



دانشگاه کاشان
University of Kashan

مجله محاسبات نرم

SOFT COMPUTING JOURNAL

تارنمای مجله: scj.kashanu.ac.ir



ارائه روشی خودکار جهت مدیریت عدم قطعیت و استخراج دانش در قالب مجموعه قوانین از داده‌ها با استفاده از الگوریتم ژنتیک[✦]

الهام آفرنده^۱، مریمی، راحیل حسینی^۲، استادیار

^۱ گروه مهندسی کامپیوتر، موسسه آموزش عالی توس، مشهد، ایران.

^۲ دانشجوی دکتری، دانشکده فنی و مهندسی، گروه مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد تهران شمال، تهران، ایران.

^۳ دانشکده فنی و مهندسی، گروه مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد شهر قدس، تهران، ایران.

چکیده

اطلاعات مقاله

در دهه اخیر کاربرد تکنیک‌های داده‌کاوی و روش‌های هوشمند جهت استخراج دانش به صورت خودکار از مجموعه انبوه داده‌ها مورد توجه پژوهش‌های زیادی قرار گرفته است. با توجه به ساختار نمایش دانش مبتنی بر قوانین و قابلیت تفسیر بالای این روش در بیان الگوهای پنهان در اطلاعات، استخراج الگوهای پنهان در قالب مجموعه قوانین، از اهمیت بالایی در سیستم‌های هوشمند تصمیم‌یار برخوردار است. این مقاله به ارائه روشی خودکار جهت استخراج قوانین به صورت مستقیم از روی مجموعه داده‌ها می‌پردازد. در روش پیشنهادی از امکان و قابلیت مدیریت عدم قطعیت تئوری مجموعه‌های فازی در مدل‌سازی قوانین استفاده شده است. به منظور رسیدن به مجموعه قوانین بهینه فازی، از الگوریتم ژنتیک استفاده شده و علاوه بر کاهش مجدد قوانین تکراری، بهبود قوانین فازی بر روی آن‌ها انجام شده است. به منظور ارزیابی، روش فازی-ژنتیک پیشنهادی بر روی هفت مجموعه داده مشهور پیاده‌سازی و بر اساس روش تحلیل Accuracy ارزیابی شده است. نتایج به دست آمده بیانگر کارآ بودن روش پیشنهادی در مجموع است.

تاریخچه مقاله:

دریافت ۱۵ اردیبهشت ماه ۱۳۹۹

پذیرش ۲۴ دی ماه ۱۳۹۹

کلمات کلیدی:

روش‌های طبقه‌بندی قوانین عدم قطعیت
قوانین مبتنی بر طبقه‌بندی فازی
تنظیم و بهبود قوانین فازی با GA

© ۱۳۹۹ - مجله محاسبات نرم، کلیه حقوق محفوظ است.

۱. مقدمه

حجم عظیمی از داده‌ها می‌پردازد که گاه از دید متخصصان علوم پزشکی پنهان می‌مانند [۱]. در این راستا الگوریتم‌های یادگیری ماشین در پیش‌بینی بیماری‌ها بسیار حایز اهمیت بوده و رو به افزایش هستند [۲]. در این میان سه الگوریتم ماشین‌های بردار پشتیبان (SVM)^۱، بی‌زین ساده^۲ و مدل‌های رگرسیون^۳ بسیار مورد استفاده و مقایسه قرار گرفته‌اند، البته این‌ها روش‌های سنتی هستند که در ساخت مدل‌ها نقش دارند

در دهه اخیر کاربرد تکنیک‌های داده‌کاوی و روش‌های هوشمند جهت استخراج دانش به صورت خودکار از مجموعه انبوه داده‌ها مورد توجه پژوهش‌های زیادی قرار گرفته است. علم داده‌کاوی به کشف الگوهای پنهان و ناشناخته در میان

✦ نوع مقاله: پژوهشی

* نویسنده مسئول

پست(های) الکترونیک: afarandeh.elham1@gmail.com (آفرنده)

rahil.hosseini@qodsiau.ac.ir (حسینی)

¹ Support Vector Machine

² Naïve Bayes

³ Regression

اما یکی از چالش‌هایی که در راستای سیستم‌های هوشمند فازی وجود دارد این است که مدل‌های فازی پویا نیستند و آموزشی در آن‌ها اتفاق نمی‌افتد، لذا برای حل این مشکل به سراغ الگوریتم‌های تکاملی می‌روند [۱۷]. از این میان الگوریتم ژنتیک یکی از بهترین روش‌های جستجوی اکتشافی است [۱۸، ۱۹].

به‌طور خاص، فرآیند تنظیم ژنتیک شامل انتخاب خودکار بهترین پارامترهای سیستم برای بهبود عملکرد مدل نهایی و بدون تغییر در قاعده موجود است. سیستم فازی ژنتیکی بهینه‌سازی قوانین را آسان می‌کند [۱۹]. برای به‌کارگیری ژنتیک در مجموعه قوانین فازی دو رویکرد را می‌توان بررسی نمود: رویکرد اول تنظیم و اصلاح پارامترها و قوانین فازی است [۴، ۹، ۲۰] (که در این حالت باید سیستم فازی طراحی شده باشد و فقط پارامترها و قوانینی که باید اصلاح شوند به عنوان کوروموزوم‌های ژنتیک در نظر گرفته شوند، تا الگوریتم ژنتیک آن‌ها را اصلاح نماید)؛ در حالی که رویکرد دوم ایجاد قوانین فازی به کمک الگوریتم ژنتیک است [۸]. هدف از این مقاله ارائه روشی خودکار جهت مدیریت عدم قطعیت و استخراج دانش در قالب مجموعه قوانین از داده‌ها در چند مرحله می‌باشد که بدین منظور بعد از مجموعه عملیات پیش‌پردازش، در ابتدا به سراغ روش استخراج قوانین به صورت مستقیم از روی مجموعه داده‌ها خواهیم رفت و بعد از آن تکنیک استخراج قوانین با روش طبقه‌بندی فازی از روی مجموعه قوانینی که در مرحله قبل به‌دست آمده را بررسی خواهیم کرد که در این مرحله قوانین ناسازگار، تکراری و متناقض از مجموعه قوانین حذف خواهند شد. سپس برای حل مشکلات فازی و به‌منظور دست یافتن به مجموعه قوانین بهینه به سراغ الگوریتم ژنتیک رفته و علاوه بر کاهش مجدد قوانین تکراری، بهبود قوانین فازی حاصل خواهد شد. در فاز دوم برای ارزیابی و مقایسه الگوریتم پیشنهادی با توجه به اهمیت و کاربردی بودن روش‌های رگرسیون، ماشین بردار پشتیبان و بیز ساده بر روی مجموعه داده‌های مربوط به پیش‌بینی بیماری‌ها و چند نوع دیگر اعمال شده و نتایج

و نمی‌توانند اطلاعات پنهان‌شده در داده‌ها را به اندازه کافی نشان دهند [۳]، علاوه بر این در هوش مصنوعی به دنبال شبیه‌سازی هوشمندی مغز انسان هستیم. این روش‌ها از جمله SVM (علی‌رغم این‌که از مدلی تولید شده توسط آن‌ها می‌تواند برای استخراج دانش استفاده شود)، فاصله زیادی با شبیه‌سازی مغز انسان دارند.

