ۰٫۴ و ۰٫۳۵ انتخاب شده و سایر پارامترهای الگوریتم پیشنهادی در جدول ۴ آمده است.

در سایر روش‌های مبتنی بر PSO، سایز جمعیت برابر ۳۰، بیشینه مقدار تکرار برابر ۱۰۰، بیشینه مقدار سرعت یا همان  $v_{max}$  برابر ۰٫۶، مقادیر  $b1$  و  $b2$  برابر ۰٫۱۴۶، مقدار  $w$  برابر ۰٫۷۲۹ و مقدار  $\theta$  برابر ۰٫۶ در نظر گرفته شده است. پارامترهای مخصوص هر روش بر اساس گزارش‌های آنها تنظیم شده و احتمال پرش در روش HMPSOFS برابر ۰٫۱ است. در روش CMDPSOFS وزن اینرسی به صورت تصادفی بین ۰٫۱ و ۰٫۵، مقادیر  $b1$  و  $b2$  اعداد تصادفی بین ۱/۵ و ۲ و همچنین نرخ جهش در این روش برابر  $1/D$  که در آن  $D$  تعداد ویژگی‌های مجموعه داده می‌باشد، در نظر گرفته شده است. برای NSGAI نیز پارامترها مشابه روش CMDPSOFS و نرخ برش برابر ۰٫۹ است. لازم به ذکر است که در روش NSGAI کروموزوم‌ها به صورت یک آرایه  $d$  بعدی هستند که اگر مقدار آن برابر یک باشد به معنی انتخاب ویژگی و اگر مقدار آن برابر صفر باشد، به معنی عدم انتخاب ویژگی است [۲۵].

### ۳-۴ بررسی نتایج شبیه‌سازی

در روش پیشنهادی، از پنج مجموعه داده "Musk"، "Isole", "Madelon"، "Arrhythmia" و "Cnae" برای مقایسه استفاده گردیده و تمامی مجموعه داده‌های ذکر شده دارای بیش از ۱۰۰ ویژگی هستند. روش AWMOPSOFS با پنج روش که دو تای آنها روش‌های پایه NSGAI [۲۵] و MOPSO [۲۶] و سه تای آنها روش‌های مؤثر با نام‌های CMDPSOFS، HMPSOFS و RFPDOFS هستند، مقایسه شده است. در شکل‌های ۶ تا ۱۰ نتایج شبیه‌سازی آمده است. همان طور که در شکل‌ها مشخص است، روش پیشنهادی به دلیل استفاده از جهش هوشمند و نخبه‌گرایی هوشمند توانسته که تعداد ویژگی کمتری با در نظر

جواب‌هایی دارد که تعداد ویژگی‌های انتخاب شده آن کمتر باشند. این کار ممکن است باعث افزایش خطا شود. بنابراین در گام‌های انتهایی، مجدداً اهمیت خطا افزایش یافته تا جواب‌ها با خطای کمتر پیدا شوند. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی جواب‌های بهتری با در نظر گرفتن خطا و تعداد ویژگی‌های انتخابی پیدا می‌کند. در ادامه به بررسی نتایج شبیه‌سازی می‌پردازیم.

### ۴- نتایج شبیه‌سازی

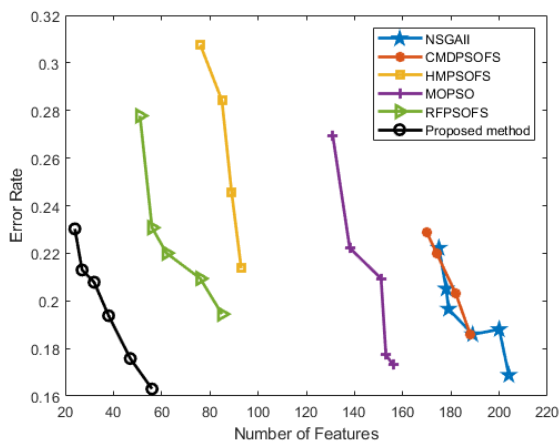
در این قسمت به تفصیل به بررسی نتایج شبیه‌سازی روش پیشنهادی AWMOPSOFS که شامل معرفی مجموعه داده، پارامترهای روش پیشنهادی و روش‌های دیگر و مقایسه نتایج است، می‌پردازیم.

#### ۴-۱ مجموعه داده

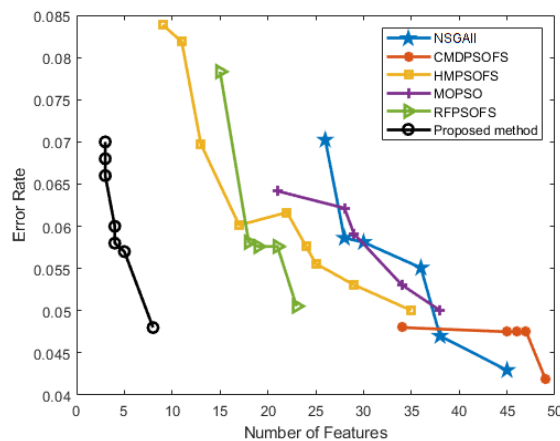
در این مقاله از مجموعه داده‌های عمومی UCI استفاده گردیده و خلاصه‌ای از ویژگی‌های این مجموعه داده‌ها در جدول ۱ آمده است. در این جدول تعداد ویژگی‌ها، تعداد داده‌ها، آیا داده‌ای دارای ویژگی با مقدار نامشخص است یا نه و تعداد کلاس هر مجموعه داده آورده شده است.

