Journal of Soft Computing and Information Technology (JSCIT)
Babol Noshirvani University of Technology, Babol, Iran
Journal Homepage: jscit.nit.ac.ir
DOI: 10.22034/jscit.2025.529609.2131
Volume 14, Number 4, Winter 2026, pp. 23-40
Received: 06/10/2025, Revised: 11/11/2025, Accepted: 11/15/2025

Inconsistency Recognition of Imbalanced Data Over Time for Just-In-Time Software Defect Prediction

Zeynab Chitsazian[1], Saeed Sedighian Kashi[2*] and Amin Nikanjam[3]
1- Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran.
2*- Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran.
3- Huawei Distributed Scheduling and Data Engine Lab, Canada.
[1]z.chitsazian@email.kntu.ac.ir, [2*] sedighian@kntu.ac.ir, and [3]amin.nikanjam@h-partners.com

Corresponding author's address: Saeed Sedighian Kashi, Faculty of Computer Engineering, K.N. Toosi University of Technology, Tehran, Iran.

Abstract— In the context of Just-In-Time Software Defect Prediction (JIT-SDP), studies have been conducted that examine the inconsistency of interpretation outputs across different datasets and models. Interpretation techniques identify the importance and contribution of features to the target class prediction outcome without requiring labeled test data. These studies have demonstrated that the interpretation output of an offline-trained model is inconsistent across different datasets. Such inconsistency may also occur within a time-dependent dataset, leading to model instability and a consequent degradation in its performance over time. This research introduces a novel method, termed Interpretation Inconsistency Detection for JIT-SDP (IID-JSDP), which predicts model instabilities by identifying significant temporal shifts in the interpretation of instances over time. For evaluation, benchmark methods based on monitoring error rate are employed, as these methods identify model instability through performance degradation. However, these benchmark methods require labeled test data to detect instability, which itself introduces a detection delay. Furthermore, the issue of data imbalance (i.e., the number of minority class defect-inducing instances is significantly lower than that of the majority class clean instances) in JIT-SDP poses a challenge, as it reduces the reliability of the overall accuracy metric and the performance of threshold-dependent metrics. For this reason, the proposed method is also tested on resampled data and compared against benchmark methods based on both threshold-dependent and threshold-independent metrics. Results from a study on 20 datasets indicate that IID-JSDP can predict model instability over time with high accuracy (achieving an average accuracy of 98%).

Keywords- Just-in-time SDP, JIT-SDP, Concept drift detection, Incremental learning, Explainable artificial intelligence, XAI.

مجله علمی رایانش نرم و فناوری اطلاعات
دانشگاه صنعتی نوشیروانی بابل
صفحه مجله: jscit.nit.ac.ir
شناسه دیجیتال: 10.22034/jscit.2025.529609.2131
جلد ۱۴، شماره ۴، زمستان ۱۴۰۴، صفحه ۲۳-۴۰
دریافت: ۱۴۰۴/۰۳/۲۰، بازنگری: ۱۴۰۴/۰۸/۲۰، پذیرش: ۱۴۰۴/۰۸/۲۴

تشخیص ناسازگاری داده‌های نامتوازن در طول زمان برای پیش‌بینی نقص نرم‌افزار به موقع

زینب چیت‌سازیان[۱]، سعید صدیقیان کاشی*[۲]، امین نیک انجام[۳]
۱- دانشکده مهندسی کامپیوتر، دانشگاه خواجه نصیرالدین طوسی، تهران، ایران.
*۲- دانشکده مهندسی کامپیوتر، دانشگاه خواجه نصیرالدین طوسی، تهران، ایران.
۳- آزمایشگاه زمان‌بندی توزیع‌شده و موتور داده هوآوی، کانادا.
[1]z.chitsazian@email.kntu.ac.ir, [2*]sedighian@kntu.ac.ir, and [3]amin.nikanjam@h-partners.com

* نشانی نویسنده مسئول: سعید صدیقیان کاشی، تهران، خیابان شریعتی، دانشگاه صنعتی خواجه نصیرالدین طوسی، دانشکده مهندسی برق و کامپیوتر.

چکیده- در زمینه پیش‌بینی نقص نرم‌افزار به‌هنگام (JIT-SDP)، مطالعاتی انجام شده‌اند که ناسازگاری خروجی تفسیر روی مجموعه‌داده‌ها و مدل‌های مختلف را بررسی کرده‌اند. تکنیک‌های تفسیر، اهمیت و تاثیر ویژگی‌ها را در نتیجه پیش‌بینی کلاس هدف بدون نیاز به داده‌های آزمون برچسب‌دار مشخص می‌کنند. آنها نشان داده‌اند که خروجی تفسیر یک مدل آموزش‌دیده برون‌خط روی مجموعه‌داده‌های مختلف با یکدیگر ناسازگار است. چنین ناسازگاری‌ای ممکن است در یک مجموعه‌داده وابسته به زمان نیز رخ دهد و منجر به ناپایداری مدل و در نتیجه افت عملکرد آن در طول زمان شود. در این پژوهش، یک روش جدید با عنوان تشخیص ناسازگاری تفسیر در JIT-SDP (IID-JSDP) معرفی می‌شود که با شناسایی تغییرات معنادار در تفسیر نمونه‌ها در طول زمان، ناپایداری‌های مدل را پیش‌بینی می‌کند. برای ارزیابی، از روش‌های مرجع مبتنی بر پایش نرخ خطا استفاده شده است؛ چرا که این روش‌ها ناپایداری مدل را از طریق افت عملکرد شناسایی می‌کنند. با این حال، روش‌های مرجع برای تشخیص ناپایداری به داده‌های آزمون برچسب‌دار نیاز دارند که خود موجب تأخیر در تشخیص می‌شود. همچنین، مسئله عدم توازن داده‌ها (یعنی تعداد نمونه‌های کلاس اقلیت به مراتب کمتر از نمونه‌های سالم کلاس اکثریت است) در JIT-SDP به عنوان یک چالش مطرح است؛ زیرا موجب کاهش قابلیت اطمینان معیار دقت کلی و عملکرد معیارهای وابسته به آستانه می‌شود. به همین دلیل، روش پیشنهادی روی داده‌های بازنمونه‌گیری‌شده نیز آزمایش و با روش‌های مرجع مبتنی بر معیارهای وابسته و مستقل از آستانه مقایسه می‌شود. نتایج حاصل از مطالعه روی ۲۰ مجموعه‌داده نشان می‌دهد که IID-JSDP قادر است ناپایداری مدل را در طول زمان با دقت بالا (به طور میانگین به دقت ۹۸٪) پیش‌بینی کند.

واژه‌های کلیدی: Just-in-time SDP، JIT-SDP، تشخیص رانش مفهوم، یادگیری افزایشی، هوش مصنوعی توضیح‌پذیر، XAI.