در [۲] بیش از ۲۷ مقاله در زمینه پیش‌بینی بیماری‌ها که نتایج آن‌ها با سه روش بالا مورد ارزیابی قرار گرفته، بررسی شده است. بر اساس گزارشات مقالات بررسی شده، این مقاله به این نتیجه رسیده است که از میان سه روش نام برده شده در بالا روش بیز ساده نسبت به دو روش دیگر دقت بالاتری داشته است.

با توجه به ساختار نمایش دانش مبتنی بر قوانین و قابلیت تفسیر بالای این روش در بیان الگوهای پنهان در اطلاعات، استخراج الگوهای پنهان در قالب مجموعه قوانین، از اهمیت بالایی در سیستم‌های هوشمند تصمیم‌یار برخوردار است [۴-۹]. لذا مقالات [۱۰، ۱۱] به بیان پیشرفت‌های اخیر در تولید دانش و مدیریت عدم قطعیت پرداخته‌اند. به‌طورکلی تکنیک‌های استخراج دانش مبتنی بر قوانین، به دو دسته قطعی و غیرقطعی تقسیم می‌شوند [۱۲-۱۴]. دسته اول شامل روش‌هایی است که به صورت مستقیم قوانین را از روی داده‌ها استخراج می‌کنند مانند الگوریتم‌های Cn2, Ripper, Sequential Covering (SQ). در میان روش‌های غیرقطعی، سیستم‌های طبقه‌بندی مبتنی بر قانون فازی^۱ (FRBS) به طور گسترده استفاده می‌شوند [۱۵]، زیرا به ما امکان می‌دهند با اطلاعات پرسروصدا، نادرست یا ناقصی که اغلب در بسیاری از مشکلات دنیای واقعی وجود دارد، مقابله کرده و توازن خوبی بین دقت تجربی تکنیک‌های مهندسی سنتی و تفسیر حاصل از استفاده‌ی برجسب‌های زبانی که معنایی آنها نزدیک به زبان طبیعی است، فراهم کنیم و تا آنها به راحتی قابل تفسیر بوده و به خوبی توانایی مدل‌سازی عدم قطعیت در مسائل تصمیم‌گیری را دارا باشند [۴، ۵، ۹، ۱۲، ۱۳، ۱۶].

¹ Fuzzy Rule-Based System

است که اگر تنوع داده‌ها زیاد باشد تعداد قوانین تولید شده از روی مجموعه داده زیاد و در نتیجه پردازش‌ها سنگین می‌شوند که این در داده‌های با تعداد زیاد اصلاً مطلوب نیست. با زیاد شدن تعداد قوانین (دانش استخراج شده) باید بتوان از بین آن‌ها تعدادی را استخراج کرد. یک روش دسته‌بندی داده‌ها و کاهش تعداد قوانین است.

۳. استخراج دانش با استفاده از قوانین فازی به

عنوان مدل‌های طبقه‌بندی عدم قطعیت

یکی از مدل‌هایی که برای استخراج دانش در سیستم‌های عدم قطعیت می‌توان استفاده کرد، مدل‌های فازی هستند که قابلیت بسیار بالایی در عدم قطعیت در دانش دارند. یکی از روش‌های متداول جهت تعریف قوانین یک سیستم استنتاج فازی، استفاده از جدول جستجو است. در این روش از ترکیبی از دانش خودآگاه و ناخودآگاه در طراحی سیستم‌های فازی استفاده می‌شود [۴]. استفاده از سیستم‌های مبتنی بر قوانین فازی (FRBS) برای شناسایی سیستم، که ساختار کلی آن در شکل (۱) نشان داده شده است، می‌تواند رویکردی باشد که برای مدل‌سازی در استفاده از یک زبان توصیفی مبتنی بر منطق فازی با گزاره‌های فازی مورد استفاده قرار گیرد. این الگو که توانایی خود را به صورت خودکار ثابت کرده است انواع مختلفی از مدل‌های فازی از داده‌های مختلف را ایجاد می‌کند و اجازه می‌دهد دانش انسانی سیستم خبره و ادغام پردازش عددی و نمادین در یک طرح مشترک ایجاد شود [۵].

مراحل طراحی سیستم مبتنی بر قانون فازی که در این مقاله پیشنهاد شده است به صورت زیر است:

گام اول: در نظر گرفتن مجموعه قوانینی را که با روش مستقیم (DM) استخراج کرده‌ایم.

گام دوم: طراحی مجموعه‌های فازی که مقادیر ورودی و خروجی در مجموعه داده‌ها را پوشش می‌دهد. تعداد عبارات‌های زبانی با توجه به دانش فرد خبره و دانش موجود در دامنه کاری تعیین می‌شود (در این مرحله پایگاه داده فازی

به‌دست آمده از روش پیشنهادی با آن‌ها مقایسه شده‌اند که نتایج بدست آمده گویای کارآیی روش پیشنهادی است. برای پیاده‌سازی نتایج از نرم‌افزار پایتون نسخه ۳/۷ استفاده شده است. ساختار مقاله به شرح زیر است: در بخش دوم توضیحات مربوط به استخراج دانش (قوانین) در طبقه‌بندی قطعی با روش مستقیم (DM) [۱۲] آورده شده است. بخش سه استخراج دانش با قوانین فازی در طبقه‌بندی‌های عدم قطعیت؛ بخش چهارم بهبود قوانین فازی با کمک الگوریتم ژنتیک؛ بخش پنجم نتایج پیاده‌سازی و ارزیابی نتایج به‌دست آمده و در انتها بخش ششم جمع‌بندی مطالب را بیان می‌کند.

۲. استخراج دانش (قوانین) با روش مستقیم

همان‌طور که گفته شد از آنجایی که مغز انسان بر پایه قوانین آگروآنگاه عمل می‌کند و هوش مصنوعی و سیستم خبره در مباحث خود همیشه به دنبال شبیه‌سازی هوشمندی مغز انسان است، تکنیک‌هایی که قوانین را استخراج می‌کنند بسیار حایز اهمیت هستند. مدل‌های طبقه‌بندی قوانین به دو دسته کلی، مدل‌های قطعی و غیرقطعی دسته‌بندی می‌شوند. یکی از روش‌های مدل‌های قطعی، روش DM است. در این روش، استخراج قوانین به طور مستقیم از روی مجموعه داده صورت می‌گیرد بدین صورت که در ابتدا از یک مجموعه خالی از قوانین شروع می‌کنیم و بعد یک رکورد را به عنوان یک قانون در نظر گرفته، تمام رکوردهایی که با این قانون تطابق دارند را از پایگاه داده بیرون می‌کشیم، حال یکسری داده دیگر باقی می‌مانند که تطابق ندارند، مجدداً رکورد بعدی را به عنوان قانون در نظر می‌گیریم و مراحل قبلی را برای آن تکرار می‌کنیم تا به یک شرط توقف برسیم. در این مقاله شرط توقف تعریف نشده است و این مراحل را برای کلیه رکوردهای مجموعه داده تکرار می‌کنیم تا مجموعه قوانین را استخراج کنیم.