#### ۴-۲ پارامترها

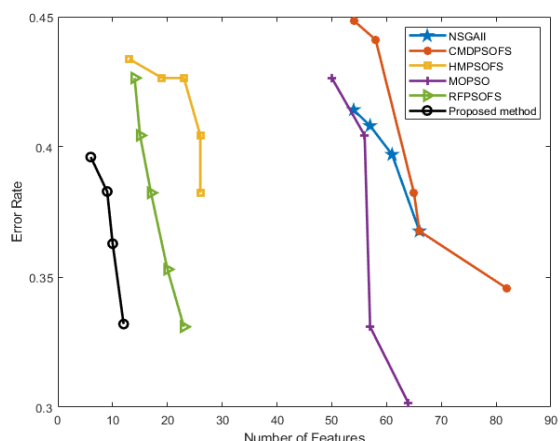
همان طور که پیشتر اشاره شد، ذرات در الگوریتم تکاملی PSO به صورت پیوسته در نظر گرفته شده‌اند. بنابراین نیاز به فرایندی است تا نقاط پیوسته به گسسته تبدیل شوند و برای این کار از پارامتر  $\theta$  استفاده شده بود که مقدار آن برابر ۰٫۶ است. از الگوریتم KNN با مقدار  $k$  برابر ۵ به عنوان الگوریتم یادگیری استفاده شده است. برای انتخاب مقادیر  $P_{uniform}$  و  $P_{non-uniform}$  روی مجموعه داده Arrhythmia آزمایش‌هایی با ۳۰ تکرار انجام شده که بهترین نتایج آن در جداول ۲ و ۳ آورده شده است. با توجه به این آزمایش‌ها، مقادیر  $P_{uniform}$  و  $P_{non-uniform}$  به ترتیب



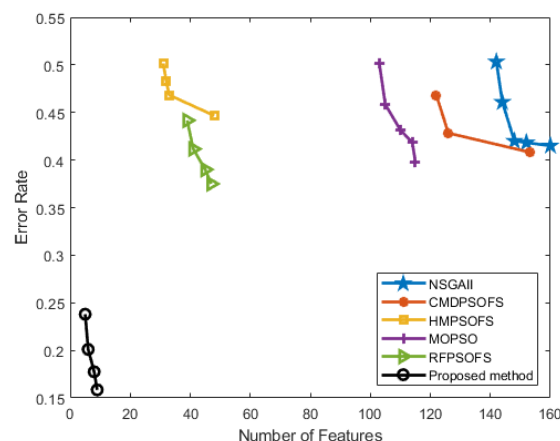
شکل ۹: نتایج شبیه‌سازی در مجموعه داده Isolte.



شکل ۶: نتایج شبیه‌سازی در مجموعه داده Musk.



شکل ۱۰: نتایج شبیه‌سازی در مجموعه داده Arrhythmia.

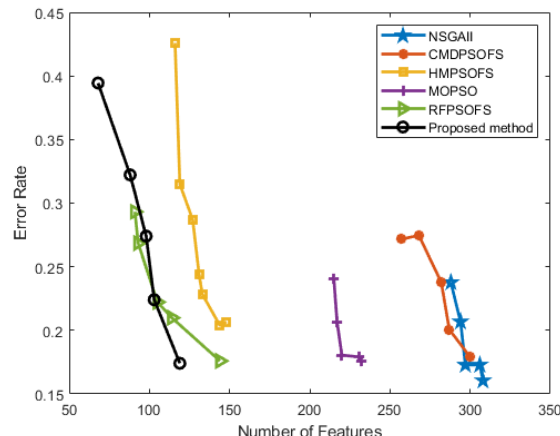


شکل ۷: نتایج شبیه‌سازی در مجموعه داده Madelon.

در جدول ۶ نیز بهترین تعداد ویژگی‌های انتخاب شده در ۳۰ تکرار اجرای هر الگوریتم آمده است. نتایج به دست آمده به دلیل شرایط مختلف و تصادفی هر الگوریتم در شروع متفاوت است. با توجه به جدول ۵، تعداد ویژگی‌های انتخاب شده توسط الگوریتم پیشنهادی در مجموعه داده "Cnae" بهتر از سایر روش‌ها در مجموعه داده‌های مختلف است. در جدول ۷ نیز شماره ویژگی‌های انتخاب شده در دو حالت بهترین تعداد ویژگی انتخاب شده و بهترین خطای الگوریتم آمده است. در جداول ۸ تا ۱۳ نیز میانگین، انحراف معیار و واریانس خطا و تعداد ویژگی روش پیشنهادی در ۳۰ بار تکرار آورده شده است. همچنین زمان اجرای الگوریتم‌ها بر روی دو مجموعه داده Isolet و Madelon بررسی گردید و نتایج به ثابته در جدول ۱۴ آورده شده است.