۱- مقدمه

پیش‌بینی نقص نرم‌افزاری به‌هنگام (JIT-SDP[۱]) تکنیکی است که در فرآیند توسعه نرم‌افزار، با تحلیل تغییرات اعمال شده در کد (از جمله ساختارهای پیچیده کدنویسی و روش‌های برنامه‌نویسی نامناسب) الگوهای مستعد خطا را شناسایی می‌کند. هدف اصلی، تشخیص به‌موقع نقص‌های احتمالی پیش از آن‌که به مشکلات پیچیده و پرهزینه تبدیل شوند، است. این رویکرد امکان رفع سریع‌تر و مقرون‌به‌صرفه‌تر خطاها را فراهم می‌کند [۱, ۲]. عملکرد

مدل‌های JIT-SDP ممکن است به مرور زمان کاهش یابد، پدیده‌ای که به عنوان رانش مفهوم (CD[۲]) شناخته می‌شود [۳]. این پدیده نه به دلیل حرکت خطاها به سمت یک نقطه مشترک، بلکه به دلیل تغییر مداوم در عوامل مؤثر بر ایجاد خطا (انسان، فناوری، فرآیند و محیط) رخ می‌دهد. این عوامل به شکل‌های مختلف می‌توانند باشند. برای مثال، ویژگی‌های نرم‌افزار تغییر چندانی نکنند، اما رابطه بین این ویژگی‌ها و احتمال خطا تغییر کرده باشد؛ یا توزیع داده‌های ورودی جدید با داده‌های آموزشی قدیمی متفاوت باشد. به عبارت دیگر، الگوهای جدیدی در کد ظاهر می‌شوند که مدل هرگز آنها را ندیده است. بنابراین، یک مدل پیش‌بینی خطا یک محصول مادام‌العمر نیست، بلکه یک سیستم پویاست که نیاز به نگهداری و به‌روزرسانی مداوم دارد. وقوع CD (که معمولاً در محیط‌های پویا و در حال تغییر رخ می‌دهد) نیازمند بازآموزی مدل با داده‌های به‌روز و سازگار برای حفظ کارایی آن دارد. در همین راستا، مطالعات متعددی به موضوع تشخیص رانش مفهوم (CDD[۳]) در حوزه‌های SDP و JIT-SDP پرداخته‌اند تا پایداری مدل را ارزیابی کنند. پژوهش‌های پیشین در حوزه CDD عمدتاً بر دو محور پایش عملکرد مدل و تحلیل توزیع داده‌ها متمرکز بوده‌اند [۴-۷]. روش های مبتنی بر پایش عملکرد، تغییرات در دقت یا نرخ خطای (ER[۴]) مدل را در بازه‌های زمانی متوالی ردیابی می‌کنند تا ناپایداری‌های احتمالی را شناسایی نمایند. ER به عنوان نسبت پیش‌بینی‌های نادرست به کل پیش‌بینی‌ها تعریف می‌شود. نقطه ضعف اصلی این روش‌ها وابستگی شدید به داده‌های آزمون برچسب‌دار است که خود موجب تأخیر در تشخیص تغییر مفهوم می‌شود، چرا که دستیابی به برچسب‌های واقعی غالباً با تأخیر همراه است. علاوه بر این، چندین مطالعه به بررسی ناسازگاری خروجی الگوریتم تفسیر روی مجموعه‌داده‌ها و مدل‌های برون‌خط[۵] مختلف JIT-SDP پرداخته‌اند [۸-۱۱]. ناسازگاری بین مجموعه‌داده‌های مختلف ناشی از CD، تغییر مؤلفه منبع (SCS[۶]) نام دارد. در سناریوی SCS، داده‌ها از چندین منبع با ویژگی‌های ذاتی متمایز جمع‌آوری می‌شوند [۱۲, ۱۳]. این تنوع ذاتی در منابع داده منجر به ناسازگاری در تفسیر مدل آموزش‌دیده روی مجموعه‌داده‌های مختلف می‌شود که بازتابی از وجود مفاهیم ناسازگار در آن‌ها است. مفهوم سازگاری به پایداری رفتار یا عملکرد یک مدل اشاره دارد. نقاط CD شناسایی‌شده در جریان داده را می‌توان به عنوان شاخصی برای پیش‌بینی ناپایداری آتی عملکرد مدل به کار گرفت [۱۴]. در صورت عدم تشخیص به‌موقع CD، دقت مدل به تدریج تنزل یافته و منجر به افزایش نرخ نتایج مثبت کاذب و منفی کاذب می‌شود [۱۵]. بنابراین توسعه روش‌های کارآمد CDD برای شناسایی CD و تنظیم به‌موقع مدل، یک ضرورت حیاتی محسوب می‌شود. در این مقاله، روشی جدید با عنوان IID-JSDP[۷] پیشنهاد می‌شود که با پایش تغییرات معنادار در بردارهای تفسیر نمونه‌ها، ناپایداری مدل‌های JIT-SDP را در طول زمان پیش‌بینی می‌کند.

از چالش‌های کلیدی در این حوزه، عدم توازن شدید طبقاتی در مجموعه‌داده‌های JIT-SDP است؛ یعنی تعداد نمونه‌های ناقص (کلاس اقلیت) به مراتب کمتر از نمونه‌های سالم (کلاس اکثریت) است. این عدم توازن، هم بر معیارهای وابسته به آستانه و هم بر قابلیت اطمینان «دقت کلی» تأثیر منفی می‌گذارد. بنابراین، برای اطمینان از ارزیابی قوی، روش پیشنهادی هم روی داده‌های اصلی و هم روی داده‌های بازنمونه‌گیری‌شده[۸] ارزیابی شده است. همچنین، ارزیابی جامعی با استفاده از طیف متنوعی از معیارهای عملکرد (شامل هر دو دسته وابسته و مستقل از آستانه) انجام گرفته تا تحلیل کاملی از توانایی پیش‌بینی مدل در شرایط گوناگون ارائه شود. همچنین روش پیشنهادی با روش CDD مبتنی بر تحلیل توزیع داده‌های اصلی به عنوان روش پایه مقایسه شده است. در این روش پایه، توزیع داده‌های اصلی (داده‌های خام تفسیرنشده) در دوره‌های متوالی زمانی با استفاده از روش MANOVA[۹] [۱۶] تحلیل می‌شود. در صورت مشاهده تفاوت آماری معنادار بین توزیع داده‌های دوره جاری و دوره‌های پیشین، وقوع CD شناسایی می‌شود. به طور خلاصه، این مقاله ایده‌های زیر را ارائه می‌دهد:

- این مقاله با تمرکز بر تحلیل ناسازگاری‌های تفسیر داده‌ها در طول زمان، رویکردی نوین ارائه می‌دهد. برخلاف روش‌های قبلی که عمدتاً بر ناسازگاری‌های بین مدل‌های مختلف متمرکز بودند، رویکرد پیشنهادی ما تحول تدریجی تفسیر داده‌ها را در گذر زمان ردیابی می‌کند.
- در حالی که مطالعات قبلی وجود CD در مسئله SDP را با روش‌های نظارت بر افت عملکرد شناسایی کرده‌اند، این مقاله به کاوش CD بر اساس تغییر مفهوم و ناپایداری در تفسیر نمونه‌ها در چارچوب مسئله JIT-SDP می‌پردازد. نکته کلیدی این است که CD در ذات خود به تغییر در مفهوم بنیادین داده‌ها اشاره دارد، نه صرفاً تغییر در الگوهای آماری.
- روش پیشنهادی ما برای کشف CD برخلاف روش‌های مرجع (Benchmark) [۱۷, ۱۸] نیازمند برچسب‌زنی داده‌های جدید نیست. این ویژگی یک مزیت عمده محسوب می‌شود، زیرا فرآیند برچسب‌زنی داده‌ها در محیط‌های عملیاتی اغلب پرهزینه و زمان‌بر است. رویکرد پیشنهادی با حذف این نیاز،

امکان تشخیص به‌موقع و مقرون‌به‌صرفه‌تر CD را فراهم می‌آورد.

