

ارائه یک روش فازی-تکاملی برای تشخیص خطاها در نرم افزار

مهدی افتخاری^{*}، مریم مجیدی مؤمن آبادی و مجتبی خمر

بخش مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه شهید باهنر کرمان، کرمان، ایران

چکیده

تشخیص خطاها در نرم افزار، یکی از بزرگ ترین چالش‌های توسعه نرم افزار است و بیشترین بودجه را در فرآیند توسعه نرم افزار به خود اختصاص می‌دهد. با توجه به اهمیت تشخیص خطاها در نرم افزار، در این مقاله روشی بر مبنای مجموعه‌های فازی و الگوریتم‌های تکاملی ارائه می‌شود. از آنجا که ماهیت مجموعه‌داده‌های تشخیص خطا نرم افزار نامتوازن است، از مزایای الگوریتم‌های خوشبندی فازی به منظور نمونه‌برداری از داده‌ها و توجه بیشتر به طبقه‌اقلیت استفاده شده است. روش پیشنهادی درواقع یک الگوریتم ترکیبی است که در ابتدا از روش خوشبندی ۴ میانگین فازی به منظور نمونه‌برداری بوت استرالیا وزن دار استفاده می‌شود. وزن داده‌ها همان درجه عضویت آنهاست و درجه عضویت داده‌های طبقه‌اقلیت افزایش می‌یابد. در گام بعدی، از الگوریتم خوشبندی کاهشی برای ایجاد طبقه‌بند استفاده می‌شود که توسط داده‌های تولید شده در مرحله قبل آموزش می‌یابند؛ همچنین از الگوریتم ژنتیک دودویی برای انتخاب ویژگی‌های مناسب استفاده می‌شود. نتایج به دست آمده و همچنین مقایسه آنها با چندین روش معروف در این زمینه، کارایی مناسب روش پیشنهادی را نشان می‌دهد. برای انجام آزمایش‌ها از ده پایگاه داده معروف با گستره وسیعی از اندازه و نرخ عدم توازن، استفاده شده است و برای تأیید نتایج از آزمون آماری تی بهره برده ایم.

واژگان کلیدی: الگوریتم‌های تکاملی، تشخیص خطا نرم افزار، طبقه‌بندی، مجموعه‌داده‌های نامتوازن، منطق فازی.

Proposing an evolutionary-fuzzy method for software defects detection

Mahdi Eftekhari, Maryam Majidi momenabadi & Mojtaba Khamar

Department computer engineering, Shahid Bahonar University of Kerman, Kerman, Iran.

Abstract

Software defects detection is one of the most important challenges of software development and it is the most prohibitive process in software development. The early detection of fault-prone modules helps software project managers to allocate the limited cost, time, and effort of developers for testing the defect-prone modules more intensively. In this paper, according to the importance of software defects detection, a method based on fuzzy sets and evolutionary algorithms is proposed. Due to the imbalanced nature of software defect detection datasets, benefits of fuzzy clustering algorithms were used to data sampling and more attention to the minority class. This method is a combined algorithm which, firstly has used fuzzy c-mean clustering as weighted bootstrap sampling. Weight of data (their membership's degrees) increases for minority class. In the next step, the subtractive clustering algorithm is applied to produce the classifier which was trained by produced data in the previous step. The binary genetic algorithm was utilized to select appropriate features. The results and also comparisons with eight popular methods in software defect detection literature, show an acceptable performance of the proposed method. The experiments were performed on ten real-world datasets with a wide range of data sizes and imbalance rates. Also T-test is used as the statistical significance test for pair wise comparison of our proposed method against the others. The final results of T-test are shown in tables

* نویسنده عهده‌دار مکاتبات • تاریخ ارسال مقاله: ۱۳۹۶/۶/۲۴ • تاریخ آخرین بازنگری: ۱۳۹۷/۶/۱۴ • تاریخ پذیرش: ۱۳۹۷/۱۰/۱۹



for three performance measures (G-mean, AUC and Balanced) over various datasets. (As the obtained results apparently show our proposed method has the ability to improve three aforementioned performance criteria simultaneously). Some methods just have improved the G-mean measure while the AUC and Balance criteria have lower values than the others. Securing a high level of three performance measures simultaneously illustrates the ability of our proposed algorithm for handling the imbalance problem of software defects detection datasets.

Keywords: classification, evolutionary algorithm, fuzzy logic, imbalance datasets, software defect detection.

مسئله تشخیص خطا نرم افزار، یک نمونه خطا نسبت به یک نمونه بدون خطا، به احتمال کمتری اتفاق میافتد؛ بنابراین داده های تشخیص خطا نرم افزار نامتوازن هستند؛ به این معنا که تعداد نمونه های طبقه بدون خطا بیشتر از تعداد نمونه های طبقه خطادار است؛ بنابراین تمرکز این مقاله بر روی مسئله طبقه بندی با توزیع نامتوازن طبقه است.

مدل های تشخیص خطا نرم افزار می توانند مازول های نرم افزاری را به دو طبقه مازول های خطادار و بدون خطا اختصاص دهند؛ اما استفاده از طبقه بندی های استاندارد برای مسائل نامتوازن مناسب نیست؛ زیرا اغلب الگوریتم های یادگیری ماشین فرض می کنند که تعداد نمونه های آموزشی در طبقه های مختلف برابر است. درنتیجه طبقه بند یادگرفته شده اغلب از طبقه اکثریت نتیجه می شود که این موضوع بهدلیل آن که آموزش طبقه اقلیت به درستی انجام نشده است به پیش بینی بسیار ضعیف از طبقه اقلیت منجر می شود. بنابراین کنترل و حل مسئله داده نامتوازن برای بهبود کارایی پیش بینی کننده های خطا نرم افزار امری ضروری است. روش های عددی برای حل مسائل داده نامتوازن به دو دسته اصلی تقسیم می شوند:

۱. نمونه گیری از داده که خود به سه دسته زیاد کردن طبقه اقلیت، کم کردن طبقه اکثریت و روش های ترکیبی تقسیم می شود.

۲. اصلاح الگوریتم های موجود یا ارائه یک الگوریتم جدید که شامل سه روش یادگیری حساس به هزینه، یادگیری مبتنی بر هسته و یادگیری مبتنی بر تشخیص است [5]. در صورتی که یک نمونه، خطادار باشد، ولی با اشتباه آن را بدون خطا طبقه بندی کنیم، کیفیت نرم افزار به شدت پایین می آید. در حالی که طبقه بندی اشتباه یک نمونه بدون خطا به عنوان خطادار، هزینه آزمایش فقط یک مازول را در بی خواهد داشت. بنابراین هزینه طبقه بندی اشتباه طبقه اقلیت، بسیار بیشتر از هزینه طبقه بندی اشتباه طبقه اکثریت است. با توجه به این مسئله، بسیار مهم است که بتوانیم طبقه خطادار یا همان طبقه اقلیت را به درستی پیش بینی کنیم [1].

۱- مقدمه و پیشینه

در سال ۲۰۰۵ که منبع داده پرومیس^۱ تولید شد، این امکان به پژوهش گران داده شد که بتوانند مطالعات و آزمایش های خود را با هم مقایسه کنند. در این منبع داده، مجموعه داده های خطا نرم افزار وجود دارد که بر اساس ویژگی های مختلف نرم افزار، خطا های نرم افزار را در دسته های مختلف دسته بندی می کنند. پس از آن، پژوهش ها در زمینه تشخیص خطا نرم افزار بسیار افزایش پیدا کرد [1]. با افزایش اندازه و پیچیدگی سامانه های نرم افزاری، صنعت نرم افزار با چالش تحويل سامانه با کیفیت، قابل اعتماد، به موقع و با هزینه تعیین شده مواجه شد. برای کیفیت، تعاریف زیادی وجود دارد؛ اما یکی از تعاریف قابل قبول برای کیفیت بالا، خطا های نرم افزاری به طور معمول به عنوان انحراف از خصوصیات مورد انتظار است که منجر به شکست هایی در عمل در زمان آینده می شود.