این روش بسیار ساده است، اما مشکل و چالشی که دارد این

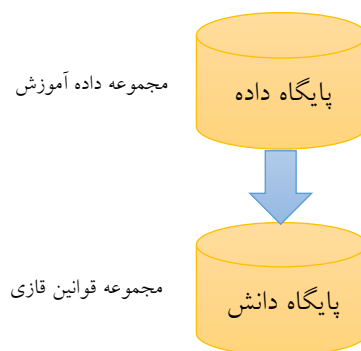
¹ Directed Method

در این رابطه x_1 و x_2 متغیرهای ورودی-خروجی (در داده‌های ما همان ویژگی‌های هر رکورد همراه با خروجی) و $\mu_{A_{i1}}$ ، $\mu_{A_{i2}}$ و μ_{C_j} به ترتیب نشان‌دهنده مقدار تابع عضویت متغیرهای ورودی (یعنی x_1 و x_2) و متغیر خروجی می‌باشند.

گام پنجم: پالایش قوانین با توجه به درجه سازگاری هر قانون شامل حذف قوانین تکراری و حذف قوانین ناسازگار.

گام ششم: تشکیل مجموعه قوانین فازی و استخراج دانش بر پایه مدل‌های عدم قطعیت.

که در شکل (۱) ساختار آن نشان داده شده است، تعریف می‌شود).



شکل (۱): ساختار سیستم‌های مبتنی بر قوانین فازی

۴. استخراج دانش و بهبود مجموعه قوانین فازی با کمک الگوریتم ژنتیک

علی‌رغم تاریخچه موفقیت‌آمیز در طراحی FRBS (مجموعه قوانین فازی)، عدم توانایی یادگیری برای توصیف اکثر کارها باعث ایجاد برنامه‌ای برای مطالعه FRBS با قابلیت یادگیری در اوایل دهه ۱۹۹۰ شد. در واقع یکی از چالش‌هایی که در راستای سیستم‌های هوشمند فازی وجود دارد این است که مدل‌های فازی پویا نیستند و آموزشی در آن‌ها اتفاق نمی‌افتد، بنابراین برای حل این مشکل می‌توان به سراغ الگوریتم‌های تکاملی و شبکه‌های عصبی رفت.

فرآیندهای یادگیری ژنتیکی سطوح مختلف پیچیدگی را مطابق با تغییرات ساختاری تولید شده توسط الگوریتم [۶]، از ساده‌ترین مورد بهینه‌سازی پارامترها تا بالاترین سطح پیچیدگی یادگیری مجموعه قاعده‌های یک سیستم مبتنی بر قانون، پوشش می‌دهند. تعداد قوانین در مجموعه قوانین فازی (RB) با افزایش ورودی‌ها، به صورت نمایی افزایش می‌یابد. از این رو، یک روش تولید قانون فازی احتمالاً مجموعه قوانین فازی را شامل می‌شود که دارای قوانین ناخواسته‌ای است. این امر باعث کاهش دقت و تفسیر مدل‌های فازی می‌شود. در بین این قوانین، ما می‌توانیم قواعد افزایشی را ایجاد کنیم که اعمال آن‌ها تحت قوانین دیگری در RB قرار دارد.

روش‌های کاهش قاعده به عنوان تکنیک‌های پس‌پردازش برای حل مشکلات بعدی، در ساختارهای FRBS استفاده می‌شوند

گام سوم: تعریف یک قانون فازی به ازای هر داده ورودی و خروجی. در این مرحله باید هر رکورد (هر قانون استخراج شده در روش DM) را به دنیای فازی نگاشت کنیم. یک قانون فازی برای استخراج دانش در مدل‌های عدم قطعیت به صورت رابطه (۱) تعریف می‌شود:

$$R_k: \text{IF } x_1 \text{ is } A_{i1} \text{ and } \dots \text{ and } x_n \text{ is } A_{in} \quad (1)$$

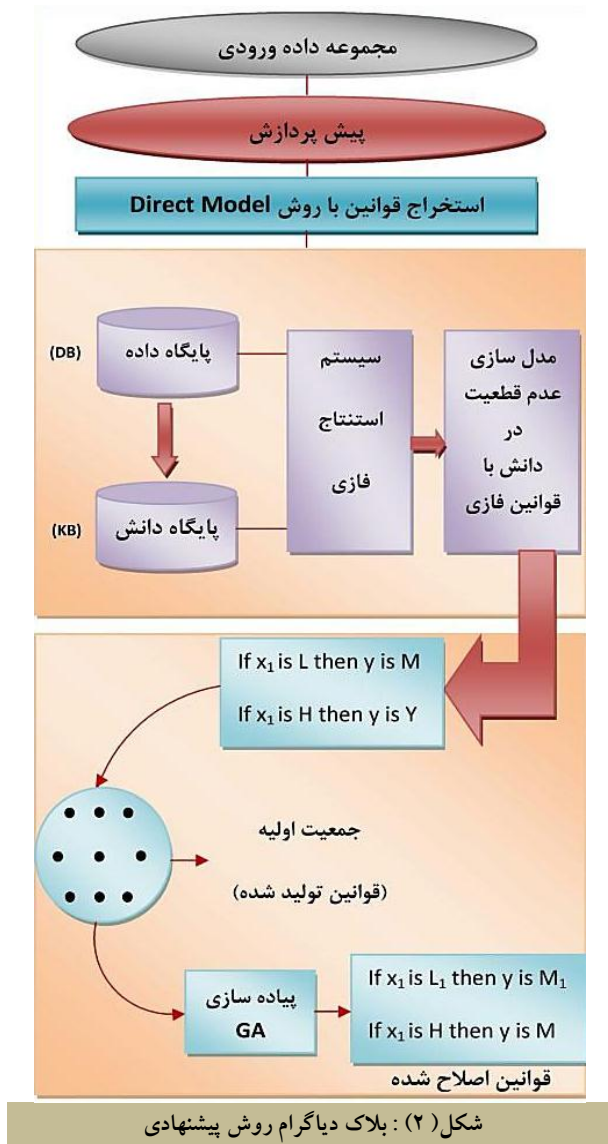
$$\text{THEN Class} = C_j$$

که در آن R_k شناسه هر قانون استخراج شده از مرحله قبل است، x_1, \dots, x_n ویژگی‌های مجموعه داده هستند و A_{i1}, \dots, A_{in} متغیرهای زبانی هستند که توسط مجموعه‌های فازی تعریف شده‌اند و برای نشان دادن صفات استفاده می‌شوند. در نهایت طبقه‌ای است که به الگو اختصاص داده می‌شود.