- برای ارزیابی دقیق‌تر، نتایج روش پیشنهادی با خروجی‌های ۷ روش مرجع از طریق آزمون‌های آماری مقایسه شده است. برخلاف پژوهش‌های قبلی که CD را صرفاً از طریق پایش ER شناسایی می‌کردند، در این مقاله از معیارهای جامع‌تری شامل چهار شاخص وابسته به آستانه (Recall، Precision، Gmean و F-measure) و سه شاخص مستقل از آستانه (AUC[۱۰]، MCC[۱۱] و Accuracy) برای تشخیص CD استفاده شده است. این رویکرد چندمعیاره امکان تحلیل همه‌جانبه‌تری از تغییرات مفهوم را فراهم می‌سازد.
- با استفاده از مطالعه آزمایشی دیگری در این پژوهش، الگوریتم تفسیر مناسب‌تر برای CDD شناسایی شده است. این بخش از تحقیق به مقایسه سیستماتیک عملکرد الگوریتم‌های مختلف تفسیر در چارچوب روش ارائه‌شده می‌پردازد.
- با توجه به مواجهه با عدم توازن داده‌ها، از تکنیک پیشرفته بازنمونه‌گیری در این پژوهش استفاده شده است. تأثیر این تکنیک بر عملکرد روش CDD پیشنهادی به دقت مورد ارزیابی قرار گرفته و مقایسه‌ای جامع بین نتایج حاصل از داده‌های اصلی و بازنمونه‌گیری‌شده انجام شده است. این مقایسه نه تنها با ارزیابی درون روشی، بلکه در مقایسه با روش‌های مرجع[۱۲] نیز صورت پذیرفته که بر اعتبار علمی یافته‌ها می‌افزاید.

ساختار این مقاله به شرح زیر است: بخش ۲ مروری بر پژوهش‌های پیشین در این حوزه ارائه می‌دهد. بخش ۳ به تشریح روش پیشنهادی می‌پردازد. بخش ۴ روش‌شناسی ارزیابی را ارائه می‌دهد. بخش ۵ نتایج تجربی به دست آمده را تحلیل می‌کند. در نهایت، بخش ۶ به جمع‌بندی یافته‌ها و ارائه پیشنهادهایی برای پژوهش‌های آینده اختصاص یافته است.

۲- کارهای پیشین

در این بخش به مرتبط‌ترین کارهایی که در گذشته در این زمینه انجام شده است می‌پردازیم.

الگوریتم تفسیر نمونه، میزان تأثیر هر ویژگی را بر پیش‌بینی برچسب کلاس مربوطه تعیین می‌کند و توضیح می‌دهد که چرا مدل یک پیش‌بینی خاص انجام داده است. برای نمونه، در مطالعه [۱۹] ویژگی‌هایی که بیشترین تأثیر را بر احتمال ناقص بودن یک نمونه کد تغییریافته (کامیت[۱۳]) داشتند، به عنوان مهم‌ترین ویژگی‌ها شناسایی شدند. الگوریتم‌های متداول تفسیر مانند IME[۱۴] [۲۰]، LIME[۱۵] [۲۱] و SHAP[۱۶] [۲۲] نتایج را برای هر نمونه به صورت مجزا تفسیر می‌کنند. این الگوریتم‌ها با اندازه‌گیری تفاوت پیش‌بینی بین نمونه اصلی و نمونه‌های مشابه عمل می‌نمایند. مطالعات Vreš و همکاران [۲۳] نشان می‌دهد که IME در مقایسه با سایر الگوریتم‌ها از پایداری بیشتری برخوردار است. همچنین Jiarpakdee اثبات کرده که الگوریتم BreakDown (BD) در مسائل SDP پایداری بهتری نسبت به LIME دارد [۱۹].

در مطالعات پیشین، الگوریتم RF به عنوان یک طبقه‌بند قابل اعتماد برای مدل‌های SDP معرفی و به طور مستمر مورد استفاده قرار گرفته است [۱, ۱۱, ۲۴, ۲۵]. شایان ذکر است که تحقیقات آتی می‌توانند الگوریتم‌های جایگزین یادگیری ماشین مانند بیز ساده (Naive Bayes) یا مدل‌های رگرسیون را مورد بررسی قرار دهند.

در پژوهش [۲۶] نشان داده‌اند، هنگامی که از پارامترهای پیش‌فرض برای الگوریتم تشخیص تغییر PH استفاده می‌کنیم، با چالشی مواجه می‌شویم که در آن، CD در تمامی مجموعه‌داده‌ها در نقاط تقریباً یکسانی رخ می‌دهد. به صورت فرضی، یک روش استاندارد CDD، نقاط تغییر را در جریان داده‌ای حاوی ۵۰۰ مشاهده تشخیص می‌دهد. فرض بر این است که پنج نقطه CD شناسایی می‌شوند. هدف ما تعیین این است که آیا این موارد، مثبت‌های کاذب هستند یا نقاط CD واقعی که توسط تشخیص‌گر شناسایی شده‌اند. اگر همه نقاط CD در فواصل یکسانی رخ دهند؛ یعنی در هر ۱۰۰ مشاهده، یک CD تشخیص داده شود، به احتمال قوی تمامی نقاط شناسایی‌شده مثبت‌های کاذب هستند. با این حال، اگر یک تشخیص‌گر، نقاط تغییر را در فواصل متفاوتی شناسایی کند، می‌توان با درجه معینی از اطمینان نتیجه گرفت که نقاط تغییر شناسایی‌شده به احتمال زیاد واقعی هستند. بنابراین با تنظیم پارامتر نرخ هشدار کاذب، نرخ رخداد مثبت‌های کاذب را کنترل شده و عملکرد روش CDD ارتقا می‌یابد.

در پژوهش [۲۷]، به صورت تخصصی به ارزیابی روش‌های بازنمونه‌گیری روی مسئله SDP پرداخت شده است. آنها روش‌های نمونه‌برداری افزایشی[۱۷]، نمونه‌برداری کاهشی[۱۸] و SMOTE[۱۹] را بررسی کرده‌اند. در نمونه‌برداری افزایشی به دلیل تکرار نمونه‌های کلاس اقلیت برای متوازن کردن داده‌ها، مشکل بیش‌برازش[۲۰] به وجود می‌آید. یعنی مدل بر روی نمونه‌های تکراری به طور افراطی آموزش می‌بیند و توانایی تعمیم خود به داده‌های جدید را از دست