برای کم کردن خطا های سامانه های نرم افزاری به آزمایش نرم افزار نیاز است؛ اما حدود ۵۰ تا ۶۰ درصد هزینه توسعه نرم افزار به آزمایش مربوط می شود [3]. به دلیل هزینه بربودن آزمایش کلیه مازول های یک سامانه نرم افزاری بزرگ، لازم است، فقط مازول هایی که احتمال خطا بیشتری دارند، آزمایش شوند؛ بنابراین احتیاج به تشخیص مازول های مستعد خطا^۲ پیدا شد. با تشخیص مازول های مستعد خطا در ابتدای یک پروژه نرم افزاری، مدیران پروژه های نرم افزاری می توانند زمان، هزینه و تلاش تیم توسعه را بیشتر صرف آزمایش مازول های مستعد خطا کنند و دیگر نیاز به آزمایش کل سامانه نرم افزاری نیست. آنها همچنین می توانند برای پیشبرد پروژه نرم افزاری برنامه ریزی، کنترل و اجرای بهتر داشته باشند و با تحويل سامانه با کیفیت، با هزینه تعیین شده و به موقع رضایت مشتری را بیشتر جلب می کنند [4]. نکته بسیار مهم این است که مازول های مستعد خطا، بخش کوچکی از کل سامانه نرم افزار را تشکیل می دهند. به عبارت دیگر، به دلیل ماهیت

¹ PROMISE

² Defect-prone modules



تعداد زیادی خطا هستند، استفاده می‌شود، در مقایسه با دیدگاه رگرسیون بردار پشتیبان، رگرسیون بردار پشتیبان فازی می‌تواند میانگین مربعات خطای کمتر و دقت بیشتری در تعداد کل خطاهای به دست آورد. رسّلی و همکارانش در سال ۲۰۱۱ با استفاده از الگوریتم ژنتیک کاربردی برای پیش‌بینی خطای نرم‌افزار طراحی [۱۰] و برای انجام آزمایش‌های خود، از معیارهای یک نرم‌افزار منبع باز به عنوان ورودی الگوریتم ژنتیک استفاده کردند. هدف آنها از طراحی پیشنهادشده، توسعه یک ابزار خودکار برای توسعه دهنده‌گان نرم‌افزار است که بتوانند تشخیص دهنده‌کدام مژول‌های نرم‌افزاری احتمال بیشتری دارد که در آینده باعث ایجاد مشکل شوند.

بخش‌های بعدی مقاله به شرح زیر ارائه خواهد شد: در بخش دو به توضیح در مورد روش خوش‌بندی فازی استفاده شده در این مقاله پرداخته می‌شود. در بخش سه روش پیشنهادی توضیح داده شده است. در بخش چهار به تحلیل و بحث در مورد نتایج پرداخته می‌شود و در بخش پنجم با نتیجه‌گیری که حاصل بررسی روش پیشنهادی و دیگر روش‌ها می‌باشد مقاله پایان می‌یابد.

۲- خوش‌بندی فازی

خوش‌بندی یکی از شاخه‌های یادگیری بدون نظارت، و فرآیند خودکاری است که در طی آن، نمونه‌ها به دسته‌هایی که اعضای آن مشابه یکدیگرند، تقسیم می‌شوند که به این دسته‌ها خوش‌گفته می‌شود. برای مشابه‌بودن می‌توان معیارهای مختلفی مثل معیار فاصله را در نظر گرفت که در آن اشیای نزدیک‌تر به هم به عنوان یک خوش‌بندی مبتنی بر فاصله می‌شود. به این نوع خوش‌بندی، خوش‌بندی مبتنی بر فاصله نیز گفته می‌شود. در خوش‌بندی کلاسیک، هر نمونه ورودی متعلق به یک و فقط یک خوش‌بندی است و نمی‌تواند عضو دو خوش‌بندی باشد و به زبان دیگر خوش‌ها هم پوشانی ندارند. حال، حالتی را در نظر بگیرید که میزان تشابه یک نمونه با دو خوش‌بندی می‌باشد. تفاوت اصلی خوش‌بندی کلاسیک و خوش‌بندی فازی در این است که در خوش‌بندی فازی، یک نمونه می‌تواند متعلق به بیش از یک خوش‌بندی باشد. چنان‌چه ایده روشی در مورد تعداد خوش‌ها وجود نداشته باشد، استفاده از خوش‌بندی کاوشی روشی سریع برای حل این موضوع محسوب می‌شود. گاهی اوقات مراکزی که با این روش تخمین زده شده‌اند، به عنوان نقاط اولیه برای دیگر الگوریتم‌های خوش‌بندی مورد استفاده قرار می‌گیرند. خوش‌بندی کاوشی که خوش‌بندی تفاضلی نیز

مدلهای پیش‌بینی خطا که رابطه بین معیارهای نرم‌افزار و مژول‌های مستعد خطا را پیدا می‌کنند، از روش‌های مختلف پیش‌بینی استفاده کرده‌اند. از جمله مهم‌ترین و متداول‌ترین آنها می‌توان به روش‌های طبقه‌بندی فازی [۵]-[۷]، الگوریتم‌های تکاملی [۱۰]-[۸]، شبکه‌های عصبی مصنوعی [۱۳]-[۷]-[۵]-[۲]، ماشین‌های بردار پشتیبان [۸]-[۱۹]-[۱۰] اشاره کرد.

وانگ و همکارش در سال ۲۰۱۲ در مورد این مسئله پژوهش کردند که چه‌طور روش‌های یادگیری طبقه‌های نامتوازن می‌توانند به حل مسئله تشخیص خطای نرم‌افزار کمک کنند [۱]. آنها انواع مختلف روش‌های یادگیری طبقه‌های نامتوازن را (در قالب هفت الگوریتم) بررسی کردند که شامل سه نوع روش نمونه‌گیری مجدد، روش‌های حرکت حد آستانه^۱ و الگوریتم‌های جمعی^۲ می‌شوند. برای به دست آوردن بیشینه پتانسیل هر یک از روش‌ها، در ابتدا آنها بهترین مقادیر پارامترها را برای هر روش جستجو کردند. این جستجو برای به دست آوردن بهترین پارامترها مبتنی بر معیارهای تعادل، میانگین هندسی و سطح زیر نمودار^۳ انجام شد؛ سپس بهترین مقادیر پارامترها برای هر روش تنظیم و الگوریتم‌ها با یکدیگر مقایسه شدند. نتیجه مقایسه این روش‌ها این بود که الگوریتم آدابوست ان سی^۴ بهترین کارایی را در معیارهای میانگین هندسی، سطح زیر نمودار و تعادل به دست آورد. آنها برای ارتقای کارایی الگوریتم آدابوست ان سی، یک نسخه پویا از آن پیشنهاد دادند و نام الگوریتم پیشنهادی خود را آدابوست ان سی پویا^۵ گذاشتند. الگوریتم آنها قادر است، در حین یادگیری، پارامترهای خود را تنظیم کند؛ بدون آن که نیاز باشد هیچ پارامتری از قبل تعریف شود. در انتها آنها الگوریتم خود را با الگوریتم‌های آدابوست ان سی و بیز ساده^۶ مقایسه کردند. آنها گزارش دادند که روش پیشنهادی شان کارآمدتر و مؤثرتر از آدابوست ان سی اصلی و بیز ساده است. یان و همکارانش در سال ۲۰۱۰ یک روش جدید با استفاده از رگرسیون بردار پشتیبان فازی^۷ برای تشخیص تعداد خطاهای نرم‌افزار پیشنهاد کردند [۶] و رودی فازی‌ساز رگرسور می‌تواند مجموعه داده‌های معیارهای نرم‌افزار نامتعادل را مدیریت کند. نتایج آزمایش‌های آنها روی مجموعه داده‌های MIS و RSDIMU نشان داد وقتی از مژول‌هایی که شامل

^۱ Threshold-moving

^۲ Ensemble learning

^۳ Area Under Curve(AUC)

^۴ AdaBoost.NC

^۵ Dynamic version of AdaBoost.NC(DNC)

^۶ Naive Bayes

^۷ Fuzzy Support Vector Regression (FSVR)

$$z = \frac{\sum_{i=1}^N w_i f_i}{\sum_{i=1}^N w_i} \quad (6)$$

که w_i قدرت شلیک^۳ قانون i ام را بیان می‌کند.