### ۵- نتیجه‌گیری

امروزه انتخاب ویژگی، نقش مهمی در الگوریتم‌های یادگیری ایفا می‌کند. بیشتر داده‌های جمع‌آوری شده دارای ویژگی‌های اضافی و تکراری هستند که باعث می‌شود الگوریتم یادگیری، مدل‌های پیچیده و پرهزینه‌ای ایجاد کند. برای رفع این مشکل، الگوریتم‌های انتخاب ویژگی بسیاری ارائه شده است اما اکثر این روش‌ها به صورت یک مسئله یک‌هدفه که همان انتخاب ویژگی کمتر است تمرکز کرده‌اند، در حالی که انتخاب ویژگی یک مسئله چندهدفه با اهداف متضاد است. همچنین به دلیل این که فضای مسئله بسیار بزرگ است، استفاده از برخی اطلاعات که در حین انتخاب ویژگی به دست می‌آید، می‌تواند به بهبود الگوریتم انتخاب ویژگی کمک کند. در این مقاله روش مجموع وزن‌دار تطبیق‌پذیر



شکل ۸: نتایج شبیه‌سازی در مجموعه داده Cnae.

گرفتن خطای الگوریتم انتخاب کند. با توجه به نتایج جدول ۵، روش پیشنهادی در مقایسه با سایر روش‌ها تعداد ویژگی کمتری انتخاب کرده است. در مجموعه داده‌های "Musk" و "Madelon" روش پیشنهادی تعداد ویژگی کمتر قابل توجهی به نسبت سایر روش‌ها انتخاب کرده است. با در نظر گرفتن تعداد ویژگی‌های انتخاب شده، خطای الگوریتم نیز به نسبت سایر روش‌ها کمتر است. با این که در برخی مجموعه داده‌ها خطای سایر الگوریتم‌ها مانند NSGAI و CMDPSOFS در مجموعه داده "Musk" بهتر از روش پیشنهادی است، اما تعداد ویژگی‌های انتخاب شده در آنها بیشتر می‌باشد. در برخی مجموعه داده‌ها مانند "Madelon" بیشینه تعداد ویژگی انتخاب شده توسط روش پیشنهادی کمتر از کمینه تعداد ویژگی انتخاب شده در سایر روش‌ها است.



جدول ۵: مقایسه تعداد ویژگی‌های انتخاب‌شده و خطای الگوریتم.

Cnae	Arrhythmia	Madelon	Musk	Isolet		
۰.۳۹	۰.۳۹۶	۰.۲۳۸	۰.۰۷	۰.۲۳	خطا	AWMOPSOFS
۶۸	۶	۵	۳	۲۴	تعداد ویژگی	
۰.۲۳۷	۰.۴۱۴	۰.۵۰۳	۰.۰۷	۰.۲۲	خطا	NSGAI
۲۸۸	۵۴	۱۴۲	۲۶	۱۷۵	تعداد ویژگی	
۰.۲۷۱	۰.۴۴۸	۰.۴۶۸	۰.۰۴۸	۰.۲۲۸	خطا	CMDPSOFS
۲۵۷	۵۴	۱۲۲	۳۴	۱۷۰	تعداد ویژگی	
۰.۴۲۵	۰.۴۳۳	۰.۵۰۱۶	۰.۰۸۳	۰.۳۰۷	خطا	HMPSOFS
۱۱۶	۱۳	۳۱	۹	۷۶	تعداد ویژگی	
۰.۲۴۰	۰.۴۲۶	۰.۵۰۱	۰.۰۶۴	۰.۲۶۹	خطا	MOPSO
۲۱۵	۵۰	۱۰۳	۲۱	۱۳۱	تعداد ویژگی	
۰.۲۹۳	۰.۴۲۶	۰.۴۴۱۶	۰.۰۷۸	۰.۲۷۷	خطا	RFPSOFS
۹۱	۱۴	۳۹	۱۵	۵۱	تعداد ویژگی	

جدول ۶: مقایسه تعداد ویژگی‌های انتخاب‌شده در روش‌های مختلف.