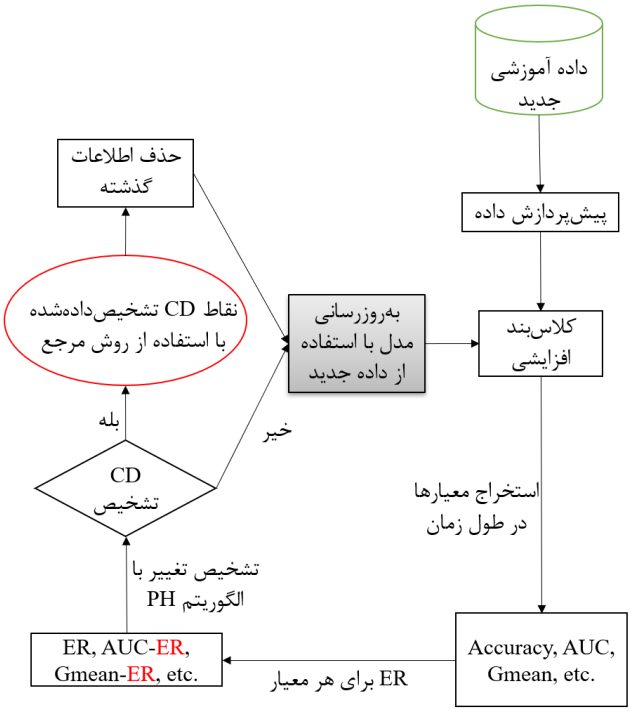
می‌دهد. در نمونه‌برداری کاهشی، حذف تصادفی نمونه‌هایی از کلاس اکثریت، منجر به از دست دادن اطلاعات ارزشمند و مفید از کلاس اکثریت می‌شود. اما تکنیک SMOTE، داده‌های مصنوعی را بر اساس فضای ویژگی‌ها (نه فضای داده) تولید می‌کند. در پژوهش [۲۷]، نشان داده‌اند که این تکنیک، عملکرد SDP را به طور قابل توجهی بهبود بخشیده و نتایج پایدارتری نسبت به حالت بدون تعادل کلاس و استفاده از روش‌های دیگر بازنمونه‌گیری ارائه می‌دهد [۲۸, ۲۹]. بنابراین، ما از تکنیک SMOTE برای بازنمونه‌گیری استفاده می‌کنیم و بررسی روش‌های دیگر بازنمونه‌گیری را به تحقیقات آینده موکول می‌کنیم.

۳- روش‌شناسی

در این بخش، طراحی روش پیشنهادی و مطالعات آزمایشی را شرح می‌دهیم. همان‌طور که پیش‌تر اشاره شد، روش IID-JSDP را برای تشخیص CD در مسئله JIT-SDP معرفی کرده‌ایم. این روش از طریق شناسایی ناسازگاری‌های تفسیر نمونه‌ها در طول زمان بر روی مجموعه‌داده‌های نقص نرم‌افزاری در سطح تغییرات یا همان کامیت‌های پروژه عمل می‌کند. مطالعات پیشین با موفقیت نقاط CD را بر اساس کاهش عملکرد مدل شناسایی کرده‌اند، اما ذاتاً CD نشان‌دهنده تغییر در مفهوم است. بنابراین می‌توان آن را با ردیابی و تحلیل تغییرات تفسیر ویژگی‌های داده نیز تشخیص داد. علاوه بر این، خروجی الگوریتم تفسیر به توسعه‌دهندگان و متخصصان کمک می‌کند تا علت ریشه‌ای خطاها را در دوره‌های زمانی مختلف درک کنند. در مقابل روش‌های مرسوم CDD که نیازمند داده‌های برچسب‌دار هستند، روش پیشنهادی ما روی داده‌های بدون برچسب عمل می‌کند. این ویژگی، کاربردپذیری آن را برای پیش‌بینی نقاط CD در شرایط واقعی افزایش می‌دهد؛ زیرا فرآیند برچسب‌زنی می‌تواند زمان‌بر بوده و باعث تأخیر در تشخیص CD شود. به همین دلیل، روش‌های مبتنی بر پایش عملکرد مدل در طول زمان (که نیازمند داده‌های برچسب‌دار هستند) را به عنوان روش‌های مرجع در نظر گرفته و نتایج روش پیشنهادی خود را با استفاده از آن‌ها ارزیابی می‌کنیم.

۳-۱- روش‌های مرجع

در این بخش، رویکردهای مرجعی که برای ارزیابی روش پیشنهادی مورد استفاده قرار گرفته‌اند، مورد بحث و تحلیل قرار می‌گیرند. فرآیند اجرای این رویکردها در شکل ۱ به صورت شماتیک به تصویر کشیده شده است. روش‌های مرجع مورد استفاده در این پژوهش، نقاط CD را از طریق پایش ER در طول زمان بر روی داده‌های آزمون برچسب‌دار شناسایی می‌کنند (شکل ۱). به طور مشخص، هرگاه مقدار ER تفاوت آماری معناداری با مقادیر دوره‌های قبلی خود نشان دهد، آن نقطه به عنوان CD در نظر گرفته می‌شود. این تفاوت معنادار بین مقادیر متوالی با استفاده از آزمون آماری PH[۲۱] تعیین می‌شود. با توجه به اینکه آزمایش‌های ما هم بر روی داده‌های اصلی و هم داده‌های بازنمونه‌گیری‌شده انجام شده، برای تحلیل دقیق‌تر، علاوه بر پایش نرخ خطای دقت، نقاط CD از طریق پایش ER سه معیار عملکرد مستقل از آستانه و چهار معیار وابسته به آستانه دیگر نیز شناسایی می‌شوند.

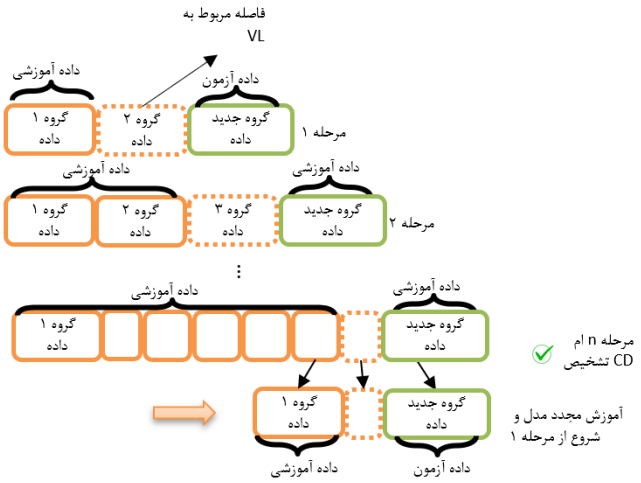


شکل ۱: فرایند اجرای روش‌های مرجع

این معیارهای عملکرد از طبقه‌بند افزایشی الگوریتم یادگیری ماشین جنگل تصادفی (RF[۲۲]) استخراج شده‌اند. از آنجا که الگوریتم PH تغییرات معنادار در مقادیر ER (و نه مستقیما در روند مقادیر معیارهای عملکرد) را در طول زمان شناسایی می‌کند، این الگوریتم را بر روی ER های استخراج‌شده از معیارهای عملکرد مدل RF اعمال می‌کنیم [۳۰]. هر تغییر معنادار آماری در ERمعیارهای عملکرد مدل نشان‌دهنده رخداد یک CD است. به صورت قراردادی، پایش نرخ خطای معیار دقت با استفاده از الگوریتم PH به اختصار ER PH نامیده می‌شود [۱۷]. در این مطالعه، نقاط CD از طریق پایش ER هفت معیار عملکرد [۲۷] زیر استخراج می‌شوند:

- سه معیار مستقل از آستانه: ER، AUC و MCC
- چهار معیار وابسته به آستانه: Recall، F-measure، Precision و Gmean