۲-۱- خوشبندی فازی c - میانگین

مشابه الگوریتم c - میانگین کلاسیک در این الگوریتم نیز تعداد خوشبندیها (c) از قبل مشخص شده است.تابع هدفی که برای این الگوریتم تعریف شده به صورت زیر است [15], [20]:

$$\begin{aligned} J &= \sum_{i=1}^c \sum_{k=1}^n u_{ik} d_{ik}^2 \\ &= \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m \|x_k - v_i\|^2 \end{aligned} \quad (7)$$

در فرمول بالا m یک عدد حقیقی بزرگ‌تر از یک است که در بیشتر موارد برای m عدد دو انتخاب می‌شود. x_k نمونه k ام و v_i نماینده یا مرکز خوشبندی i ام است. u_{ik} میزان تعلق نمونه k ام در خوشبندی i ام را نشان می‌دهد. علامت $\| \cdot \|$ میزان تشابه (فاصله) نمونه با (از) مرکز خوشبندی است که می‌توان از هر تابعی که بیان‌گر تشابه نمونه و مرکز خوشبندی باشد، استفاده کرد. از روی u_{ik} می‌توان یک ماتریس U تعریف کرد که دارای c سطر و n ستون است و مؤلفه‌های آن می‌توانند هر مقداری بین صفر تا یک را اختیار کنند. اگر تمامی مؤلفه‌های ماتریس U به صورت صفر و یا یک باشند، الگوریتم مشابه c - میانگین کلاسیک خواهد بود. با این‌که مؤلفه‌های ماتریس U می‌توانند هر مقداری بین صفر تا یک را اختیار کنند، اما مجموع مؤلفه‌های هر یک از ستون‌ها باید برابر یک باشد و داریم [6], [20]:

$$\sum_{i=1}^c u_{ik} = 1, \forall k = 1, \dots, n \quad (8)$$

معنای این شرط این است که مجموع تعلق هر نمونه به c خوشبندی باید برابر یک باشد. با استفاده از شرط بالا و کمینه‌کردن تابع هدف خواهیم داشت [20]:

$$u_{ik} = \left(\frac{1}{\sum_{j=1}^c \frac{d_{jk}}{d_{ik}}} \right)^{2/m-1} \quad (9)$$

$$v_i = \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m} \quad (10)$$

² Firing strength

نامیده می‌شود، در اصل یک فرم تغییریافته از روش خوشبندی کوهستان^۱ است. در این الگوریتم هر نقطه به عنوان یک پتانسیل برای مرکز خوشبندی در نظر گرفته می‌شود که اندازه‌گیری پتانسیل طبق معادله زیر به دست می‌آید [6]:

$$P_i = \sum_{j=1}^N e^{\alpha \|x_i - x_j\|^2} \quad (1)$$

$$\alpha = \frac{4}{(r_a)^2} \quad (2)$$

که x_i بیان‌گر نمونه i ام است. همچنین مقدار α طبق رابطه (2) به دست می‌آید. $r_a > 0$ مقداری مثبت است که به عنوان شاعع همسایگی برای هر مرکز خوشبندی تعریف می‌شوند. بنابراین پتانسیل تخصیص داده شده به هر خوشبندی به فاصله آن از نقاط دیگر واپسی است و در مواردی با همسایه مترکم، منجر به خوشبندی باشد. بعد از محاسبه پتانسیل برای هر نقطه، یک نقطه که دارای بالاترین پتانسیل است به عنوان مرکز خوشبندی انتخاب می‌شود. ابتدا x_1 به عنوان مرکز نخستین گروه با پتانسیل P_1 تعیین می‌شود. سپس پتانسیل P_i برای x_i مطابق معادله (3) کاهش می‌باید:

$$P_i = P_i - p_1 e^{(-\beta \|x_i - x_1\|^2)} \quad (3)$$

$$\beta = \frac{4}{r_b^2} \quad (4)$$

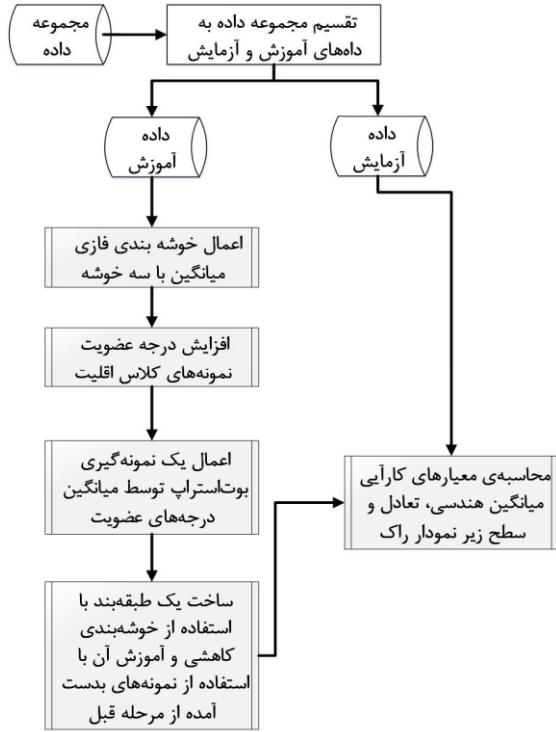
که مقدار β از رابطه (4) به دست می‌آید و $r_b > 0$ عددی مثبت است که همسایه‌ای را با کاهش قابل توجهی در چگالی تعیین می‌کند. بنابراین، مقدار چگالی نقطه داده‌ای که نزدیک به نخستین خوشبندی است، به طور قابل توجهی کاهش می‌باید. این روش (انتخاب مرکز و کاهش پتانسیل) به کزار اجرا شده تا ضوابط و معیارهای توقف تأمین شود. خروجی خوشبندی کاهشی، یک سامانه استنتاج فازی سوگنو است. یک قانون سوگنو به صورت زیر تعریف می‌شود [6]:

$$\text{Rule1: If } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ then } f_1 = p_1 x + q_1 y + r_1 \quad (5)$$

که x, y متغیرهای ورودی، A_1, B_1 مجموعه‌های فازی و f_1 تابع خروجی را نشان می‌دهد. خروجی نهایی برای یک نمونه در یک سامانه فازی سوگنو با استفاده از روش میانگین وزن دار که در رابطه (6) نشان داده شده است، به دست می‌آید.

¹ Mountain





(شکل-۱): روندnamای روش پیشنهادی
(Figure-1): Flow chart of proposed method

۴. درجه عضویت‌های اصلاح شده، به عنوان پارامترهای ورودی به الگوریتم نمونه‌گیری بوتاستریپ داده می‌شوند. اضافه کردن وزن ممکن است، باعث افزایش درجه‌های عضویت به مقداری بیش از یک شود؛ بنابراین در الگوریتم بوتاستریپ، ابتدا درجه‌های عضویت اصلاح شده را نرمال می‌کنیم که به مفهوم نگاشت مقادیر بین صفر و یک است؛ سپس احتمال هر درجه عضویت اصلاح شده و نرمال شده را از طریق رابطه (۱۲) به دست می‌آوریم.

$$pr = \frac{max_{un}}{sum(max_{un})} \quad (12)$$

که در آن max_{un} درجه عضویت اصلاح و نرمال شده است؛ سپس احتمال تجمعی را برای احتمالات درجه‌های عضویت، با استفاده از رابطه (۱۳) محاسبه می‌کنیم.

$$Cpr(i) = Cpr(i - 1) + pr(i) \quad (13)$$

سپس با انتخاب چرخ گردان، اندیس نمونه‌های انتخابی را به دست می‌آوریم. ۵. با استفاده از خوشبندی کاهشی و داده‌های آموزشی، طبقه‌بند ساخته می‌شود. این طبقه‌بند در واقع یک سامانه استنتاج سوگنی فازی است که از تصویر کردن خطاها روی ابعاد مختلف به دست می‌آید. از آنجا که خروجی سامانه‌های سوگنی به طور معمول مقدار پیوسته‌ای است،

۳- روش پیشنهادی

بهینه‌سازی، یکی از مهم‌ترین موضوعات در زمینه مهندسی محسوب می‌شود و هدف آن، تنظیم ورودی‌های مسئله و ایجاد نتایج بهینه است. در بسیاری از مسائل، انتخاب بهترین جواب از طریق جستجوی کل فضای مسئله عملی زمانبر است؛ بنابراین یک راه حل مناسب، استفاده از الگوریتم‌های تکاملی است. بر این اساس در روش پیشنهادی، ابتدا از روش‌های خوشبندی فازی برای تولید یک مدل استنتاج فازی از نوع سوگنی جهت تشخیص خطای نرم‌افزار استفاده و سپس با استفاده از الگوریتم‌های تکاملی مدل‌های به دست آمده بهینه می‌شود. این کار از طریق تنظیم پارامترهای مدل‌های فازی انجام خواهد شد. روندnamای روش پیشنهادی در شکل (۱) آمده است. در ادامه به توضیح مراحل مختلف روش پیشنهادی می‌پردازیم.