نام مجموعه داده	نام الگوریتم	تعداد ویژگی‌های انتخاب‌شده
Musk	AWMOPSOFS	۳، ۳، ۳، ۴، ۴، ۵، ۷
	NSGAI	۲۶، ۲۸، ۳۰، ۳۶، ۳۸، ۴۵
	CMDPSOFS	۳۴، ۴۵، ۴۶، ۴۷، ۴۹
	HMPSOFS	۹، ۱۱، ۱۳، ۱۷، ۲۲، ۲۴، ۲۵، ۲۹، ۳۵
	MOPSO	۲۱، ۲۸، ۲۹، ۳۴، ۳۸
Madelon	RFPSOFS	۱۵، ۱۸، ۱۹، ۲۱، ۲۳
	AWMOPSOFS	۵، ۶، ۸، ۹
	NSGAI	۱۴۲، ۱۴۴، ۱۴۸، ۱۵۲، ۱۶۰
	CMDPSOFS	۱۲۲، ۱۲۶، ۱۵۳
	HMPSOFS	۳۱، ۳۲، ۳۳، ۴۸
Cnae	MOPSO	۱۰۳، ۱۰۵، ۱۱۰، ۱۱۴، ۱۱۵
	RFPSOFS	۳۹، ۴۱، ۴۵، ۴۷
	AWMOPSOFS	۶۸، ۸۸، ۹۸، ۱۰۳، ۱۱۹
	NSGAI	۲۸۸، ۲۹۴، ۲۹۷، ۳۰۶، ۳۰۸
	CMDPSOFS	۲۵۷، ۲۶۸، ۲۸۲، ۲۸۷، ۳۰۰
Isolet	HMPSOFS	۱۱۶، ۱۱۹، ۱۲۷، ۱۳۱، ۱۳۳، ۱۴۴، ۱۴۸
	MOPSO	۲۱۵، ۲۱۷، ۲۲۰، ۲۳۱، ۲۳۲
	RFPSOFS	۹۱، ۹۳، ۱۰۵، ۱۱۴، ۱۴۴
	AWMOPSOFS	۲۴، ۲۷، ۳۲، ۳۸، ۴۷، ۵۶
	NSGAI	۱۷۵، ۱۷۸، ۱۷۹، ۱۸۹، ۲۰۰، ۲۰۴
Arrhythmia	CMDPSOFS	۱۷۰، ۱۷۴، ۱۸۲، ۱۸۸
	HMPSOFS	۷۶، ۸۵، ۸۹، ۹۳
	MOPSO	۱۳۱، ۱۳۸، ۱۵۱، ۱۵۳، ۱۵۶
	RFPSOFS	۵۱، ۵۶، ۶۲، ۷۶، ۸۵
	AWMOPSOFS	۶، ۹، ۱۰، ۱۲
Arrhythmia	NSGAI	۵۴، ۵۷، ۶۱، ۶۶
	CMDPSOFS	۵۴، ۵۸، ۶۵، ۶۶، ۸۲
	HMPSOFS	۱۳، ۱۹، ۲۳، ۲۶، ۲۶
	MOPSO	۵۰، ۵۶، ۵۷، ۶۴
	RFPSOFS	۱۴، ۱۵، ۱۷، ۲۰، ۲۳

که الگوریتم بتواند از نقاط بهینه محلی خارج شود و عمل نخبه‌گرایی هوشمند باعث بهبود جواب‌های موجود در آرشیو گردید.

روش پیشنهادی با تعدادی از بهترین روش‌های موجود مقایسه شد و نتایج شبیه‌سازی نشان داد که این روش در مقایسه با روش‌های پایه NSGAI و MOPSO و روش‌های مؤثر با نام‌های CMDPSOFS، HMPSOFS و RFPSOFS عملکرد بهتری دارد. نتایج شبیه‌سازی بر روی ۵ مجموعه داده که هر کدام دارای بیش از ۱۰۰ ویژگی بودند، اجرا گردید. الگوریتم AWMOPSOFS دارای خطای ۰.۰۴۸ و تعداد ویژگی انتخاب‌شده ۷ در مجموعه داده Musk بود و در مقایسه با روش RFPSOFS که روش جدیدی است بهتر عمل کرد. سایر نتایج الگوریتم AWMOPSOFS برای مجموعه داده‌های Cnae، Madelon و Isolet، Arrhythmia به ترتیب دارای خطا و تعداد ویژگی ۰.۱۷۴ و ۰.۱۱۹، ۰.۱۶۲ و ۰.۵۶ و ۰.۳۳۱ و ۱۲ بود.

### مراجع

- [1] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. on Evolutionary Computation*, vol. 20, no. 4, pp. 606-626, Aug. 2015.
- [2] J. Miao and L. Niu, "A survey on feature selection," *Procedia Computer Science*, vol. 91, pp. 919-926, 2016.
- [3] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. of Machine Learning Research*, vol. 3, pp. 1157-1182, 2003.
- [4] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273-324, Dec. 1997.
- [5] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16-28, Jan. 2014.
- [6] S. S. Darshan and C. Jaidhar, "Performance evaluation of filter-based feature selection techniques in classifying portable executable files," *Procedia Computer Science*, vol. 125, pp. 346-356, 2018.
- [7] S. Karasu, A. Altan, S. Bekiros, and W. Ahmad, "A new forecasting model with wrapper-based feature selection approach using multi-objective optimization technique for chaotic crude oil time series," *Energy*, vol. 212, Article ID: 118750, 2020.
- [8] H. Liu, M. Zhou, and Q. Liu, "An embedded feature selection method for imbalanced data classification," *IEEE/CAA J. of Automatica Sinica*, vol. 6, no. 3, pp. 703-715, May 2019.
- [9] R. Vijayanand, D. Devaraj, and B. Kannapiran, "Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection," *Computers & Security*, vol. 77, pp. 304-314, Aug. 2018.
- [10] M. Amoozegar and B. Minaei-Bidgoli, "Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism," *Expert Systems with Applications*, vol. 113, pp. 499-514, 15 Dec. 2018.