این روش‌ها به ترتیب به صورت ER PH، AUC-Er PH، MCC-Er PH، Recall-Er PH، F Measure-Er PH، Precision-Er PH و Gmean-Er PH نام‌گذاری می‌شوند. روش CDD مبتنی بر پایش ER این معیارهای عملکرد را به عنوان رویکردهای مرجع در نظر گرفته‌ایم. ما از یادگیری افزایشی برای محاسبه معیارهای عملکرد در طول زمان استفاده می‌کنیم. در این فرآیند، نمونه‌ها به تدریج به داده‌های آموزشی اضافه شده و مدل پس از افزوده شدن هر بخش از داده‌ها به‌روزرسانی می‌شود (مطابق شکل ۲). در شکل ۲، داده‌های آموزشی (مشخص شده با کادر نارنجی) می‌توانند داده‌های اصلی یا بازنمونه‌گیری‌شده باشند. داده‌های آزمون (داده‌های جدید بدون برچسب) با کادر سبز رنگ نمایش داده شده‌اند. همان‌طور که در شکل ۲ مشاهده می‌شود، پس از تشخیص CD، داده‌های تاریخی حذف شده و مدل صرفاً با آخرین دسته از داده‌های آموزشی که تا آن مرحله جمع‌آوری شده‌اند، بازآموزی می‌شود. تقسیم داده‌ها به بخش‌های صدتایی در این پژوهش مبتنی بر دو ملاحظه اصلی انجام شده است. از یک سو، نمونه‌های آموزشی در مسائل JIT-SDP برخط، اغلب با چالش تأخیر تأیید (VL[۲۳]) مواجه هستند. به عنوان مثال، یک کامیت نرم‌افزاری، زمانی به عنوان خطا برچسب می‌خورد که خطایی کشف شود، یا پس از گذشت مدت زمان مشخصی (هر کدام زودتر رخ دهد) به عنوان سالم علامت‌گذاری می‌شود (کادر نقطه‌چین در شکل ۲). اگر در این دوره انتظار هیچ خطایی شناسایی نشود، تغییر به عنوان سالم برچسب می‌خورد [۳۱]. این دوره زمانی مشخص، (عموماً ۹۰ روز) میانه زمان کشف خطاها در پروژه‌های نرم‌افزاری است. این دوره، تعادل مناسبی بین دقت برچسب‌زنی و کارایی تشخیص CD ایجاد می‌کند. به همین دلیل، این فاصله به عنوان VL بین داده‌های آموزشی و آزمون در نظر گرفته می‌شود [۲]. از سوی دیگر، تحلیل‌های انجام‌شده روی مجموعه داده‌ها نشان می‌دهد که فاصله زمانی جمع‌آوری هر ۱۰۰ نمونه کامیت به طور متوسط معادل یا بیش از ۹۰ روز بوده است، به طوری که هر بلوک ۱۰۰ تایی نماینده یک بازه زمانی معنادار (حدود ۹۰ روز) بوده و اطمینان حاصل می‌شود که برچسب‌های اکثر نمونه‌ها نهایی شده و امکان تشخیص تغییرات مفهوم با اطمینان آماری مناسب فراهم می‌شود.

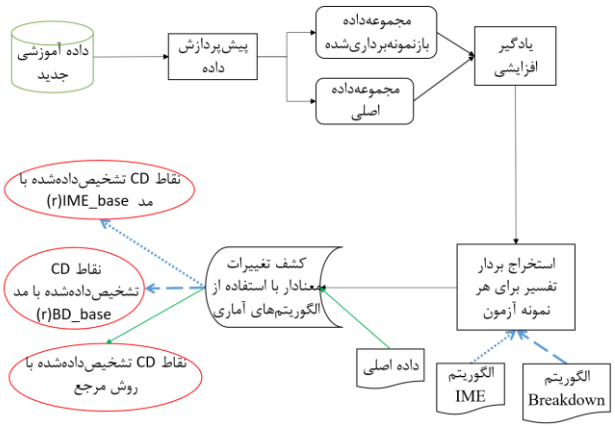


شکل ۲: فرآیند توسعه داده آموزشی در یادگیری افزایشی

۳-۲- روش پیشنهادی

هدف این بخش ارائه توصیف دقیقی از روش پیشنهادی IID-JSDP است که فرآیند اجرای آن در شکل ۳ به تصویر کشیده شده است. این روش با بهره‌گیری از تحلیل تفسیر نمونه‌ها در طول زمان، تغییرات مفهوم در داده‌ها را شناسایی می‌کند. رویکرد ارائه‌شده از سه مرحله اصلی تشکیل شده است: ابتدا داده‌های ورودی به صورت افزایشی پردازش می‌شوند، سپس بردارهای تفسیر برای هر نمونه محاسبه می‌گردد. در نهایت با تحلیل تغییرات این بردارها، نقاط CD شناسایی می‌شوند. روش پیشنهادی به‌گونه‌ای طراحی شده که نیاز به داده‌های برچسب‌دار جدید را مرتفع ساخته و امکان تشخیص بهنگام تغییرات مفهوم را فراهم می‌آورد. برای بررسی ثبات نتایج تفسیر نمونه‌ها در طول زمان، داده‌ها به بخش‌های متوالی تقسیم می‌شوند. سپس هر بخش از داده‌ها را پیش‌پردازش کرده و برای هر نمونه یک بردار تفسیر نمونه ایجاد می‌نماییم. برای اعتبارسنجی روش پیشنهادی، انتخاب الگوریتم تفسیر بهینه از اهمیت بالایی برخوردار است. بر این اساس، در این پژوهش، روش پیشنهادی را با دو الگوریتم تفسیر BD و IME برای تشخیص CD مقایسه کرده‌ایم. در گام اول، بردارهای اهمیت ویژگی‌ها برای هر نمونه توسط این الگوریتم‌های تفسیر استخراج شده‌اند. سپس از طریق شناسایی تفاوت‌های معنادار آماری بین این بردارها در بازه‌های زمانی مختلف، نقاط CD تشخیص داده شده‌اند. نتایج حاصل از دو الگوریتم تفسیر با یکدیگر و با حالت پایه (بدون استفاده از تفسیر) مقایسه شده‌اند. این سه حالت به ترتیب با نام‌های IME_base (مبتنی بر الگوریتم IME)، BD_base (مبتنی بر الگوریتم BreakDown) و روش پایه[۲۴] نامگذاری شده‌اند. همچنین به منظور بررسی تأثیر بازنمونه‌گیری بر عملکرد CDD، این حالات

با حالت‌های انجام شده روی داده‌های بازنمونه‌گیری‌شده rIME_base و rBD_base مورد مقایسه قرار گرفته‌اند (مطابق شکل ۳).



شکل ۳: فرآیند اجرایی روش پیشنهادی با توجه به الگوریتم‌های تفسیر مورد استفاده (BD: پیکان خط‌چین یا IME: پیکان نقطه‌چین)

به عنوان مثال، شبه‌کد ۱ فرایند تفسیر نمونه‌های داده آزمون را با استفاده از الگوریتم IME در زبان R نشان می‌دهد.