۱. داده‌ها به روش اعتبارسنجی متقابل دهتایی^۱ تقسیم شده‌اند. بنابراین اجرای الگوریتم پیشنهادی با رویه دهتایی، ۵۰ مرتبه تکرار شد. هم‌چنین برای به دست آوردن نتایج قابل اعتمادتر، رویه پیشنهادی نیز ۵۰ مرتبه تکرار شده است.

۲. خوشبندی c میانگین فازی با تعداد خوشه‌های سه اجرا شد. البته در ابتدا با توجه به این که هدف اولیه، خوشبندی مجموعه داده به دو دسته بود، یک دسته برای نمونه‌های طبقه اکثریت و دیگری برای نمونه‌های طبقه اقلیت، تعادل خوشه‌ها دو درنظر گرفته شد؛ اما بعد از چندین سعی و خطا، بهترین نتایج با تعداد خوشه‌های برابر سه به دست آمد.

۳. درجه‌های عضویت نمونه‌ها در خوشه‌ها برای اجرای نمونه‌گیری بوتاستریپ وزن دار در مرحله بعد اصلاح شدند. از این‌رو درجه‌های عضویت نمونه‌های اقلیت توسط نرخ عدم توازن افزایش یافتند. این افزایش درجه عضویت بر اساس رابطه زیر صورت می‌گیرد:

$$\mu(x_i) = \mu(x_i) + (IR \times Coef) \quad (11)$$

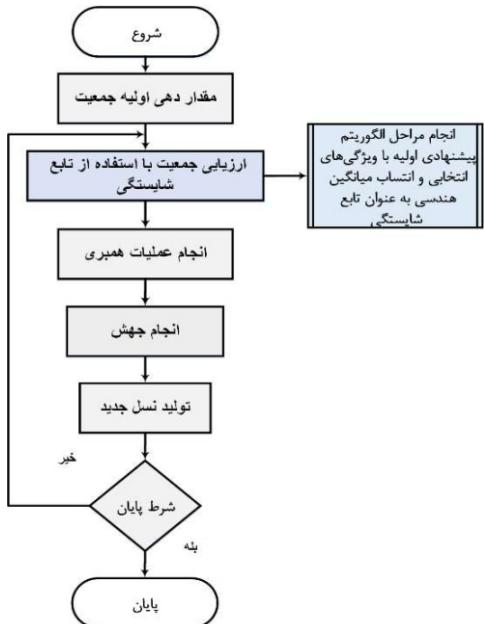
به طوری x_i نمونه‌ای است که به طبقه اقلیت تعلق دارد و IR نرخ عدم توازن که برابر است با نسبت تعداد نمونه‌های طبقه اکثریت به تعداد نمونه‌های کلاس اقلیت. مسئله تشخیص خطا جزو مسائل نامتوازن و تعداد نمونه‌های طبقه اکثریت همواره بسیار بیشتر از تعداد نمونه‌های طبقه اقلیت است؛ بنابراین نرخ عدم توازن مقداری بیشتر از یک خواهد بود. مقدار $Coef$ را در بیشتر مجموعه داده‌ها برابر با 0.5 در نظر گرفتیم.

^۱ 10 fold cross validation

کردیم. برای به دست آوردن بهترین انتخاب ویژگی در هر مجموعه داده، الگوریتم ژنتیک استفاده شده است. در شکل (۲)، روند نمای الگوریتم ژنتیک استفاده شده آمده و سپس توضیحات بیشتر در مورد آن بیان شده است.

این الگوریتم برای هر ده مجموعه داده به طور جداگانه انجام می شود. طول هر کروموزوم در الگوریتم ژنتیک، به تعداد ویژگی های هر مجموعه داده است. به ازای هر ویژگی در مجموعه داده، صفر و یک در کروموزوم کد می شود. صفر به معنای انتخاب نشدن ویژگی و یک بدین معناست که ویژگی منتظر آن، انتخاب می شود. در ابتدا به طور تصادفی اعداد صفر و یک در کروموزوم کد می شود؛ سپس الگوریتم پیشنهادی اولیه با ویژگی های انتخاب شده در کروموزوم، اجرا می شود. در هر بار اجرا، شایستگی افراد جمعیت محاسبه می شود. همان طور که در شکل (۲) نشان داده شده است، برای محاسبه شایستگی افراد جمعیت، الگوریتم پیشنهادی اولیه دو مرتبه اجرا که در هر مرتبه از روش اعتبارسنجی متقابل دستاورد استفاده و سپس میانگین مقادیر میانگین هندسی برای محاسبه تابع شایستگی برگردانده می شود. تابع شایستگی طبق رابطه (۱۹) در نظر گرفته شده است که باید کمینه شود. این بدان معناست که مقدار میانگین هندسی باید بیشینه شود.

$$Fitness = 1 - Gmean \quad (19)$$



(شکل-۲): روند نمای الگوریتم ژنتیک استفاده شده در این مقاله
(Figure-2): Flow chart of genetic algorithm used in this paper

در جدول (۱) پارامترهای الگوریتم ژنتیک استفاده شده آورده شده است.

به منظور گسترش سازی خروجی از تابع جزء صحیح استفاده می کنیم. همچنین می توان از تابع سیگموید^۱ جهت مقیاس کردن خروجی سامانه فازی بین صفر و یک استفاده کرد و بدین ترتیب از سامانه سوگنوی فازی به عنوان یک طبقه بند استفاده کرد.

تعداد قوانین در سامانه سوگنوی فازی تولید شده توسط خوشبندی کاهشی به شعاع خوشبندی وابسته است. مقادیر شعاع بزرگ باعث تولید قوانین کمتر و مقادیر کوچک، باعث تولید قوانین بیشتر می شود. در این مقاله برای هر قانون، دو خروجی به صورت زیر در نظر گرفته شده است. دلیل این کار، راحت بودن در استفاده از ماتریس گیجی بوده است.

If x_1 is A_1 and x_2 is B_1 then $y_1^{(1)} = 0.2x_1 - 0.3x_2 - 1$ and $y_2^{(1)} = 0.3x_1 + 0.4x_2 - 2$.

If x_1 is A_2 and x_2 is B_2 then $y_1^{(2)} = 5x_1 - 3x_2 + 4$ and $y_2^{(2)} = 4x_1 + 5x_2 - 5$.

$$W_1 = \mu_{A_1}(x_1) \cdot \mu_{B_1}(x_2) \quad W_2 = \mu_{A_2}(x_1) \cdot \mu_{B_2}(x_2)$$

$$Y_1^{Total} = \frac{W_1 y_1^{(1)} + W_2 y_1^{(2)}}{W_1 + W_2}$$

$$Y_2^{Total} = \frac{W_1 y_2^{(1)} + W_2 y_2^{(2)}}{W_1 + W_2}$$

درنهایت طبقه خروجی با استفاده از دو مقدار Y_1^{Total} و Y_2^{Total} به صورت زیر تعیین می شود:

$$Out_1 = sigmoid(Y_1^{Total}) \quad (14)$$

$$Out_2 = sigmoid(Y_2^{Total}) \quad (15)$$

مقادیر $[Out_1, Out_2]$ در کنار یکدیگر طبقه پیشنهادی را تعیین می کنند. اگر این مقادیر $[1, 0]$ باشند، طبقه نخست و اگر $[0, 1]$ باشند، طبقه دوم را تعیین می کنند. ۶ در آخرین مرحله، مقادیر معیارهای کارایی برای روش پیشنهادی طبق روابط زیر محاسبه می شود^[۱]:

$$G - mean = \sqrt{Tpr \cdot Tnr} \quad (16)$$

$$AUC = \frac{1 + Tpr - Fpr}{2} \quad (17)$$

$$Balance = 1 - \frac{\sqrt{(1 - Tnr)^2 + (1 - Tpr)^2}}{\sqrt{2}} \quad (18)$$

در الگوریتم پیشنهادی اولیه از تمام ویژگی های مجموعه داده ها، استفاده کردیم. برای بهبود این روش و درنتیجه به دست آوردن نتایج بهتر، به جای استفاده از تمام ویژگی های مجموعه داده ها، مهم ترین ویژگی ها را انتخاب

^۱ Sigmoid



می‌کنیم. برای تحلیل نتایج گزارش شده در این مقاله، از آزمون تی^۹ با دو نمونه مستقل که جزو مدل‌های آماری پارامتریک است، استفاده و همچنین برای ارزیابی کارایی روش‌ها و نشان‌دادن عملی بودن نتایج ارائه شده در این پژوهش، از سه معیار تعادل، سطح زیر نمودار و میانگین هندسی استفاده کردیم. این معیارها روی توزیع طبقه‌داده‌ها حساس هستند و بنابراین برای مسئله تشخیص خطا مناسب هستند. هر چه مقدار این سه معیار بیشتر باشد، پیش‌بینی کننده عملکرد بهتری دارد.