گام چهارم: تعیین یک درجه قدرت برای هر قانون. به ازای هر داده ورودی-خروجی در مجموعه داده، میزان عضویت هر عنصر ورودی-خروجی با توابع عضویت ورودی و خروجی تعریف شده تعیین می‌شود، سپس درجه قدرت قانون طبق رابطه (۲) با ضرب مقدار تابع عضویت متغیرهای ورودی و خروجی در یکدیگر، بدست می‌آید.

$$\text{IF } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ then } y \text{ is } C_j \quad (2)$$

$$\text{درجه قدرت هر قانون} = \mu_{A_{i1}} \times \mu_{A_{i2}} \times \mu_{C_j}$$



اولین و یکی از مهم‌ترین گام‌ها قبل از انتخاب نوع روش، پیش‌پردازش داده‌ها می‌باشد که در این راستا اقدامات زیر بر روی مجموعه داده‌ها انجام شده است [۲۸]:

- پیدا کردن داده‌های NULL (آنهایی که در مجموعه داده مقدار ندارند) و جایگذاری میانگین کل ویژگی‌های مربوط به هر کلاس به جای مقادیر خالی (علت انتخاب این روش در ادامه توضیح داده خواهد شد).
- حذف ویژگی‌هایی که تعداد مقادیرهای یونیک آنها زیاد باشد و در واقع ویژگی خوبی به حساب نیایند.
- اختصاص دادن شماره‌های شناسه یکتا برای مقادیری که با فرمت رشته هستند، به عنوان مثال اختصاص 0 به

[۷]. انتخاب قانون، گسترده‌ترین روش کاهش قاعده برای مدل‌های فازی زبانی است و الگوریتم‌های ژنتیک معمول‌ترین روش بهینه‌سازی برای اجرای آن‌ها هستند. برای به‌کارگیری ژنتیک در مجموعه قوانین فازی دو رویکرد را می‌توان بررسی نمود [۲۱-۲۵]:

۱. تنظیم و اصلاح پارامترها و قوانین فازی (که در این حالت باید سیستم فازی طراحی شده باشد و فقط پارامترها و قوانینی که می‌خواهیم اصلاح شوند به عنوان کوروموزوم‌های ژنتیک در نظر گرفته شده، تا الگوریتم ژنتیک آن‌ها را اصلاح نماید).
۲. ایجاد قوانین فازی به کمک GA [۸، ۹]

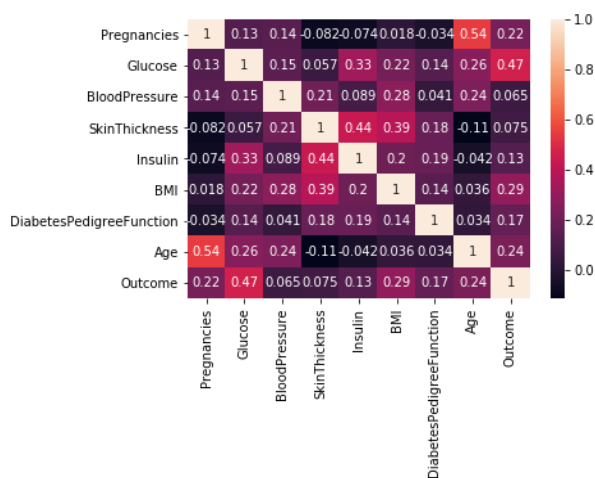
۵. روش پیشنهادی و پیاده‌سازی نتایج

در این مقاله، بعد از پیش‌پردازش، در ابتدا روش استخراج قوانین به صورت مستقیم بر روی مجموعه داده‌ها پیاده‌سازی شده، سپس در گام بعد بر روی مجموعه قوانین بدست آمده در مرحله قبل، قوانین فازی برای استخراج دانش به کار برده شده و در مرحله آخر الگوریتم ژنتیک برای اصلاح پارامترها و قوانین فازی استفاده شده است (جزئیات کامل آنها در بخش ۳ و ۴ توضیح داده شد). از آنجایی که الگوریتم ژنتیک مشهورترین تکنیک در تحقیقات مربوط به الگوریتم‌های تکاملی است و به راحتی در دامنه وسیعی از مسائل، از جمله مسائل بهینه‌سازی کاربرد دارد، این الگوریتم برای اصلاح قوانین انتخاب شده است.

برای بررسی و پیاده‌سازی روش‌های توضیح داده شده در قسمت‌های قبلی از ۷ مجموعه داده مختلف استفاده خواهد شد. این مجموعه داده‌ها از پایگاه داده UCI و Kaggle گردآوری شده‌اند [۲۶، ۲۷]. مراحل کار در دو فاز چند مرحله‌ای دنبال می‌گردد. بلاک دیاگرام روش پیشنهادی در شکل (۲) نشان داده شده است. در ادامه بر اساس این دیاگرام، مراحل کار پیشنهادی شرح داده خواهد شد.

مشاهده کرد. به عنوان مثال ماتریس همبستگی برای داده دیابت در شکل (۳) نشان داده شده است. نتایج شکل (۳) مستقیماً خروجی حاصل از اجرا در محیط پایتون می‌باشد، لذا اطلاعات انگلیسی است. عبارات نوشته شده در این ماتریس مربعی نام ویژگی‌ها و خروجی داده هستند. اعداد داخل آن میزان وابستگی و مستقل بودن متغیرها را نسبت به یکدیگر نشان می‌دهد. هر چه عدد مربوط به دو متغیر به صفر نزدیک‌تر باشد یعنی مستقل‌تر نسبت به هم هستند و هر چه عدد به یک نزدیک باشد میزان وابستگی بیشتر را نشان می‌دهد (به عنوان مثال، اعداد قطر اصلی یک هستند که نشان می‌دهند یک متغیر کاملاً به خودش وابسته است). همان‌طور که از مقادیر ماتریس همبستگی مشاهده می‌شود به‌عنوان مثال متغیرهای Age و Bmi تقریباً نسبت به یکدیگر مستقل هستند (عدد مرتبط با این دو ویژگی در ماتریس ۰/۰۳۶ می‌باشد).

Correlation of Features



شکل (۳): ماتریس همبستگی داده دیابت

بعد از پیش‌پردازش به سراغ اولین روش گفته شده (یعنی DM) می‌رویم که نتایج آن در جدول (۳) آورده شده است. همان‌طور که نتایج نشان می‌دهند، تعداد قوانین در این روش زیاد است و لذا به دنبال کاهش قوانین و بهبود آن‌ها هستیم. نتایج نشان می‌دهد هر چه تنوع داده‌ها زیاد باشد تعداد قوانین تولید شده از روی مجموعه داده زیاد خواهد بود و در نتیجه پردازش‌ها سنگین‌تر می‌شوند.

Male و 1 به Female.

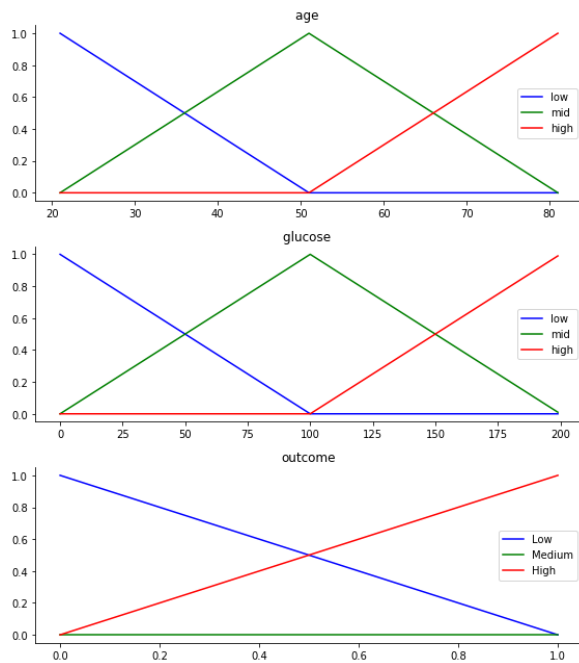
- در مواردی که ویژگی‌ها گسسته هستند ردیفی که مقدار خالی دارد حذف می‌شود.
- در جدول (۱) مشخصات مجموعه داده‌ها همراه با تغییرات قبل و بعد از پیش‌پردازش آورده شده است.