مبتنی بر الگوریتم PSO برای حل این مشکلات ارائه شده است. در روش پیشنهادی از مکانیزم آرشیو در کنار امتیازدهی به ویژگی‌ها برای جهش هوشمند و نخبه‌گرایی هوشمند استفاده گردید. جهش هوشمند باعث شد



جدول ۷: شماره ویژگی‌های انتخاب‌شده توسط AWMOPSOFS.

خطای الگوریتم	شماره ویژگی‌های انتخاب‌شده	تعداد ویژگی انتخاب‌شده	نام مجموعه داده
۰٫۰۵۴	۱۲۳، ۳۵، ۸	۳	Musk (بهترین تعداد ویژگی)
۰٫۰۴۸	۱۳۹، ۱۳۱، ۹۱، ۵، ۳، ۰	۷۷	Musk (بهترین خطا)
۰٫۲۰۹	۴۷۵، ۳۷۸، ۳۱۸، ۱۵۳، ۰	۵	Madelon (بهترین تعداد ویژگی)
۰٫۱۳۳۵	۴۷۵، ۴۵۵، ۴۵۱، ۴۳۳، ۴۸، ۲	۶	Madelon (بهترین خطا)
۰٫۲۱۶	۳۲۸، ۲۱۱، ۲۰۴، ۲۰۲، ۱۷۴، ۱۶۸، ۱۴۵، ۱۳۹، ۱۰۶، ۸۴، ۸۰، ۷۳، ۳۴، ۳۱، ۲۳، ۲۱، ۲۰، ۱۹، ۱۷، ۱۳، ۹، ۸، ۳، ۰، ۵۸۴، ۵۸۱، ۵۵۰، ۵۳۶، ۵۲۲، ۴۸۲، ۴۷۹، ۴۵۶، ۴۳۷، ۴۱۶، ۴۱۱، ۳۹۵، ۳۹۴، ۳۹۲، ۳۸۸، ۳۸۶، ۳۵۷، ۳۴۷، ۳۰۷، ۲۴۴، ۰، ۳، ۴، ۵، ۶، ۸، ۱۰، ۱۳، ۱۵، ۱۶، ۲۰، ۲۱، ۲۳، ۲۴، ۲۷، ۲۸، ۲۹، ۳۴، ۳۸، ۳۹، ۴۱، ۴۲، ۴۳، ۴۴، ۴۵، ۴۶، ۴۹، ۵۲، ۰، ۱۶۳	۲۴	Isolet (بهترین تعداد ویژگی)
۰٫۱۶۳	۴۲۶، ۴۱۷، ۴۱۱، ۳۹۳، ۳۶۳، ۳۵۷، ۲۰۳، ۱۹۶، ۱۸۸، ۱۷۶، ۱۶۶، ۱۳۱، ۱۱۴، ۱۰۹، ۱۰۷، ۱۰۳، ۱۰۰، ۸۵، ۷۱، ۵۸، ۵۸۰، ۵۶۶، ۵۱۵، ۵۰۹، ۵۰۰، ۴۹۵، ۴۸۸، ۴۵۲	۵۶	Isolet (بهترین خطا)
۰٫۳۲۹	۶۴، ۶۲، ۵۸، ۵۷، ۵۱، ۴۵، ۴۴، ۳۸، ۳۵، ۳۲، ۲۹، ۲۷، ۲۴، ۲۳، ۲۲، ۲۰، ۱۹، ۱۸، ۱۶، ۱۳، ۱۱، ۸، ۷، ۶، ۴، ۳، ۲، ۱، ۰، ۵۷۰، ۵۴۵، ۴۹۴، ۴۸۳، ۴۲۰، ۳۶۶، ۳۳۷، ۳۳۱، ۲۸۲، ۲۴۸، ۲۱۵، ۲۱۰، ۲۰۱، ۱۹۰، ۱۶۹، ۱۵۶، ۱۵۴، ۱۹، ۷۶، ۷۲، ۷۰، ۶۱، ۶	۵۶	Cnae (بهترین تعداد ویژگی)
۰٫۱۷۴	۸۳۱، ۸۱۸، ۷۹۰، ۷۲۵، ۷۰۶، ۶۱۴، ۰، ۴، ۵، ۶، ۸، ۹، ۱۰، ۱۱، ۱۲، ۱۴، ۱۵، ۲۱، ۲۲، ۲۴، ۲۵، ۲۶، ۲۷، ۲۸، ۲۹، ۳۰، ۳۱، ۳۲، ۳۳، ۳۴، ۳۶، ۳۷، ۳۸، ۴۱، ۴۳، ۴۴، ۴۵، ۴۸، ۵۰، ۵۳، ۵۶، ۵۸، ۶۶، ۶۸، ۶۹، ۷۱، ۷۲، ۷۶، ۷۸، ۷۹، ۸۴، ۸۵، ۸۸، ۹۹، ۱۰۴، ۱۰۷، ۱۰۸، ۱۲۳، ۱۳۸، ۱۴۱، ۱۵۶، ۱۷۴، ۱۸۹، ۱۹۰، ۱۹۸، ۲۰۱، ۲۱۰، ۲۳۸، ۲۴۰، ۲۴۵، ۲۵۶، ۲۸۵، ۳۰۹، ۳۱۸، ۳۳۲، ۳۵۵، ۳۶۴، ۳۷۱، ۳۷۲، ۳۷۸، ۳۷۹، ۴۱۴، ۴۲۰، ۴۳۱، ۴۴۱، ۴۴۲، ۴۸۶، ۴۸۸، ۵۱۵، ۵۱۷، ۵۱۸، ۵۲۱، ۵۴۵، ۵۵۸، ۵۸۲، ۵۹۴، ۵۹۶، ۶۱۸، ۶۳۰، ۶۴۴، ۶۸۴، ۶۸۵، ۷۲۵، ۷۳۲، ۷۳۶، ۷۸۲، ۷۹۷، ۸۰۷، ۸۳۷	۱۱۹	Cnae (بهترین خطا)
۰٫۳۳۲۷	۲۷۶، ۱۱۱، ۹۰، ۸، ۴	۵	Arrhythmia (بهترین تعداد ویژگی)
۰٫۲۶۷	۱۹۶، ۱۱۳، ۹۴، ۹۰، ۱۴، ۷	۶	Arrhythmia (بهترین خطا)