(جدول-۲): مجموعه داده‌های پرمیس استفاده شده در ساخت مدل‌های پیش‌بینی

(Table-2): PROMISE dataset utilized in obtaining proposed models

%خطا	تعداد ویزگی	تعداد نمونه	زبان برنامه‌نویسی	مجموعه داده
۳۲/۲۹	۳۹	۱۶۱	C++	mc2
۲۰/۴۹	۲۱	۵۲۲	C++	kc2
۱۹/۳۵	۲۱	۱۰۸۸۵	C	jml
۱۵/۴۵	۲۱	۲۱۰۹	C++	kc1
۱۲/۲۰	۳۷	۱۴۵۸	C	pc4
۱۰/۲۳	۳۷	۱۵۶۳	C	pc3
۹/۸۳	۲۱	۴۹۸	C	cm1
۹/۳۸	۳۹	۴۵۸	Java	kc3
۷/۶۹	۳۷	۴۰۳	C	mw1
۶/۹۴	۲۱	۱۱۰۹	C	pc1

جدول‌های ۳، ۴ و ۵ نشان‌گر میانگین و انحراف معیار معیارهای میانگین هندسی، تعادل و سطح زیر نمودار است که توسط نه روش یادگیری بر روی ده مجموعه داده به دست آمدند. نتایج مربوط به هشت روش یادگیری غیر از روش پیشنهادی از مرجع [1] آورده شده است. هر عدد در جدول، میانگین صد اجرای مستقل است. مقادیر پرنگ شده در هر سطر بیان گر این مسئله هستند که از بقیه موارد به طور معنادار بهتر هستند. تفاوت معناداری میان چند مقدار پرنگ شده در یک سطر، وجود ندارد. جداول ۶، ۷ و ۸ نشان‌دهنده برد، باخت یا مساوی بودن نتایج روش پیشنهادی با هشت روش متداول دیگر به ترتیب در معیارهای میانگین هندسی، تعادل و سطح زیر نمودار هستند. در این جداول حروف W, L و T به ترتیب بیان گر برد، باخت و مساوی بودن نتیجه هستند. از مشاهده این جداول متوجه می‌شویم که روش پیشنهادی بهترین نتیجه را در معیارهای میانگین هندسی و تعادل، در شش مجموعه داده از ده مجموعه داده به دست آورد. برای داده‌هایی که نرخ عدم توازن زیادی دارند، مثل مجموعه داده‌های cm1، cm2 و pc1، روش پیشنهادی در معیارهای میانگین هندسی و تعادل عملکرد بسیار خوبی دارد.

(جدول-۱): پارامترهای الگوریتم ژنتیک
(Table-1): Parameters of Genetic algorithm

مقدار	پارامتر الگوریتم ژنتیک
۱۰۰	اندازه جمعیت
۱۰۰	تعداد نسل (تعداد تکرار)
۱	طول ژن
۰/۷	احتمال همبزی
۰/۱	احتمال جهش

۴- نتایج و بحث

در این بخش، کارایی الگوریتم پیشنهادی خود را با هشت الگوریتم متداول در مسئله تشخیص خطا مقایسه می‌کنیم و سپس به تحلیل نتایج می‌پردازیم. هشت الگوریتم پرکاربرد یادگیری طبقه‌های نامتوارن عبارتند از: الگوریتم کاهش طبقه اکثریت به طور تصادفی^۱، الگوریتم کاهش کلاس اکثریت به طور تصادفی و متعادل^۲، الگوریتم مبتنی بر حرکت حد آستانه^۳، الگوریتم اسموت ارتقا یافته^۴، الگوریتم آدابوست ان سی^۵، الگوریتم بیز ساده با لگ فیلتر^۶، الگوریتم جنگل تصادفی^۷، الگوریتم آدابوست ان سی پویا^۸. مشخصات مجموعه داده‌ای که در ساخت مدل‌های پیشنهادی استفاده شده‌اند، در جدول (۲) فهرست شده است [1]. این مجموعه داده‌ها در انبار داده پرمیس در دسترس هستند و به ترتیب نرخ عدم توازن در جدول قرار گرفته‌اند. به عنوان مثال، مجموعه داده mc2 دارای ۱۶۱ نمونه و مجموعه داده jml دارای ۱۰۸۸۵ نمونه است. همچنین درصد نمونه‌های خطدار از ۳۲/۲۹ درصد تا ۶/۹۴ درصد متغیر است.

در این مقاله برای اطمینان از وجود تفاوتی معنی‌دار میان روش‌ها از آزمون‌های آماری استفاده شده است [21]. منظور از تفاوت معنی‌دار میان روش‌ها این است که اختلاف میان روش‌ها به حد کافی بزرگ باشد تا بتوان اطمینان حاصل کرد که نتایج به طور تصادفی یا براثر شناسی، بهتر نشده‌اند. انتخاب آزمون آماری مناسب، بستگی به تعداد متغیرهای پژوهش، تعداد گروههای مقایسه شده، مستقل یا وابسته بودن گروه‌ها، نرمال بودن یا نبودن توزیع داده‌ها و نوع داده‌ها (عددی، رتبه‌ای، اسمی) دارد. اگر نوع داده‌ها کمی و توزیع آن‌ها نرمال باشد از مدل‌های آماری پارامتریک استفاده

^۱ Random undersampling(RUS)

^۲ Balanced random undersampling (RUS-bal)

^۳ Threshold-moving(THM)

^۴ SMOTEBoost (SMB)

^۵ AdaBoost.NC (BNC)

^۶ Naive Bayes with the log filter (NB)

^۷ Random Forest (RF)

^۸ Dynamic AdaBoost.NC(DNC)

^۹ T-Test



چهار مورد و روش آدابوست ان سی در دو مورد عملکردهای بهتری نسبت به روش پیشنهادشده داشتند. بقیه روش‌ها در هر ده مجموعه‌داده به طور معنادار، نسبت به روش پیشنهادی عملکرد ضعیفتری داشتند. بنابراین روش پیشنهادشده در معیار تعادل تعداد ۶۸ برد، ۴ مساوی و هشت باخت از مجموع هشتاد مقایسه انجام شده به دست آورد که این نتایج نشان‌دهنده پیش‌بینی درست خطاها با دقت به نسبه خوب، توسط روش پیشنهادی است. به طور کلی از مشاهده جداول در می‌یابیم که برای داده‌هایی با نرخ عدم توازن بالا، روش پیشنهادی انتخاب خوبی است. به طور کلی روش‌های آدابوست ان سی، آدابوست ان سی پویا و روش پیشنهادی کارایی بهتری نسبت به روش‌های کاهش کلاس اکثریت به طور تصادفی، کاهش طبقه اکثریت به طور تصادفی و متعادل، حرکت حد آستانه، اسموت ارتقا‌یافته، بیز ساده و جنگل تصادفی دارند.

شکل (۳) نمودار میله‌ای میانگین مقادیر معیارهای کارایی میانگین هندسی، سطح زیرنومدار و تعادل همراه با مقادیر انحراف‌معیار را بر روی ده مجموعه‌داده خطای نرم‌افزار نشان می‌دهد. این مقادیر توسط هشت روش متداول در زمینه یادگیری مسائل نامتوازن و یک روش پیشنهادشده به دست آمده است که محور افقی نمودار را تشکیل می‌دهند. محور عمودی این نمودار نشان‌گر میانگین مقادیر سه معیار کارایی بر روی ده مجموعه‌داده است. همان‌طور که در نمودار میله‌ای مشاهده می‌شود، ارتفاع مقادیر معیارهای کارایی میانگین هندسی و تعادل در روش پیشنهادی از تمام روش‌های متداول دیگر بالاتر است. این نتیجه به معنای آن است که روش پیشنهادی نسبت به روش‌های دیگر، در پیش‌بینی نمونه‌های هر دو طبقه از دقت بالاتری برخوردار است. چون معیار میانگین هندسی نشان‌دهنده تشخیص درست طبقه‌های مثبت و منفی و معیار تعادل، بیان‌گر کارایی پیش‌بینی کننده در ایجاد تعادل میان نرخ تشخیص درست و نرخ عدم تشخیص درست است. ارتفاع معیار سطح زیرنومدار در روش جنگل تصادفی از همه روش‌ها بیشتر است، اماً میانگین مقادیر معیارهای تعادل و میانگین هندسی آن از همه روش‌ها کمتر است. بنابراین، این روش دقت کافی برای پیش‌بینی طبقه‌های مثبت و منفی ندارد و روش مناسبی برای حل مسئله تشخیص خطای نرم‌افزار نیست. به طور کلی از مشاهده نمودار میله‌ای نتیجه می‌گیریم که روش پیشنهادی و روش‌های آدابوست ان سی و آدابوست ان سی پویا، کارایی بهتری نسبت به روش‌های کاهش طبقه اکثریت به طور تصادفی، کاهش طبقه اکثریت به طور تصادفی و متعادل، حرکت حد آستانه، اسموت ارتقا‌یافته، بیز ساده و جنگل تصادفی دارند و برای مسئله خطای نرم‌افزار مناسب هستند.