شبه‌کد ۱:

```
Explain <- explainVis(modelRF,
trainData, testData, method="IME",
visLevel="instance")
```

در این شبه‌کد، متغیر `Explain` یک ماتریس دوبعدی است که سطرهای آن متناظر با تعداد نمونه‌ها در هر بخش داده و ستون‌های آن مقادیر تفسیرشده ویژگی‌ها را نمایش می‌دهند. این مقادیر، میزان تأثیر هر ویژگی بر پیش‌بینی برچسب مربوط به هر نمونه را مشخص می‌کنند. `ModelRF` مدلی است که بر روی مجموعه داده `trainData` در شبه‌کد ۱ آموزش دیده است که می‌تواند به صورت اصلی یا بازنمونه‌گیری شده باشد. در آزمایش‌های انجام‌شده، هر دو حالت داده اصلی (تفسیرنشده) و داده تفسیرشده مورد بررسی قرار می‌گیرند. در مرحله بعد، از روش MANOVA برای شناسایی بخش‌هایی از داده که تفاوت آماری معناداری با بخش‌های قبلی دارند، استفاده می‌شود. این روش آماری رابطه بین چندین متغیر وابسته (ویژگی‌های تفسیرشده هر نمونه) و یک متغیر مستقل (گروه‌بندی داده‌ها بر اساس ترتیب زمانی) را تحلیل می‌کند. همانطور که پیشتر گفتیم، در این تقسیم‌بندی، ۱۰۰ نمونه اول به عنوان گروه ۱، نمونه‌های ۱۰۱ تا ۲۰۰ به عنوان گروه ۲، و به همین ترتیب گروه‌های بعدی در نظر گرفته می‌شوند. در نهایت، نقاط CD با استفاده از الگوریتم تشخیص تغییر PH بر روی نتایج حاصل از MANOVA شناسایی می‌شوند. از آنجا که در حوزه JIT-SDP مجموعه‌داده‌ای با نقاط CD برچسب‌گذاری‌شده وجود ندارد، برای ارزیابی روش پیشنهادی، خروجی آن با خروجی روش‌های مرجع شناخته‌شده مقایسه شده است. با فرض در نظر گرفتن تغییرات ER معیارهای عملکرد مختلف در طول زمان به عنوان نقاط CD اصلی (روش مرجع)، نتایج نشان می‌دهد که روش پیشنهادی عملکرد بهتری نسبت به روش پایه دارد. همچنین، با به‌کارگیری الگوریتم IME عملکرد روش پیشنهادی در مقایسه با BD بهبود می‌یابد، زیرا خروجی آن با نقاط CD شناسایی‌شده توسط روش‌های مرجع همخوانی بیشتری دارد. در نهایت، میزان تطابق روش پیشنهادی با روش‌های مرجع از نظر معیارهای دقت، حساسیت و استحکام مورد مقایسه و تحلیل قرار گرفته است.

۳-۲-۱- پیاده‌سازی روش پیشنهادی روی داده‌های بازنمونه‌گیری‌شده

همانطور که پیش‌تر اشاره شد، روشی را برای شناسایی نقاط CD از طریق بررسی ثبات تفسیر در طول زمان پیشنهاد می‌کنیم. با این حال، مشکل عدم توازن در مجموعه‌داده‌ها می‌تواند منجر به کاهش عملکرد مدل آموزشی شده و همچنین بر نتایج تفسیر تأثیر بگذارد [۳۲]. به همین دلیل، از تکنیک بازنمونه‌گیری برای مقابله با چالش توزیع نامتوازن کلاس‌ها در مجموعه‌داده‌های آموزشی نقص‌های نرم‌افزاری استفاده می‌کنیم. در یک مطالعه تجربی بر روی ۲۰ مجموعه‌داده، تأثیر بازنمونه‌گیری داده‌ها بر نتایج روش پیشنهادی CDD را با استفاده از روش‌های مرجع ارزیابی می‌کنیم. در بخش نتایج نشان خواهیم داد که استفاده از داده‌های اصلی یا بازنمونه‌گیری شده کدامیک مزیت بیشتری برای تشخیص CD دارد.

۴- ارزیابی

در این بخش، از طریق یک مطالعه موردی گسترده روی ۲۰ مجموعه‌داده معتبر از پروژه‌های متن‌باز، به ارزیابی دو سؤال تحقیقاتی کلیدی زیر می‌پردازیم:

RQ1: عملکرد روش پیشنهادی CDD با استفاده از الگوریتم‌های تفسیر مختلف در مقایسه با روش‌های مرجع چگونه است؟

RQ2: عملکرد روش پیشنهادی مبتنی بر داده‌های بازنمونه‌گیری‌شده در مقایسه با روش‌های مرجع که به معیارهای وابسته و مستقل از آستانه متکی هستند، چگونه است؟

۴-۱- تنظیم‌های آزمایش

در این بخش، چارچوب آزمایش روش پیشنهادی به طور کامل تشریح می‌شود. به طور خاص، پارامترهای آزمایش، ابزارهای مورد استفاده برای پیاده‌سازی، مجموعه‌داده‌های مورد استفاده و مراحل پیش‌پردازش داده‌ها ارائه می‌گردند.

مجموعه‌داده: در این مطالعه، روش CDD پیشنهادی را با بررسی ثبات تفسیر نمونه‌ها در طول زمان برای مسئله JIT-SDP در حال تکامل ارائه می‌دهیم و برای ارزیابی آن از ۲۰ مخزن GitHub که پیشتر در مطالعات معتبر مورد استفاده قرار گرفته‌اند [۹] بهره می‌بریم. لین و همکاران [۹] از این مجموعه‌داده برای استخراج مهم‌ترین ویژگی‌ها در مسئله JIT-SDP استفاده کرده و ثبات و پایداری عملکرد پیش‌بینی را در مجموعه‌داده‌ها و مدل‌های مختلف بررسی کردند. دو ویژگی کلیدی این مجموعه‌داده، دسترسی عمومی و توسعه بلندمدت است که آن را به منبعی ایده‌آل برای آزمایش تبدیل می‌کند. مجموعه‌داده مورد استفاده شامل معیارهای سطح کامیت است که به سه دسته تقسیم می‌شوند (جدول ۱). دسته انتشار (Diffusion) میزان پراکندگی و گستردگی تغییرات یک کامیت را در بخش‌های مختلف سیستم مشخص می‌کند. کامیت‌های با پراکندگی بالا که تغییرات در بخش‌های گسترده‌ای از سیستم ایجاد می‌کنند، به طور تجربی مستعد خطاهای بیشتری هستند [۳۳, ۳۴]. دسته اندازه (Size)، ابعاد یک کامیت را از طریق معیارهای کمّی زیر مشخص می‌کند. کامیت‌های با اندازه بزرگ‌تر معمولاً پیچیدگی بیشتری داشته و احتمال بروز خطا در آن‌ها افزایش می‌یابد. دسته هدف (Purpose)، ماهیت و هدف یک کامیت را نشان می‌دهد. پژوهش‌ها نشان داده‌اند کامیت‌هایی که به منظور رفع خطا ایجاد می‌شوند، با احتمال بالاتری ممکن است خود منجر به ایجاد خطاهای جدید در سیستم شوند [۳۵, ۳۶]. این پدیده که به خطاهای القایی[۲۵] معروف است، اهمیت توجه ویژه به کامیت‌های اصلاحی را در فرآیند پیش‌بینی نقص نشان می‌دهد.