جدول (۱): مشخصات مجموعه داده‌های انتخابی

Feature	قبل از پیش‌پردازش		بعد از پیش‌پردازش		Count
	تعداد نمونه‌ها	ویژگی‌ها	تعداد نمونه‌ها	ویژگی‌ها	
پارکینسون	۱۹۵	۲۳	۱۹۵	۲۳	۲
دیابت	۷۶۸	۹	۷۶۸	۸	۲
بیماری قلبی	۳۰۳	۱۴	۳۰۳	۱۳	۲
هپاتیت	۱۵۵	۱۹	۱۵۰	۱۹	۲
iris	۱۵۰	۵	۱۵۰	۵	۳
پوکمون	۸۰۰	۱۲	۴۱۴	۹	۲
تخصیص وام	۶۱۴	۱۲	۵۱۱	۸	۲

از مجموعه داده‌های انتخابی، داده‌های پارکینسون، دیابت، بیماری قلبی و Iris داده‌های تمیز بودند که قبل و بعد از پیش‌پردازش تغییری نیافتند. داده هپاتیت ۱۶۷ مقدار خالی در ۱۵ ستون داشت که به جای آن‌ها میانگین مقادیر هر ویژگی (هر ستون در مجموعه داده) قرار داده شد. در این قسمت از دانش فرد خبره استفاده شد و با بررسی انجام شده مشخص شد ستون‌هایی که مقدار آن‌ها خالی بود در ستون‌هایی از مجموعه داده قرار داشتند که سایر مقادیر آن‌ها ۱ یا ۲ بود، لذا از روش میانگین‌گیری برای جایگزاری مقدار استفاده شد. داده‌های پوکمون و تخصیص وام مقدار خالی نداشتند اما داده‌های رشته‌ای و گسسته بدون مقدار بودند که به جای رشته‌ها مقادیر عددی قرار داده شد و داده‌های گسسته حذف گردیدند.

یکی از اقدامات دیگری که جهت کمک به فرد خبره در این مرحله انجام شد، بدست آوردن ماتریس همبستگی است که به کمک آن می‌توان رابطه بین ویژگی‌ها، متغیرها، میزان وابستگی و مستقل بودن آن‌ها و تاثیر آنها در نتیجه نهایی را



شکل (۵): توابع عضویت ترسیم شده برای دو ویژگی دیابت و ستون خروجی آن

با توجه به توابع عضویتی که در مرحله قبل بدست آوردیم حال بررسی می‌کنیم هر یک از رکوردها چه درجه عضویتی نسبت به تابع عضویت دارند، هر کدام که درجه عضویت بیشتری داشت به آن تابع عضویت مرتبط می‌کنیم، قوانین را پالایش کرده و قوانین تکراری را حذف می‌نماییم و با این کار یک مجموعه قانون فازی تعریف می‌کنیم. همان‌طور که در جدول (۳) ملاحظه می‌کنید توانسته‌ایم تعداد قوانین را نسبت به روش مستقیم بسیار کاهش دهیم.

در گام بعدی برای اصلاح قوانین فازی و رفع چالش‌های آن که در بخش ۴ به آن‌ها پرداخته شد، به سراغ الگوریتم ژنتیک می‌رویم و هر قانون فازی بدست آمده از گام قبل را به عنوان یک کوروموزوم (ژن‌ها، ویژگی‌های مجموعه داده و مقدار آنها مقدار متغیر زبانی بدست آمده در فازی) در نظر می‌گیریم. توجه کنید که ما ۷۰ درصد از قوانین را برای آموزش و مابقی را برای آزمون در نظر می‌گیریم (برای دیگر روش‌ها که برای مقایسه انتخاب کرده‌ایم نیز همین نسبت انتخاب شده است). عمل نخبه‌گزینی برای دو بهترین کوروموزوم اعمال می‌گردد. انتخاب والد‌ها هم بر اساس این است که بعد از نخبه‌گزینی،

از آنجایی که تعداد قوانین بدست آمده از روش قبلی معمولاً زیاد است و همچنین به دنبال این هستیم که بتوانیم روشی را انتخاب نماییم که توازن خوبی بین دقت تجربی روش‌های مهندسی سنتی و تفسیر حاصل از برچسب‌های زبانی فراهم کند و همچنین توانایی مدل‌سازی عدم قطعیت در مسائل تصمیم‌گیری را داشته باشد، در گام بعد برای استخراج دانش به سراغ روش FRBS رفتیم که توضیحات کامل آن در بخش ۳ گفته شد. در این روش ویژگی‌های هر مجموعه داده را به همراه ستون خروجی (که مشخص کننده کلاس مجموعه داده است) به عنوان متغیرهای سیستم فازی (x_1, x_2, \dots, y) در نظر می‌گیریم و برای آنها بر اساس بازه ماکزیمم و مینیمم‌ای که دارند توابع عضویت تعریف می‌کنیم (نمونه‌ای از نحوه انتخاب بازه ماکزیمم و مینیمم برای داده دیابت در جدول (۲) آورده شده است). توابع عضویت را مثلی در نظر گرفته (به دلیل سادگی و پرکاربرد بودن، این نوع تابع انتخاب شده است؛ یک راس در مرکز منطقه قرار دارد، دو رأس دیگر به ترتیب در مراکز دو منطقه همسایه قرار دارند و مقادیر عضویت آن‌ها برابر با صفر است) و برای هر متغیر سه تابع عضویت Low، Mid و High تعریف می‌کنیم.

جدول (۲): مقادیر ماکزیمم و مینیمم سه تا از متغیرها برای مجموعه داده دیابت در تشکیل توابع عضویت

نام متغیر	کمترین مقدار	بیشترین مقدار
Glucose	۹۹	۱۳۴
Age	۲۱	۸۱
Outcome	۰	۱

برای نشان دادن مراحل کار دو متغیر از ویژگی‌های داده دیابت به همراه خروجی این داده را انتخاب نمودیم و مقادیر بیشترین و کمترین مقدار آن‌ها را در جدول (۲) نشان دادیم و در گام بعد توابع عضویت این سه متغیر را بر اساس تقسیم‌بندی انجام شده بین بیشترین و کمترین مقدار تعریف کردیم که شکل (۵) نمونه‌ای از توابع عضویت ترسیم شده برای این متغیرها را نشان می‌دهد.

تابع برازندگی، کارایی مجموعه قوانین در سیستم فازی را نشان می‌دهد که با معیار Accuracy اندازه‌گیری می‌شود که برای هر کوروموزوم بر اساس کسری از قوانین که مقدم و نتیجه یک قانون را شامل می‌شود، محاسبه می‌شود.

$$Accuracy = (n * 100) / s \quad (۳)$$

در رابطه (۳)، پارامتر s تعداد قوانینی است که قسمت مقدم آن‌ها با قسمت مقدم قانون مورد نظر برابر است. در واقع s میزان همپوشانی (Coverage) یک قانون را محاسبه می‌کند. پارامتر n تعداد قوانینی است که بعد از برابری قسمت مقدم آن‌ها، قسمت نتیجه یا تالی آن‌ها با هم برابر است و در انتها Accuracy درصدی از قوانین که با هر دو قسمت مقدم و نتیجه تطابق دارد، را محاسبه کرده که به عنوان تابع هزینه الگوریتم ژنتیک در نظر گرفته شده است.