در مقایسه روش پیشنهادی با هشت روش متداول دیگر، در مجموع، روی ده مجموعه‌داده، تعداد ۶۷ مرتبه در معیار میانگین هندسی، به طور معنادار نتیجه‌ای بهتر از نتایج روش‌های دیگر به دست آورده است. همچنین در هفت مورد، نتیجه مساوی و تنها در شش مورد، نتیجه بدتری داشته است. به عبارت دیگر الگوریتم پیشنهادی در تمامی مجموعه‌داده‌ها، در معیار میانگین هندسی، از روش‌های حرکت حد آستانه، اسموت ارتقا‌یافته و جنگل تصادفی عملکرد بهتری دارد. در پنج مورد به طور معنادار نتیجه‌ای مساوی با روش‌های کاهش طبقه اکثریت به طور تصادفی و کاهش طبقه اکثریت به طور تصادفی و متعادل و بیز ساده به دست آمده است. در مجموعه داده‌های pc3 و pc4 از روش‌های آدابوست ان سی و آدابوست ان سی پویا مقدار میانگین هندسی کمتری به دست آورد. با توجه به نتایج به دست آمده در می‌یابیم که روش پیشنهادشده در پیش‌بینی درست نمونه‌های طبقه خطدار و بدون خط، قابلیت زیادی دارد و دارای دقت بالا در هر دو طبقه است. چون مقدار معیار میانگین هندسی، نرخ تشخیص درست را بازتاب می‌کند.

معیار سطح زیر نمودار میزان تعادل در نرخ تشخیص درست و نرخ تشخیص اشتباه را نشان می‌دهد. از مجموع هشتاد مقایسه انجام شده بر روی ده مجموعه داده در معیار سطح زیر نمودار راک، روش پیشنهادی در ۳۴ مرتبه نسبت به دیگر روش‌ها، به طور معناداری به نتایج بهتری دست پیدا کرده است. همچنین با سیزده نتیجه روش‌های متداول دیگر، بنابر آزمون تی، از لحاظ مفهومی برابر است. بنابراین می‌توان گفت در مجموع از هشتاد مقایسه انجام شده، در ۴۷ مورد نتیجه‌ای بهتر یا مساوی با دیگر روش‌های متداول تشخیص خطای به دست آورده است. تنها در ۳۳ مورد، عملکرد خوبی نداشته است. در معیار سطح زیر نمودار راک، روش پیشنهادی از روش‌های کاهش طبقه اکثریت به طور تصادفی، کاهش طبقه اکثریت به طور تصادفی و متعادل و حرکت حد آستانه، در نه مجموعه داده بهتر عمل کرده و نسبت به بقیه روش‌ها عملکرد خوبی نداشته است.

معیار تعادل، فاصله اقلیدسی بین ایده‌آل‌ترین نقطه را (آنجا که نرخ تشخیص خطای یک و نرخ عدم تشخیص خطای صفر است) در نمودار راک و موقعیت واقعی نشان می‌دهد. مقدار این معیار بین صفر و یک متغیر است. اگر مقدار آن برابر با یک باشد، به معنای این است که تمام خطای بدون اشتباه تشخیص داده شده‌اند. از مشاهده جداول ۳، ۴، ۵ در می‌یابیم که در شش مجموعه‌داده از ده مجموعه‌داده، روش پیشنهادی نسبت به هشت روش متداول دیگر، به نتایج بهتری در معیار تعادل دست پیدا کرده است. روش آدابوست ان سی پویا در



(جدول-۳): مقایسه میانگین و انحراف معیار روش پیشنهادی با سایر روش‌ها برای معیار میانگین هندسی
 (Table-3): comparing the mean and standard deviation of proposed method with
 the others for G-mean measure

G-mean	RUS	RUS-bal	THM	SMB	BNC	NB	RF	DNC	Proposed method
mc2	0.571±0.150	0.578±0.140	0.566±0.173	0.627±0.169	0.597±0.135	0.6070.137	0.5350.191	0.571±0.152	0.682±0.120
kc2	0.571±0.150	0.578±0.140	0.675±0.120	0.642±0.108	0.679±0.016	0.734±0.070	0.647±0.105	0.777±0.701	0.75±0.081
jm1	0.571±0.150	0.578±0.140	0.647±0.019	0.541±0.025	0.723±0.046	0.734±0.040	0.494±0.025	0.672±0.017	0.692±0.073
kc1	0.571±0.150	0.578±0.140	0.643±0.064	0.582±0.072	0.694±0.031	0.780±0.036	0.562±0.072	0.734±0.040	0.7190.043
pc4	0.571±0.150	0.578±0.140	0.643±0.064	0.541±0.025	0.694±0.031	0.780±0.036	0.562±0.072	0.734±0.040	0.8050.037
pc3	0.571±0.150	0.578±0.140	0.643±0.064	0.642±0.108	0.645±0.018	0.686±0.050	0.649±0.022	0.777±0.701	0.688±0.072
cm1	0.571±0.150	0.578±0.140	0.643±0.064	0.642±0.108	0.645±0.018	0.686±0.050	0.649±0.022	0.672±0.071	0.698±0.095
kc3	0.571±0.150	0.578±0.140	0.643±0.064	0.642±0.108	0.645±0.018	0.686±0.050	0.649±0.022	0.670±0.071	0.699±0.095
mw1	0.571±0.150	0.578±0.140	0.643±0.064	0.642±0.108	0.645±0.018	0.686±0.050	0.649±0.022	0.670±0.081	0.670±0.097
pc1	0.571±0.150	0.578±0.140	0.643±0.064	0.642±0.108	0.645±0.018	0.686±0.050	0.649±0.022	0.670±0.071	0.673±0.155
									0.743±0.154
									0.729±0.154
									0.730±0.102

(جدول-۴): مقایسه میانگین و انحراف معیار روش پیشنهادی با سایر روش‌ها برای معیار تعادل.
(Table-4): comparing the mean and standard deviation of proposed method with the others for Balance

															Proposed method
balance	RUS	RUS-bal	THM	SMB	BNC	NB	RF	DNC							
Mc2	Kc2	Jm1	Kc1	Pc4	Pc3	Cm1	Kc3	Mw1	Pc1	pc1					
0.665±0.113		0.720±0.097	0.688±0.513	0.716±0.042	0.789±0.032	0.663±0.083	0.719±0.091	0.733±0.123	0.717±0.136	0.722±0.099	0.730±0.102				
0.569±0.134		0.777±0.071	0.672±0.027	0.733±0.042	0.850±0.043	0.739±0.064	0.653±0.117	0.655±0.143	0.623±0.177	0.682±0.133	0.698±0.145				
0.538±0.130		0.617±0.101	0.474±0.019	0.529±0.060	0.593±0.083	0.432±0.064	0.345±0.225	0.389±0.110	0.444±0.157	0.496±0.108	0.505±0.167				
0.595±0.128		0.719±0.062	0.584±0.020	0.663±0.026	0.753±0.031	0.651±0.031	0.663±0.284	0.693±0.120	0.636±0.137	0.553±0.037	0.576±0.050				
0.588±0.128		0.753±0.084	0.678±0.017	0.718±0.050	0.854±0.049	0.749±0.078	0.607±0.118	0.655±0.144	0.567±0.193	0.665±0.126	0.691±0.130				
0.620±0.146		0.68±0.0101	0.518±0.021	0.551±0.063	0.682±0.080	0.478±0.071	0.407±0.122	0.472±0.132	0.470±0.159	0.541±0.128	0.553±0.191				
0570±0.135		0.660±0.117	0.643±0.0.0 20	0.627±0.067	0.721±0.104	0.571±0.100	0.502±0.145	0.483±0.151	0.450±0.165	0.596±0.149	0.601±0.195				
0.574±0.122		0.709±0.078	0.642±0.001 9	0.677±0.052	0.808±0.050	0.683±0.052	0.577±0.112	0.669±0.114	0.619±0.126	0.688±0.078	0.711±0.089				
0.571±0.125		0.705±0.086	0.646±0.023	0.659±0.073	0.784±0.086	0.659±0.093	0.526±0.146	0.581±0.164	0.566±0.161	0.636±0.085	0.645±0.157				

(جدول-۵): مقایسه میانگین و انحراف معیار روش پیشنهادی با سایر روش‌ها

برای معیار سطح زیرنمودار.