جدول ۸: میانگین، انحراف معیار و واریانس خطا و تعداد ویژگی.

Cnae	Arrhythmia	Madelon	Musk	Isolet	AWMOPSOFS	میانگین
۰٫۲۷۷	۰٫۳۶۷۷	۰٫۱۹۳۷	۰٫۰۶۱	۰٫۱۹۶	خطا	تعداد ویژگی
۹۵٫۲	۹٫۲۵	۷٫۱	۴٫۲۸	۳۷٫۳۳	خطا	انحراف
۰٫۰۸۵	۰٫۰۲۷	۰٫۰۳۴۲	۰٫۰۰۷	۰٫۰۲۵۰	تعداد ویژگی	معیار
۱۸٫۸۸	۲٫۵	۱٫۸۲	۱٫۷۹	۱۲٫۲۹	خطا	واریانس
۰٫۰۰۷۲	۰٫۰۰۰۷۷	۰٫۰۰۱۱	۰٫۰۰۰۰۵	۰٫۰۰۰۰۶	تعداد ویژگی	
۳۵۶٫۷	۶٫۲۵	۳٫۳۳	۳٫۲۳	۱۵۱٫۰۶		

جدول ۹: میانگین، انحراف معیار و واریانس خطا و تعداد ویژگی.

Cnae	Arrhythmia	Madelon	Musk	Isolet	RFPSOFS	میانگین
۰٫۲۳۳	۰٫۳۷۹۴	۰٫۴۰۴	۰٫۰۶۰	۰٫۲۲۶۴	خطا	تعداد ویژگی
۱۰٫۹۴	۱۷٫۸	۴۳٫۲	۱۹٫۲	۶۶٫۱	خطا	انحراف
۰٫۰۴۶۸	۰٫۰۳۸۴	۰٫۰۲۸	۰٫۰۱۰۴	۰٫۰۳۱۶	تعداد ویژگی	معیار
۲۱٫۴۷	۳٫۷۰	۳٫۶۵	۳٫۰۳۱	۱۴٫۱۵	خطا	واریانس
۰٫۰۰۲۱	۰٫۰۰۱۴	۰٫۰۰۰۸۳	۰٫۰۰۰۱۱	۰٫۰۰۱۰	تعداد ویژگی	
۴۶۱٫۳	۱۳٫۷	۱۳٫۳۳	۹٫۱۹	۲۰۰٫۵		

[14] N. Jain, U. Nangia, and J. Jain, "A review of particle swarm optimization," *J. of the Institution of Engineers India: Series B*, vol. 99, no. 4, pp. 407-411, 2018.

[15] N. Gunantara, "A review of multi-objective optimization: methods and its applications," *Cogent Engineering*, vol. 5, no. 1, Article ID: 1502242, 2018.

[16] T. C. Bora, V. C. Mariani, and L. dos Santos Coelho, "Multi-objective optimization of the environmental-economic dispatch with reinforcement learning based on non-dominated sorting genetic

[11] A. Lin, W. Sun, H. Yu, G. Wu, and H. Tang, "Global genetic learning particle swarm optimization with diversity enhancement by ring topology," *Swarm and Evolutionary Computation*, vol. 44, pp. 571-583, Feb. 2019.