بعد از اجرای GA مجموعه قوانین اصلاح شده را خواهیم داشت که ما توانسته‌ایم تعداد قوانین را کاهش و قوانین تکراری را نیز حذف نماییم. حال مجموعه داده آزمون را با قوانین استخراج شده از GA بررسی نموده و بررسی می‌نماییم چند درصد از این قوانین در قوانین GA وجود دارند. نتایج حاصل از شبیه‌سازی در جدول (۴) آورده شده است.

جدول (۴): نتایج حاصل از شبیه‌سازی

مجموعه داده	تعداد قوانین استخراج شده		تعداد قوانین بهبود داده شده با GA
	از روش DM	از روش FRBS	
دیابت	۷۶۸	۱۵۴	۱۰۹
پارکینسون	۱۹۵	۳	۲
بیماری قلبی	۳۰۳	۲۶۰	۱۹۰
هپاتیت	۱۵۵	۱۴۴	۸۳
Iris	۱۴۷	۲۰	۱۷
پوکمون	۴۱۳	۲۸۶	۲۳۶
تخصیص وام	۵۱۱	۲۲۷	۱۵۱

برای بیان بهتر توضیحات داده شده، شبه‌کد الگوریتم ژنتیک بعد از استخراج قوانین به روش مستقیم و اصلاح و کاهش آن‌ها با کمک فازی در شکل (۶) ذکر شده است.

نصف بهترین والد‌ها انتخاب و بعد از ادغام و جهش فرزندان جدید جایگزین نصفه جمعیت که تابع برازندگی کمی دارند می‌گردند. برای بهتر کار کردن، بعد از خواندن قوانین فازی مقادیر سه تابع عضویت (Low, Mid, High) را عددی کرده و معادل عددی آن را جایگزین می‌کنیم و سپس برای ایجاد و کار راحت‌تر در هر کوروموزوم مقادیر هر قانون را به صورت یک رشته عددی تبدیل می‌کنیم. در جدول (۳) برخی از پارامترهای GA آورده شده است.

جدول (۳): پارامترهای الگوریتم GA-FUZZY RULE BASE

مقدار	مؤلفه
Fuzzy	اندازه جمعیت
تعداد قوانین بدست آمده از روش Fuzzy Rule	عملگر باز ترکیب و جهش
هر دو تک نقطه ای	نرخ باز ترکیب و جهش
۰/۹	تعداد تکرار جمعیت
۱۰۰	تابع برازندگی
دقت قانون بدست آمده با مجموعه قوانین اولیه	طول کوروموزوم
هر ردیف قانون بدست آمده از روش RFBS	ژن‌های کوروموزوم
ویژگی‌های هر مجموعه داده و مقدار آن‌ها	مقدار متغیر زبانی بدست آمده در فازی

ساختار هر کوروموزومی که در روش ما نظر گرفته شده است، به صورت زیر است.

$\mu(x_1)$	$\mu(x_2)$	$\mu(x_3)$...	$\mu(c)$
low	high	mid	...	low

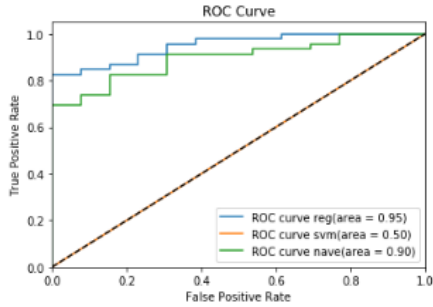
کوروموزوم

ژن‌های هر کوروموزوم مقادیر متغیرهای زبانی هر ویژگی و ژن آخر مقدار متغیر زبانی کلاس قانون مورد نظر می‌باشد. برای سهولت کار هر یک از کوروموزوم‌ها را بر اساس مقدار تابع زبانی آن تبدیل به رشته عددی می‌نماییم. بدین صورت که اگر مقدار تابع عضویت هر ویژگی Low باشد عدد 0، برای Mid=1 و برای High=2 را در نظر می‌گیریم. بنابراین کوروموزوم‌هایی عددی خواهیم داشت.

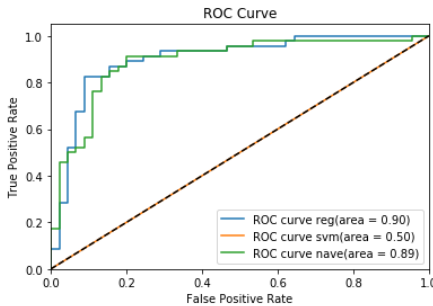
$\mu(x_1)$	$\mu(x_2)$	$\mu(x_3)$...	$\mu(c)$
0	2	1	...	0

کوروموزوم

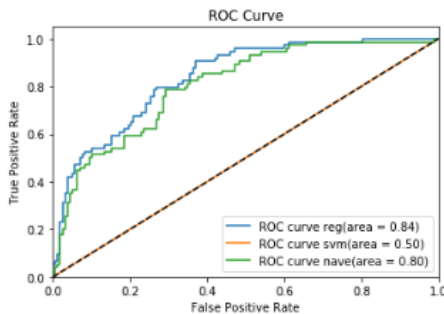
در فاز بعدی این مقاله، برای مقایسه کار خود به سراغ روش‌های رگرسیون، بیز ساده و ماشین بردار پشتیبان بر روی مجموعه داده‌های انتخابی می‌رویم. در شکل (۷) نتایج حاصل از این پیاده‌سازی آورده شده است.



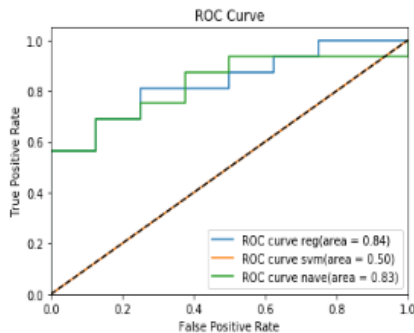
مجموعه داده پارکینسون



مجموعه داده Heart



مجموعه داده دیابت



مجموعه داده هیپاتیت

شکل (۷): نتایج حاصل از منحنی ROC بر روی ۴ مجموعه داده با

مدل‌های رگرسیون، SVM و NAIVE

جدول (۳) Set parameters according the

```

Pop1=get_fuzzy rules
Convert rules to numeric string(low=0,mid=1,high=2)
Pop,test=split_train_test
While (–stop condition)
  Evaluate(pop)
  rule=sort(pop)
  If (len(rule)%2=0)
    i=1
    n=len(rule)/2
    For(i<n)
      Point=get_random_point
      rule1,rule2=cross_mut(rule[i],rule[i+1])
      rule[n+i]=rule[i]
      rule[n+i+1]=rule[i+1]
    else
      i=2
      n=len(rule)/2
      For(i<n)
        Point=get_random_point
        rule1,rule2=cross_mut(rule[i],rule[i+1])
        rule[n+i]=rule[i]
        rule[n+i+1]=rule[i+1]
  evaluate(rule)
  pop=sort(rule)
  eliminate duplicate rules in pop
  For each test_data compare with each rule in pop and
  calculate accuracy
    
```

شکل (۶): شبه کد الگوریتم GA در روش پیشنهادی

نتایج حاصل از پیاده‌سازی روش پیشنهادی در جدول (۵) آمده است. همان‌طور که از نتایج مشخص است توانستیم تعداد قوانین را کاهش دهیم و قوانینی که خیلی در نتیجه تاثیرگذار نیستند و یا ناسازگار هستند را حذف کنیم. حال آن که دقت سیستم خوب باقی مانده است.