جدول ۱: خلاصه‌ای از متریک‌های سطح کامیت [۹]

Category	Name	Description
Diffusion	NS[۲۶]	تعداد زیرسیستم‌های اصلاح شده
	ND[۲۷]	تعداد پوشه‌های اصلاح‌شده
	NF[۲۸]	تعداد فایل‌های اصلاح شده
	Entropy	آنتروپی توزیع کد اصلاح‌شده مابین فایل‌ها
Size	LA[۲۹]	خط‌های اضافه‌شده به کد
	LD[۳۰]	خط‌های حذف‌شده از کد
	LT[۳۱]	تعداد خطوط کل تغییر یافته
Purpose	FIX	آیا کامیت مذکور به منظور رفع یک خطای موجود ایجاد شده است یا خیر

پیش‌پردازش داده: مجموعه‌داده‌های مورد استفاده در این مطالعه حاوی متریک‌های سطح کامیت هستند که دارای چولگی و مقیاس‌های متفاوتی بوده و برخی از متریک‌ها همبستگی بالایی نشان می‌دهند. برای حل این مشکلات، از روشی مشابه کار لین و همکاران [۹] پیروی می‌کنیم. به طور خاص، مقادیر متریک‌های سطح کامیت (به جز متریک بولین FIX) را با استفاده از تابع scale در R استانداردسازی می‌کنیم. همچنین، همانطور که حسن پیشنهاد کرده است [۳۴]، آنتروپی را با log2n نرمال‌سازی می‌کنیم تا تاثیر تفاوت تعداد فایل‌ها در تغییرات مختلف را خنثی کنیم. برای متوازن‌سازی کلاس‌ها و غلبه بر محدودیت‌های نمونه‌برداری افزایشی و نمونه‌برداری کاهشی از تکنیک SMOTE استفاده می‌کنیم.

پارامترها و ابزارهای آزمایش: تکنیک SMOTE دارای پارامترهای قابل تنظیمی است، اما همانطور که در [۲۷] نشان داده شده، تغییر این پارامترها تاثیر معناداری بر یافته‌ها ندارد. بنابراین، در این مطالعه، از مقادیر پیش‌فرض برای پارامترهای این الگوریتم استفاده شده است.

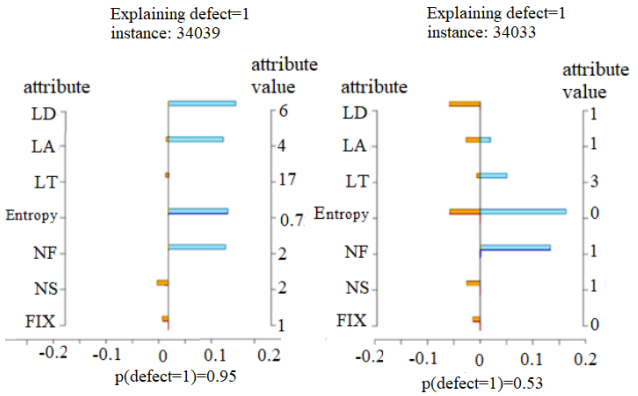
پارامترهای پیش‌فرض برای الگوریتم PH عبارت‌اند از: آستانه هشدار $\beta = 0.01$ (متناظر با نرخ مجاز هشدار کاذب)، حساسیت $\delta = 0.001$ (نرخ خطای قابل قبول) و فاکتور محوشوندگی $\gamma = 0.999$ (وزن به‌روزرسانی برای مقادیر تاریخی) [۱۷]. با پیکربندی پارامترهای الگوریتم PH با این مقادیر پیش‌فرض، روش‌های پیشنهادی، پایه و مرجع نقاط تغییر را در فواصل مساوی تشخیص می‌دهند. این یافته‌ها حاکی از آن است که این نقاط، مثبت‌های کاذب هستند نه نقاط تغییر واقعی. برای رفع این مشکل، نرخ مجاز هشدار کاذب الگوریتم PH را به $\beta = 0.1$ افزایش می‌دهیم تا احتمال هشدارهای کاذب کاهش یابد و بررسی CD با استفاده از مقادیر دیگر این پارامتر نیز توصیه می‌شود. پارامتر β در آزمون PH، نرخ مجاز هشدار کاذب را کنترل کرده و در نتیجه به طور مستقیم بر حساسیت روش تشخیص CD تأثیر می‌گذارد. به طور خاص، کاهش β، تشخیص‌گر را حساستر می‌کند که منجر به شناسایی حتی نوسانات جزئی (که برخی از آن‌ها ممکن است FA باشند) می‌شود. در مقابل، افزایش β، حساسیت به تغییرات کوچک را کاهش داده و تعداد نقاط تغییر شناسایی‌شده را کم می‌کند و تشخیص را بر روی رانش‌های معنادارتر متمرکز می‌سازد. در روش پیشنهادی، هنگامی که از مقادیر β کوچکتر از ۰/۱(مانند ۰/۰۱ یا ۰/۰۰۱ و غیره) استفاده می‌شود، نقاط تغییر به طور پیوسته و بسته به طول مجموعه‌داده، در حدود موقعیت‌های ۲۹، ۵۸، ۸۷، ۱۱۶، ۱۴۵ و به همین ترتیب شناسایی شدند. فاصله ثابت تقریباً

۲۹ مشاهده‌ای بین این نقاط، نشان می‌دهد که این موارد به احتمال زیاد FA هستند و نه CD واقعی. با افزایش β به ۰/۱، نقاط تغییر شناسایی‌شده در موقعیت‌های متنوع‌تری در طول جریان داده رخ دادند. بر اساس [۳۱]، تغییرپذیری و عدم منظم بودن این نقاط دلالت بر آن دارد که آن‌ها با رانش‌های مفهوم واقعی مطابقت دارند و نه تشخیص‌های کاذب. بسته به اینکه چه میزان حساسیت به تغییرات کوچک مورد نظر است، می‌توان β را متناسب با آن تنظیم کرد. مسلماً مقادیر بالاتر β، حساسیت به رانش‌های جزئی را کاهش می‌دهد. با این حال، از آنجا که همه روش‌های پایه و پیشنهادی و مرجع برای مقایسه از مقادیر یکسان β استفاده می‌کنند، ارزیابی نتایج، یکپارچه و بدون سوگیری باقی می‌ماند. بنابراین، مقدار β به طور یکسان بر همه روش‌ها تأثیر می‌گذارد و تغییر آن، ارزیابی مدل را مخدوش نمی‌سازد.

تکنیک‌های فرامدلی[۳۲] مورد مطالعه شامل IME و BD در محیط R پیاده‌سازی شده‌اند. تکنیک SMOTE نیز با استفاده از بسته نرم‌افزاری R و تابع SMOTE پیاده‌سازی شده است [۳۷]. این پیکربندی بر اساس مطالعات پیشین انتخاب شده تا امکان مقایسه عادلانه بین روش‌های مختلف فراهم گردد.