[12] R. Tanabe and H. Ishibuchi, "An easy-to-use real-world multi-objective optimization problem suite," *Applied Soft Computing*, vol. 89, Article ID: 106078, Apr. 2020.

[13] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th IEEE Int. Symp. on Micro Machine and Human Science, MHS'95*, pp. 39-43, Nagoya, Japan, 4-6 Oct. 1995.

جدول ۱۰: میانگین، انحراف معیار و واریانس خطا و تعداد ویژگی.

Cnae	Arrhythmia	Madelon	Musk	Isolet	CMDPSOFS	
۰٫۲۳۲۷	۰٫۳۷۹۴	۰٫۴۳۴۹	۰٫۰۴۶۴	۰٫۲۰۹۴	خطا	میانگین
۲۷۸٫۸	۶۵٫۱	۱۳۰٫۳۳	۴۴٫۲	۱۷۸٫۵	تعداد ویژگی	
۰٫۰۴۲۴۶	۰٫۰۴۹۷۱	۰٫۰۳۰۵	۰٫۰۰۲۵	۰٫۰۱۸۹۵	خطا	انحراف معیار
۱۲٫۷۱۳	۱۰٫۷۲	۱۶٫۳۱	۵٫۸۹	۸۰٫۶۲۲	تعداد ویژگی	
۰٫۰۰۱۸	۰٫۰۰۲۴۷	۰٫۰۰۰۹	۰٫۰۰۰۰۰۶	۰٫۰۰۰۳۵	خطا	واریانس
۱۶۱٫۶۲	۱۱۵٫۲	۲۶۶٫۳۳	۳۴٫۷	۶۵٫۱	تعداد ویژگی	

جدول ۱۱: میانگین، انحراف معیار و واریانس خطا و تعداد ویژگی.

Cnae	Arrhythmia	Madelon	Musk	Isolet	HMPISOFS	
۰٫۲۷۲۹	۰٫۴۱۴۷۰	۰٫۴۷۵	۰٫۰۶۳۷۲	۰٫۲۶۲۸	خطا	میانگین
۱۳۱٫۱۴	۲۱٫۴	۳۶٫۲	۲۰٫۵۵	۸۵٫۷۵	تعداد ویژگی	
۰٫۰۷۸۹۰	۰٫۰۲۱۱۸	۰٫۰۸۶۳	۰٫۰۱۸۶۲	۰٫۰۴۱۵۴	خطا	انحراف معیار
۱۱٫۸۸	۵٫۵۰۴۵	۸۰٫۴۱۵	۸٫۷۱۹۳	۷٫۲۷۴۳	تعداد ویژگی	
۰٫۰۰۶۲۲	۰٫۰۰۰۴۴	۰٫۰۰۰۷۴	۰٫۰۰۰۰۳۴	۰٫۰۰۱۷۲۵	خطا	واریانس
۱۴۱٫۱۴۲	۳۰٫۳	۶۴٫۶۶	۷۶٫۰۲۷	۵۲٫۹۱۶۶	تعداد ویژگی	

جدول ۱۲: میانگین، انحراف معیار و واریانس خطا و تعداد ویژگی.

Cnae	Arrhythmia	Madelon	Musk	Isolet	MOPSO	
۰٫۱۹۶۶	۰٫۳۶۵۸	۰٫۴۴۱۶	۰٫۰۵۷۷۰	۰٫۲۱۰۲۵۶	خطا	میانگین
۲۲۳٫۲	۵۶٫۷۵	۱۰۹٫۴	۳۰٫۱	۱۴۵٫۸	تعداد ویژگی	
۰٫۰۲۷۵۹	۰٫۰۵۵۷۰	۰٫۰۴۰۰	۰٫۰۰۶۰۰۴	۰٫۰۳۹۰۰	خطا	انحراف معیار
۷۹۶٫۸۶	۵٫۷۳۷۳	۵۳٫۱۹۷	۶٫۴۴۲۰	۱۰٫۷۵۶۳	تعداد ویژگی	
۰٫۰۰۰۷۶	۰٫۰۰۳۱۰۳	۰٫۰۰۱۶	۰٫۰۰۰۰۳۶	۰٫۰۰۱۵۲۱	خطا	واریانس
۶۳٫۵	۳۲٫۹۱۶	۲٫۸۳	۴۱٫۵	۱۱۵٫۷	تعداد ویژگی	

جدول ۱۳: میانگین، انحراف معیار و واریانس خطا و تعداد ویژگی.