جدول (۵): نتایج حاصل از روش پیشنهادی

مجموعه داده	تعداد قوانین پس از بهبود توسط GA	درصد صحت عملکرد مجموعه قوانین داده تست
دیابت	۱۰۹	۷۱٫۵۵
پارکینسون	۲	۸۰
بیماری قلبی	۱۹۰	۵۶٫۶
هیپاتیت	۸۳	۸۸
Iris	۱۷	۹۲
پوکمون	۲۸۶	۷۳٫۲
تخصیص وام	۲۲۷	۷۴

پیش‌پردازش، در ابتدا روش استخراج قوانین به صورت مستقیم از روی مجموعه داده‌ها ذکر شده و بعد از آن تکنیک استخراج قوانین با روش طبقه‌بندی فازی از روی مجموعه قوانینی که در مرحله قبل بدست آمده، مورد بررسی قرار گرفت. روش پیشنهادی در این مرحله موفق شد قوانین ناسازگار، تکراری و متناقض از مجموعه قوانین را حذف نماید. در گام بعد برای حل مشکلات فازی و به منظور دست یافتن به مجموعه قوانین بهینه، الگوریتم ژنتیک به کار گرفته شد و علاوه بر کاهش مجدد قوانین تکراری، بهبود قوانین فازی نیز حاصل شد (استخراج، اصلاح و کاهش قوانین در سه گام دنبال گردید). برای ارزیابی و مقایسه روش پیشنهادی دو گام صورت گرفت: در ابتدا نتایج به دست آمده با روش‌های رگرسیون، ماشین بردار پشتیبان، بیز ساده طبقه‌بندی و مقایسه شد و در گام دوم نتایج سایر مقالات با روش پیشنهادی مقایسه گردید. برای مقایسات روش‌ها بر روی هفت مجموعه داده معروف و پرکاربرد پیاده‌سازی شدند. نتایج نشان می‌دهد روش پیشنهادی قادر است در کنار کاهش، اصلاح و بهبود قوانین، دقت بدست آمده را در مقایسه با روش‌های مورد بررسی، بهبود دهد. برای ادامه کار، پیشنهاد می‌گردد که علاوه بر اصلاح قوانین توسط GA، پارامترها، نوع و تعداد متغیرهای زبانی مجموعه قوانین نیز اصلاح شود. همچنین می‌توان نتایج حاصل از دسته‌بندی‌های مختلف را بر روی مجموعه قوانینی که بعد از اصلاح بدست می‌آیند، پیاده‌سازی کرد و نتایج را با هم مقایسه نمود.

تعارض منافع: نویسندگان اعلام می‌کنند که هیچ تعارض منافی ندارند.

با توجه به نتایج شکل (۷) مشاهده می‌گردد که مدل رگرسیون بهتر جواب می‌دهد و دقت بالاتری دارد، در حالی که روش SVM جواب خوبی برای مجموعه داده‌ها تولید نکرده است. به منظور مقایسه روش پیشنهادی با سایر روش‌های دسته‌بندی کننده، جدول (۶) ذکر شده است.

جدول (۶): مقایسه روش پیشنهادی با دیگر روش‌ها

مجموعه داده	درصد صحت عملکرد مجموعه داده تست			
	رگرسیون	Naive	SVM	روش پیشنهادی GA-FUZZY
دیابت	۷۸	۷۲	۶۷	۷۱.۵۵
بیماری قلبی	۸۴	۸۳	۵۰	۵۶
پارکینسون	۷۷	۶۴	۷۶	۸۰
هپاتیت	۷۸	۷۶	۶۷	۸۸
Iris	۹۲	۶۲	۶۴	۹۲
پوکمون	۸۸	۶۴.۵	۷۰	۷۳.۲
تخصیص وام	۷۳	۶۴.۵	۷۵	۷۴

جدول (۷) مجموعه داده‌هایی را که با روش‌های انتخابی ما در جدول (۶) مورد مقایسه قرار گرفته و نتایج نسبتاً بهتری به دست نیاورده‌اند، با دیگر مقالات و روش‌ها مقایسه کرده است.

جدول (۷): مقایسه نتایج شبیه‌سازی شده با دیگر روش‌ها

مجموعه داده	روش (مرجع)	دقت	دقت روش پیشنهادی
Iris	PSO [29]	۶۷/۴۲	۹۲
دیابت	2slave [30]	۶۶/۴۵	۷۱.۵۵
بیماری قلبی	GP-Coach [31]	۵۵/۲۳	۵۶
بیماری قلبی	2slave [30]	۴۶/۲۲	۵۶
تخصیص وام	رگرسیون [32]	۷۳	۷۴

۶. نتیجه‌گیری

این مقاله روشی خودکار جهت مدیریت عدم قطعیت و استخراج دانش در قالب مجموعه قوانین از داده‌ها را پیاده‌سازی کرد. بدین منظور بعد از مجموعه عملیات