(Table-5): comparing the mean and standard deviation of proposed method with the others for AUC

(جدول-۶): نتایج مقایسه بر اساس t-test به صورت برد، مساوی یا باخت برای معیار میانگین هندسی.

(Table-6): comparing results of T-test as Win, Tie or Lost for G-mean

G-mean	RUS	RUS-bal	THM	SMB	BNC	NB	RF	DNC
mc2	W	W	W	W	W	W	W	W
kc2	W	W	W	W	T	W	W	L
jm1	W	W	W	W	W	W	W	W
kc1	W	W	W	W	T	W	W	L
pc4	T	T	W	W	L	W	W	L
pc3	W	T	W	W	L	T	W	L
cm1	W	W	W	W	W	W	W	W
kc3	W	W	W	W	W	W	W	W
mw1	W	W	W	W	W	W	W	W
pc1	W	T	W	W	W	W	W	W
Total Wins = 67			Total losses = 6			Total Ties = 7		

(جدول-۷): نتایج مقایسه بر اساس آزمون تی به صورت برد، مساوی یا باخت برای معیار تعادل.

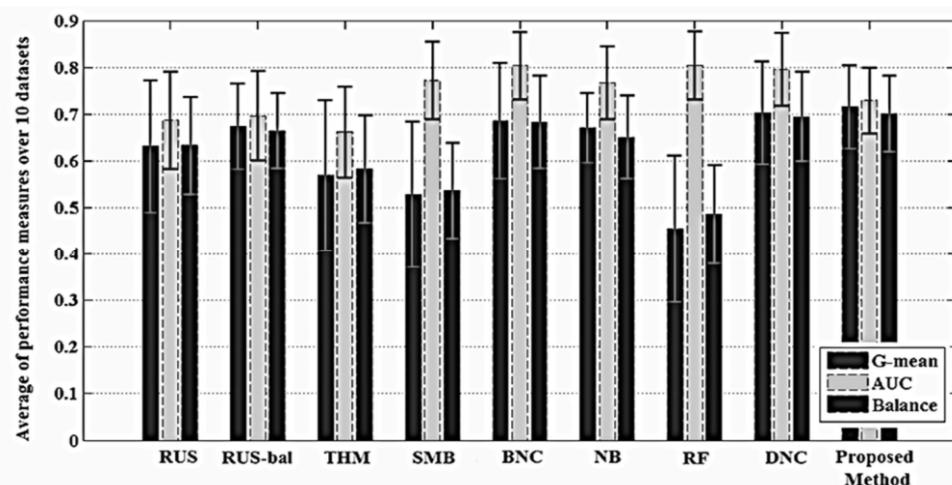
(Table-7): comparing results of T-test as Win, Tie or Lost for Balance

Balance	RUS	RUS-bal	THM	SMB	BNC	NB	RF	DNC
mc2	W	W	W	W	W	W	W	W
kc2	W	W	W	W	L	T	W	L
jm1	W	W	W	W	W	W	W	W
kc1	W	W	W	W	T	W	W	L
pc4	T	L	W	W	L	W	W	L
pc3	W	T	W	W	L	W	W	L
cm1	W	W	W	W	W	W	W	W
kc3	W	W	W	W	W	W	W	W
mw1	W	W	W	W	W	W	W	W
pc1	W	W	W	W	W	W	W	W
Total Wins = 68			Total losses = 8			Total Ties = 4		

(جدول-۸): نتایج مقایسه بر اساس آزمون تی به صورت برد، مساوی یا باخت برای معیار سطح زیرنمودار.

(Table-8): comparing results of T-test as Win, Tie or Lost for AUC

AUC	RUS	RUS-bal	THM	SMB	BNC	NB	RF	DNC
mc2	W	W	W	L	W	T	T	W
kc2	W	W	W	W	T	L	L	L
jm1	W	W	W	W	L	W	L	L
kc1	W	W	W	L	L	L	L	L
pc4	T	L	W	L	L	L	L	L
pc3	W	W	W	L	L	L	L	L
cm1	W	W	W	W	L	T	T	L
kc3	W	W	W	T	L	L	L	L
mw1	W	W	W	T	T	T	T	W
pc1	W	T	T	L	L	W	L	L
Total Wins = 34			Total losses = 33			Total Ties = 13		



(شکل-۳): مقایسه کارایی روش پیشنهادی با بقیه روشها با نمودار میله‌ای: میانگین مقادیر معیارهای کارایی میانگین هندسی، سطح زیرنمودار و تعادل، همراه با مقادیر انحراف معیار آنها بر روی داده خطای نرم افزار.

(Figure-3): performance comparison of our proposed method with the others by bar chart: the mean of G-mean, AUC and balance values as well as their standard deviation for ten datasets.

استفاده از منطق فازی و الگوریتم‌های تکاملی اشاره کرد. روش پیشنهادی با وجود آن که در معیارهای میانگین هندسی و تعادل نتایج بسیار خوبی به دست آورده است، اما در معیار سطح زیرنومدار نتایج چندان رضایت‌بخش نیست. علت این است که روش پیشنهادی روی داده‌های با نرخ عدم توازن بالا، خوب عمل می‌کند و روی داده‌های با نرخ عدم توازن پایین، به خوبی روش آدابوست ان سی پویا نیست.

6- References

۶-مراجع

- [1] S. Wang and X. Yao, "Using class imbalance learning for software defect prediction," *IEEE Trans. Reliab.*, vol. 62, no. 2, pp. 434–443, 2013.
- [2] B.-J. Park, S.-K. Oh, and W. Pedrycz, "The design of polynomial function-based neural network predictors for detection of software defects," *Inf. Sci. (Ny.)*, vol. 229, pp. 40–57, 2013.
- [3] P. C. Pendharkar, "Exhaustive and heuristic search approaches for learning a software defect prediction model," *Eng. Appl. Artif. Intell.*, vol. 23, no. 1, pp. 34–40, 2010.
- [4] J. Zheng, "Cost-sensitive boosting neural networks for software defect prediction," *Expert Syst. Appl.*, vol. 37, no. 6, pp. 4537–4543, 2010.
- [5] مهدی زاده محبوبه، افتخاری مهدی. ارائه روش جدید مبتنی بر برنامه‌نویسی ژنتیک برای وزن‌دهی قوانین فازی در طبقه‌بندی نامتوازن. پردازش علائم و داده‌ها. ۱۳۹۳؛ ۱۱ (۲): ۱۱۱-۱۲۵.
- [5] Mahdizadeh Mahboubeh, Eftekhari Mahdi. "A new fuzzy rules weighting approach based on Genetic Programming for imbalanced classification". *JSDP.*, no. 11 (2), pp.111-125, 2015
- [6] Z. Yan, X. Chen, and P. Guo, "Software defect prediction using fuzzy support vector regression," *Adv. Neural Networks-ISNN 2010*, pp. 17–24, 2010.
- [7] A. K. Pandey and N. K. Goyal, "A fuzzy model for early software fault prediction using process maturity and software metrics," *Int. J. Electron. Eng.*, vol. 1, no. 2, pp. 239–245, 2009.
- [8] S. Di Martino, F. Ferrucci, C. Gravino, and F. Sarro, "A genetic algorithm to configure support vector machines for predicting fault-prone components," in *International Conference on Product Focused Software Process Improvement*, 2011, pp. 247–261.
- [9] P. S. Sandhu, S. Khullar, S. Singh, S. K. Bains, M. Kaur, and G. Singh, "A Study on Early Prediction of Fault Proneness in Software Modules using Genetic Algorithm," *World Acad. Sci. Eng. Technol.*, vol. 72, 2010.