Cnae	Arrhythmia	Madelon	Musk	Isolet	NSGAI	
۰٫۱۹۰۱	۰٫۳۹۶۷	۰٫۴۴۳۴	۰٫۰۵۵۳۳	۰٫۱۹۴۴۴	خطا	میانگین
۲۹۸٫۶	۵۹٫۵	۱۴۹٫۲	۳۳٫۸۳	۱۸۷٫۵	تعداد ویژگی	
۰٫۰۳۱۶۷	۰٫۰۲۰۶۶	۰٫۰۳۸۳	۰٫۰۰۹۶۲۸	۰٫۰۱۸۳۳	خطا	انحراف معیار
۸٫۳۵۴	۵٫۱۹۶۱۵	۷٫۱۵۵۴	۷٫۱۶۷۰	۱۲٫۲۴۳۳	تعداد ویژگی	
۰٫۰۰۱۰	۰٫۰۰۰۴۲	۰٫۰۰۱۴	۰٫۰۰۰۰۹۲	۰٫۰۰۰۳۳	خطا	واریانس
۶۹٫۸	۲۷٫۱۲	۵۱٫۲	۵۱٫۳۶	۱۴۹٫۹	تعداد ویژگی	

جدول ۱۴: مقایسه زمان اجرای الگوریتم‌ها.

AWMOPSOFS	RFPISOFS	CMDPSOFS	HMPISOFS	MOPSO	NSGAI	دیتاست
۱۸٫۲۴۹	۱۶٫۸۶۸	۱۷٫۱۰۲	۱۴٫۲۳۳	۱۵٫۰۹۶	۲۵٫۰۸۷	Isolet
۱۹٫۱۰۶	۱۷٫۸۳۷	۱۸٫۳۸۲	۱۵٫۸۲۹	۱۶٫۹۲۷	۳۰٫۷۸۸	Madelon

[21] Y. Zhang, D. W. Gong, and J. Cheng, "Multi-objective particle swarm optimization approach for cost-based feature selection in classification," *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, vol. 14, no. 1, pp. 64-75, Jan.-Feb. 2015.

[22] H. B. Nguyen, B. Xue, I. Liu, P. Andreae, and M. Zhang, "New mechanism for archive maintenance in PSO-based multi-objective feature selection," *Soft Computing*, vol. 20, no. 10, pp. 3927-3946, 2016.

[23] M. L. Zhang and Z. H. Zhou, "ML-KNN: a lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038-2048, Jul. 2007.

[24] S. Gu, R. Cheng, and Y. Jin, "Feature selection for high-dimensional classification using a competitive swarm optimizer," *Soft Computing*, vol. 22, no. 3, pp. 811-822, 2018.

[25] T. M. Hamdani, J. M. Won, A. M. Alimi, and F. Karray, "Multi-objective feature selection with NSGA II," in *Proc. Int. Conf. on algorithm*, *Applied Thermal Engineering*, vol. 146, pp. 688-700, Jan. 2019.

[17] R. Zhang, F. Nie, X. Li, and X. Wei, "Feature selection with multi-view data: a survey," *Information Fusion*, vol. 50, pp. 158-167, Oct. 2019.

[18] B. Venkatesh and J. Anuradha, "A review of feature selection and its methods," *Cybernetics and Information Technologies*, vol. 19, no. 1, pp. 3-26, Mar. 2019.

[19] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119-1125, Nov. 1994.

[20] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: a multi-objective approach," *IEEE Trans. on Cybernetics*, vol. 43, no. 6, pp. 1656-1671, Dec. 2012.

**مینا زلفی لقیوان** تحصیلات خود را در مقطع کارشناسی در رشته مهندسی کامپیوتر سخت افزار و کارشناسی ارشد در رشته مهندسی کامپیوتر سخت افزار گرایش معماری کامپیوتر و دکترای تخصصی در رشته مهندسی الکترونیک گرایش الکترونیک دیجیتال به ترتیب در سال‌های ۱۳۷۸ از دانشگاه تهران و ۱۳۸۱ از دانشگاه تهران و ۱۳۹۱ از دانشگاه تبریز به پایان رسانده است. هم‌اکنون به عنوان هیأت علمی در دانشگاه تبریز در حال فعالیت است.

**جعفر تنها** تحصیلات دکترای تخصصی خود در را در رشته هوش مصنوعی در سال ۱۳۹۲ از دانشگاه آمستردام هلند دریافت کرده است. ایشان هم‌اکنون به عنوان هیأت علمی در دانشگاه تبریز در حال فعالیت است. یکی از حوزه‌های تخصصی نام‌برده در زمینه یادگیری ماشین نیمه‌نظارتی است.

*Adaptive and Natural Computing Algorithms*, pp. 240-247, Warsaw, Poland, . 2007.

- [26] L. Cagnina, S. C. Esquivel, and C. C. Coello, "A particle swarm optimizer for multi-objective optimization," *J. of Computer Science and Technology*, vol. 5, no. 4, pp. 204-210, 11-14. 2005.

**محمود پرنده** تحصیلات خود را در مقاطع کارشناسی در رشته مهندسی فناوری اطلاعات و کارشناسی ارشد در رشته مهندسی کامپیوتر گرایش هوش مصنوعی به ترتیب در سال‌های ۱۳۹۲ و ۱۳۹۴ از دانشگاه تبریز به پایان رسانده است و هم‌اکنون در رشته مهندسی فناوری اطلاعات گرایش سیستم‌های چندرسانه‌ای در مقطع دکترا در حال تحصیل است.