- [1] Bodenhofer U., "Tuning Of Fuzzy Systems Using Genetic Algorithms", A thesis submitted to Prof. Erich Peter Klement in partial satisfaction of the requirements for the obtainment of the academic degree of Diplom Ingenieur der Studienrichtung Technische Mathematik, 1996.
- [2] Cordón O., "A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems", *International Journal of Approximate Reasoning* 52 (6):894–913, 2011, <https://doi.org/10.1016/j.ijar.2011.03.004>.
- [3] De Jong K. A., "Learning with genetic algorithms: an overview", *Machine Learning* 3 (3):121–138, 1988, <https://doi.org/10.1007/BF00113894>.
- [4] Casillas J., Cordón O., Herrera F., Magdalena L. (Eds.), *Interpretability Issues in Fuzzy Modeling*, Springer, 2003, <https://doi.org/10.1007/978-3-540-37057-4>.
- [5] Ishibuchi H., Nakashima T., Murata T., "comparison of the Michigan and Pittsburgh approach to the design of classification systems", *electronic and communications in japan* 80 (12):10-19, 1997, [http://dx.doi.org/10.1002/\(SICI\)1520-6440\(199712\)80:123.0.CO;2-W](http://dx.doi.org/10.1002/(SICI)1520-6440(199712)80:123.0.CO;2-W).
- [6] Fernández A., del Jesus M. J., Herrera F., "On the 2-tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced data-sets", *Information Sciences* 180 (8):1268–1291, 2010, <https://doi.org/10.1016/j.ins.2009.12.014>.
- [7] Tan P.-N., Steinbach M. S., Karpatne A., Kumar V., *Introduction to Data Mining*, Pearson, 2nd Edition, 2019.
- [8] Verma N.K., Singh V., Rajurkar S., Aqib M., "Fuzzy Inference Network with Mamdani Fuzzy Inference System", In: Verma N., Ghosh A. (eds) *Computational Intelligence: Theories, Applications and Future Directions -Volume I. Advances in Intelligent Systems and Computing*, vol 798. Springer, 2019, https://doi.org/10.1007/978-981-13-1132-1_29.
- [9] Kumar A., Venkateswaran J., "Reverse Sequential Covering Algorithm for Medical Data Mining", *Procedia Computer Science* 47:109-117, 2015, <http://dx.doi.org/10.1016/j.procs.2015.03.189>.
- [10] Wang L.-X., Mendel J. M., "Generating fuzzy rules by learning from examples", *IEEE Transactions on Systems, Man, and Cybernetics* 22(6):1414-1427, 1992, <https://doi.org/10.1109/21.199466>.
- [11] Sanz J. A., Fernández A., Sola H. B., Herrera F., "A genetic tuning to improve the performance of Fuzzy Rule-Based Classification Systems with Interval-Valued Fuzzy Sets", *International Journal of Approximate Reasoning* 52(6):751–766, 2011, <https://doi.org/10.1016/j.ijar.2011.01.011>.
- [12] Cordona O., Gomide F. A. C., Herrera F., Hoffmann F., Magdalena L., "Ten years of genetic fuzzy systems: current framework and new trends", *Fuzzy Sets and Systems*, 141(1):5–31, 2004, [https://doi.org/10.1016/S0165-0114\(03\)00111-8](https://doi.org/10.1016/S0165-0114(03)00111-8).
- [13] Casillas J., Cordón O., del Jesus M. J., Herrera F., "Genetic Tuning of Fuzzy Rule Deep Structures Preserving Interpretability and Its Interaction With Fuzzy Rule Set Reduction", *IEEE Transaction on Fuzzy Systems* 13(1):13-29, 2005, <https://doi.org/10.1109/TFUZZ.2004.839670>.
- [14] Ishibuchi H., Nakashima T., Murata T. "A fuzzy classifier system that generates linguistic rules for pattern classification problems", *IEEE/Nagoya-University World Wisepersons Workshop*, 35-54, 1995, https://doi.org/10.1007/3-540-61988-7_15.
- [15] Herrera F. "Genetic fuzzy systems: taxonomy, current research trends and prospects", *Evolutionary Intelligence*, 1(1):27-46, 2008, <https://doi.org/10.1007/s12065-007-0001-5>.
- [16] Castro P. A. D., Camargo H. D. A., "Improving the genetic optimization of fuzzy rule base by imposing a constraint condition on the number of rules", *congresso da sociedade brasileira de computacao* December 2004.
- [17] <https://www.kaggle.com/datasets>
- [18] <https://archive.ics.uci.edu/ml/datasets.php>
- [19] Tan C. H., Yap K. S., Yap H. J., "Application of genetic algorithm for fuzzy rules optimization on semi expert judgment automation using Pittsburg approach", *Applied Soft Computing* 12(8): 2168–2177, 2012, <https://doi.org/10.1016/j.asoc.2012.03.018>.
- [20] Soui M., Gasmi I., Smiti S., Ghédira K., "Rule-based credit risk assessment model using multi-objective evolutionary algorithms", *Expert Systems With Applications* 126:144–157, 2019, <https://doi.org/10.1016/j.eswa.2019.01.078>.
- [21] Han J., Kamber M., Pei J., "Data Mining Concepts

- and Techniques”, Third Edition, the Morgan Kaufmann Series in DataManagement Systems, 2011.
- [22] Kulluk S., Özbakır L., Baykasoğlu A., “Fuzzy difaconn-miner: a novel approach for fuzzy rule extraction from neural networks” *Expert Syst. Appl.*, 40(3): 938-946, 2013, <https://doi.org/10.1016/j.eswa.2012.05.050>.
- [23] Ganapathy S., Sethukkarasi R., Yogesh P., Vijayakumar P., Kannan A., ”An intelligent temporal pattern classification system using fuzzy temporal rules and particle swarm optimization”, *Sadhana*, 39(2): 283-302, 2014, <https://doi.org/10.1007/s12046-014-0236-7>.
- [24] Koshiyama A.S., Vellasco M.M.B.R., Tanscheit R., “GPFIS-class. A genetic fuzzy system based on genetic programming for classification problems”, *Appl. Soft Comput.*, 37:561-571, 2015, <https://doi.org/10.1016/j.asoc.2015.08.055>.
- [25] Zhu L., Qiu D., Ergu D., Ying C., Liu K., “A study on predicting loan default based on the random forest algorithm”, 7th International Conference on Information Technology and Quantitative Management, ITQM 2019, <https://doi.org/10.1016/j.procs.2019.12.017>.
- [26] Safari A., Mazinani M., Hossein R., “A Novel Type-2 Adaptive Neuro Fuzzy Inference System Classifier for Modelling Uncertainty in Prediction of Air Pollution Disaster”, *IJE* 30(11):1746-1751, 2017.
- [27] Hosseini R., Qanadli S. D., Barman S., Mazinani M., Ellis T., Dehmeshki J., “An Automatic Approach for Learning and Tuning Gaussian Interval Type-2 Fuzzy Membership Functions Applied to Lung CAD Classification System”, *IEEE Transactions on Fuzzy Systems* 27(8), 2019, <https://doi.org/10.1109/TFUZZ.2019.2921503>.
- [۲۸] آخوندی ر.، حسینی ر.، «ارایه مدل هوشمند هایپریدی فازی-تکامل ژنتیکی تفاضلی در یک سیستم خبره فازی برای پیش بینی خطر ابتلا به بیماری قلبی»، *مجله محاسبات نرم*، جلد ۶، شماره ۲، ص ۳۲-۴۷، ۱۳۹۶.
- [۲۹] قاسم‌احمد ل.، «مروری بر ۷ الگوریتم برتر داده کاوی در پیش‌بینی مبتلایان به سرطان»، *فصلنامه بیماری‌های پستان ایران*، سال ششم، شماره اول، بهار ۱۳۹۲.
- [۳۰] لنگری‌زاده م.، مقبلی ف.، «مرور نظام‌مند کاربرد شبکه بیزین ساده در پیش‌بینی بیماری‌ها»، *مجله انفورماتیک سلامت و زیست پزشکی*، دوره سوم، شماره چهارم، ۱۳۹۵.
- [۳۱] مرادی فراهانی ح.، عسگری ج.، ذکری م.، «مروری بر منطق فازی نوع-۲: از پیدایش تا کاربرد»، *مجله محاسبات نرم*، جلد ۲، شماره ۱، ص ۲۲-۴۳، ۱۳۹۲.
- [۳۲] قاسمی ر.، محمدی ح.، طاهر س.ع.، «کنترل فرکانس یک زیرشبکه جزیره‌ای با استفاده از کنترل هوشمند پاسخ‌گویی بار مبتنی بر منطق فازی و الگوریتم بهینه سازی ازدحام ذرات»، *مجله محاسبات نرم*، جلد ۶، شماره ۲، ص ۱۸-۳۱، ۱۳۹۶.