۵-نتیجه‌گیری

تشخیص مازول‌های مستعد خطا در ابتدای پروژه‌های نرم‌افزاری، باعث صرفه‌جویی در تلاش و هزینه‌های گروه توسعه می‌شود. این مازول‌ها در صدد کمی از سامانه نرم‌افزاری را شامل می‌شوند؛ بنابراین مسئله تشخیص خطا جزو مسائل نامتوازن است و تشخیص مازول‌های مستعد خطا نسبت به مازول‌های بدون خطا اهمیت بیشتری دارد.

در این مقاله با استفاده از خوشبندی کاهشی و خوشبندی فازی ۵ میانگین فازی یک مدل پیش‌بینی خطای نرم‌افزار تولید و بدین منظور در ابتدا از تمام ویژگی‌های نمونه‌ها در مجموعه داده‌ها استفاده کردیم؛ اما با توجه به این که ممکن است در هر مجموعه داده بعضی از ویژگی‌ها از بقیه مهم‌تر باشند، برای بهبود نتایج به دست آمده از الگوریتم ژنتیک برای انتخاب ویژگی استفاده کردیم. برای کنترل و حل نامتوازن بودن طبقه‌ها، نمونه‌های طبقه‌اقلیت را وزن دار کردیم تا طبقه‌بند به طبقه‌اقلیت توجه بیشتری کند.

نتایج به دست آمده از روش پیشنهادی با هشت الگوریتم متداول مقایسه و ارزیابی شده‌اند. این الگوریتم‌ها شامل سه نوع روش یادگیری جمعی، نمونه‌گیری مجدد و حرکت حد آستانه می‌شوند. این ارزیابی بر روی ده مجموعه داده خطای نرم‌افزار که گستره وسیعی از اندازه و نرخ عدم توازن را در بردازند، انجام شد. برای نشان دادن این که نتایج به دست آمده در عمل مفید هستند، از سه معیار ارزیابی کارایی استفاده کردیم. این معیارها شامل میانگین هندسی، تعادل و معیار سطح زیرنومدار هستند. نتایج مقایسه‌ها نشان دادند که الگوریتم پیشنهادی بهترین نتایج را در معیارهای میانگین هندسی و تعادل به دست آورد. الگوریتم جنگل تصادفی مقدار معیار سطح زیر نومدار بالا و مقدار معیارهای تعادل و میانگین هندسی بسیار کمی به دست آورد. بنابراین این الگوریتم برای مسئله تشخیص خطای نرم‌افزار مناسب نیست. بالابودن مقدار سطح زیرنومدار در الگوریتم آدابوست ان سی پویا نسبت به روش پیشنهادی و کمتر بودن مقدار میانگین هندسی و تعادل نشان دهنده این است که مقدار نرخ تشخیص درست هر دو طبقه، نسبت به روش پیشنهادی اختلاف بیشتری دارند؛ در حالی که در روش پیشنهادی ما، این مقدار هر دو بالا و به هم نزدیک هستند. علاوه بر معیارهای دقت، از دیگر مزایای الگوریتم پیشنهادی می‌توان به مواردی چون واپس‌نمودن زیاد به پارامترها، زمان محاسباتی به نسبه کم، پیچیده‌نمودن طبقه‌بندی‌های به دست آمده و ساخت طبقه‌بند قدرتمند با



مهدی افتخاری مدرک کارشناسی خود را در سال ۱۳۷۹ در رشته مهندسی کامپیوتر گرایش سختافزار از دانشگاه شیراز و مدارک کارشناسی ارشد و دکتراخود را بهترتیپ در سال‌های ۱۳۸۲ و ۱۳۸۶ در رشته مهندسی کامپیوتر گرایش هوش مصنوعی از همان دانشگاه اخذ کرد. وی از سال ۱۳۸۶ تا کنون عضو هیئت علمی بخش مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه شهید باهنر کرمان است و در سال ۱۳۹۳ به مرتبه دانشیاری ارتقا پیدا کرده است. حوزه‌های تخصصی ایشان مجموعه‌ها و مدل‌های فازی، الگوریتم‌های تکاملی و یادگیری ماشین است. وی تاکنون بیش از ۱۲۰ مقاله علمی در نشریات و کنفرانس‌های معتبر داخلی و خارجی به چاپ رسانیده است.

نشانی رایانماء ایشان عبارت است از:

m.eftekhari@uk.ac.ir



مریم مجیدی مؤمن آبادی مدرک کارشناسی خود را در سال ۱۳۸۹ در رشته مهندسی کامپیوتر گرایش نرم‌افزار از دانشگاه شهید باهنر کرمان و مدرک کارشناسی ارشد خود را در سال ۱۳۹۳ در رشته مهندسی فناوری اطلاعات گرایش طراحی و تولید نرم‌افزار دانشگاه آزاد اسلامی واحد کرمان اخذ کرد. حوزه‌های تخصصی ایشان مجموعه‌ها و مدل‌های فازی، طراحی نرم‌افزار و توسعه نرم‌افزار برای تلفن‌های همراه است.

نشانی رایانماء ایشان عبارت است از:

Maryam.majidi93@gmail.com



مجتبی خمر مدرک کارشناسی خود را در سال ۱۳۹۳ در رشته مهندسی کامپیوتر گرایش سختافزار از دانشگاه سیستان و بلوچستان و مدرک کارشناسی ارشد خود را در سال ۱۳۹۶ در رشته مهندسی کامپیوتر گرایش هوش مصنوعی از دانشگاه شهید باهنر کرمان اخذ کرد. حوزه‌های تخصصی ایشان مجموعه‌ها و مدل‌های فازی، و یادگیری ماشین است.

نشانی رایانماء ایشان عبارت است از:

Mojtabakhammar69@gmail.com

- [10] M. M. Rosli, N. H. I. Teo, N. S. M. Yusop, and N. S. Mohammad, "The design of a software fault prone application using evolutionary algorithm," in *Open Systems (ICOS), 2011 IEEE Conference on*, 2011, pp. 338–343.
- [11] M.-Y. Chen, "A hybrid ANFIS model for business failure prediction utilizing particle swarm optimization and subtractive clustering," *Inf. Sci. (Ny.)*, vol. 220, pp. 180–195, 2013.
- [12] M. E. R. Bezerra, A. L. I. Oliveira, and S. R. L. Meira, "A constructive rbf neural network for estimating the probability of defects in software modules," in *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, 2007, pp. 2869–2874.
- [13] M. E. R. Bezerra, A. L. I. Oliveira, and P. J. L. Adeodatoz, "Predicting software defects: A cost-sensitive approach," in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, 2011, pp. 2515–2522.
- [14] H. A. Al-Jamimi and L. Ghouti, "Efficient prediction of software fault proneness modules using support vector machines and probabilistic neural networks," in *Software Engineering (MySEC), 2011 5th Malaysian Conference in*, 2011, pp. 251–256.
- [15] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A possibilistic fuzzy c-means clustering algorithm," *IEEE Trans. fuzzy Syst.*, vol. 13, no. 4, pp. 517–530, 2005.
- [16] D. Gray, D. Bowes, N. Davey, Y. Sun, and B. Christianson, "Using the Support Vector Machine as a Classification Method for Software Defect Prediction with Static Code Metrics," in *EANN, 2009*, vol. 2009, pp. 223–234.
- [17] K. O. Elish and M. O. Elish, "Predicting defect-prone software modules using support vector machines," *J. Syst. Softw.*, vol. 81, no. 5, pp. 649–660, 2008.
- [18] S. Wang, A. Mathew, Y. Chen, L. Xi, L. Ma, and J. Lee, "Empirical analysis of support vector machine ensemble classifiers," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6466–6476, 2009.
- [19] I. Gondra, "Applying machine learning to software fault-proneness prediction," *J. Syst. Softw.*, vol. 81, no. 2, pp. 186–195, 2008.
- [20] S. R. Kannan, S. Ramathilagam, and P. C. Chung, "Effective fuzzy c-means clustering algorithms for data clustering problems," *Expert Syst. Appl.*, vol. 39, no. 7, pp. 6292–6300, 2012.
- [21] O. T. Yıldız, O. Aslan, and E. Alpaydın, "Multivariate statistical tests for comparing classification algorithms," *Lect Notes Comp Sci*, vol. 6683, pp. 1–15, 2011.