۵- نتایج آزمایش‌ها

در این بخش، یافته‌های مربوط به هر سوال تحقیق (RQ) مورد بررسی قرار می‌گیرد. در این مقاله، هدف اصلی کشف CD از طریق بررسی ناسازگاری‌های تفسیر نمونه‌ها در طول زمان است. برای دستیابی به این هدف، بردار تفسیر برای هر نمونه جدید تولیدشده محاسبه می‌شود. این رویکرد نوین امکان شناسایی به‌موقع تغییرات مفهوم را قبل از تأثیر منفی بر عملکرد مدل فراهم می‌سازد. در شکل ۴، تفسیر برای دو نمونه مختلف ارائه شده است. زیرشکل سمت چپ نشان می‌دهد که نمونه مورد نظر با احتمال ۹۵٪ معیوب است. میزان تاثیر هر ویژگی با مقدار خاص بر پیش‌بینی کلاس هدف توسط میله‌های آبی (تأثیر مثبت) و نارنجی (تأثیر منفی) نمایش داده شده است. به عنوان مثال، میزان تأثیر ویژگی nld=2.6 با احتمال ۹۵٪ تعلق به کلاس معیوب، حدود ۰,۲ است که از سایر ویژگی‌ها بیشتر می‌باشد. به طور کلی، میله‌های آبی نشان‌دهنده ویژگی‌هایی هستند که احتمال تعلق نمونه به کلاس معیوب را افزایش می‌دهند، در حالی که میله‌های نارنجی نشان‌دهنده ویژگی‌هایی هستند که این احتمال را کاهش می‌دهند. این نمایش بصری به درک چگونگی تأثیر هر ویژگی بر تصمیم‌گیری مدل کمک می‌کند.



شکل ۴: خروجی تفسیر برای دو نمونه مختلف. احتمال پیش‌بینی‌شده تعلق هر نمونه به کلاس مثبت (معیوب) در زیر هر شکل نمایش داده شده است.

برای پاسخ به سوالات تحقیق، نتایج روش پیشنهادی را با استفاده از معیارهای عملکرد استاندارد CDD ارزیابی می‌کنیم. معیارهای مناسب ارزیابی برای تشخیص CD بر اساس پژوهش‌های بیفت و همکاران به شرح زیر است [۳۸, ۳۹].

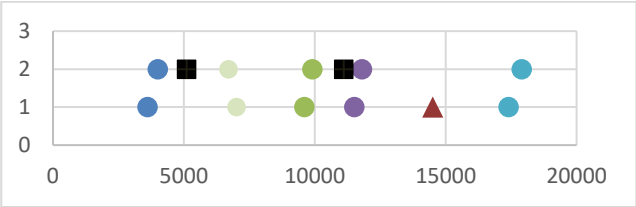
- **دقت تشخیص تغییر مفهوم (CDD_Accuracy):** این معیار نشان‌دهنده میزان دقت روش پیشنهادی در تشخیص صحیح نقاط CD است. هرچه این مقدار بالاتر باشد، نشان‌دهنده عملکرد بهتر روش در تمایز بین تغییرات واقعی مفهوم و نوسانات تصادفی در داده‌ها می‌باشد. این معیار با نسبت تشخیص‌های صحیح به کل تشخیص‌ها محاسبه می‌شود.
- **میانگین زمان تشخیص (MTD[۳۳]):** این شاخص، میانگین تأخیر بین تشخیص نقطه تغییر مفهوم و موقعیت واقعی وقوع آن را اندازه‌گیری می‌کند. یک سیستم تشخیص مناسب باید دارای مقدار MTD کوچکی باشد تا بتواند تغییرات را به موقع شناسایی کند.
- **میانگین زمان بین هشدارهای کاذب (MTFA[۳۴]):** این معیار مدت زمان بین هشدارهای کاذب (FA[۳۵]) را در مواقعی که هیچ تغییر مفهومی رخ نداده است، مشخص می‌کند. یک سیستم تشخیص مطلوب باید دارای مقدار MTFA بالایی باشد تا از ایجاد هشدارهای نادرست جلوگیری کند و قابلیت اطمینان سیستم را افزایش دهد.
- **نسبت میانگین زمان (MTR[۳۶]):** این شاخص تعادل بین حساسیت و استحکام سیستم را اندازه‌گیری می‌کند. یک سیستم تشخیص مناسب باید دارای مقدار MTR بالایی باشد که از طریق معادله (۱) محاسبه می‌شود. مقدار بالای این شاخص نشان‌دهنده توازن مناسب بین توانایی تشخیص تغییرات واقعی و مقاومت در برابر هشدارهای کاذب است.

$$MTR = \frac{MTFA}{MTD} \times CDD_Accuracy \quad (۱)$$

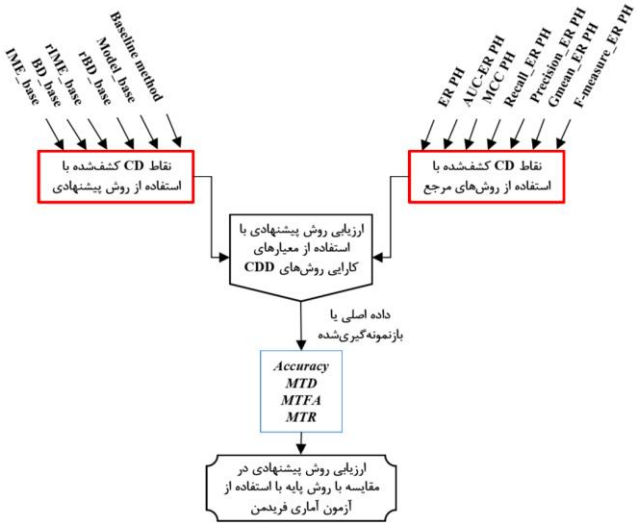
برای ارزیابی روش پیشنهادی، معیارهای عملکرد روش CDD را محاسبه می‌کنیم. در ادامه، ساز و کار این معیارها را با یک مثال عددی که در شکل ۵ آمده است، روشن‌تر می‌کنیم. فرض می‌کنیم نقاط روی محور ۱ عمودی، نقاط CD واقعی (بدست آمده از یک روش مرجع) از فایلی با ۲۰ هزار رکورد (کامیت) هستند و نقاط روی محور ۲، CD های کشف‌شده با روش پیشنهادی هستند. برای تطابق دادن هر نقطه CD کشف‌شده با نقطه CD اصلی، فاصله آن با همه نقاط CD اصلی را به دست می‌آوریم؛ به هر کدام نزدیک‌تر بود، یعنی آن CD به درستی کشف شده است و با یک رنگ در شکل ۵ نمایش داده می‌شوند. مثلا نقطه اول محور ۲ همرنگ با نقطه اول محور ۱ هست، پس به عنوان نقطه CD کشف‌شده اول شناسایی می‌شود. همانطور که از شکل ۵ مشخص است به ازای همه نقاط CD روی محور ۱، یک نقطه همرنگ روی محور ۲ داریم به جز نقطه قرمز رنگ که با شکل مثلث نشان داده شده است. پس می‌توان گفت، این نقطه CD با استفاده از روش پیشنهادی کشف نشده است. نقاط مربع سیاه رنگ، نقاطی هستند که از نظر فاصله نمی‌توان به هیچ یک از نقاط CD نسبت داد پس به عنوان FA شناسایی می‌شوند.

شکل ۶ فرآیند ارزیابی را نشان می‌دهد. در سمت چپ شکل، نام حالت‌های مختلف روش پیشنهادی و روش پایه آورده شده که با استفاده از خروجی روش‌های مرجع (نام آنها در سمت راست شکل ۶ آمده) ارزیابی شده‌اند. این حالت‌های مختلف به تفصیل در بخش ۳,۲ توضیح داده شده‌اند. در نهایت، نتایج حالت‌های مختلف با استفاده از آزمون فریدمن[۳۷] و آزمون تعقیبی نمنی[۳۸]، همانطور که دمسار توصیه کرده است [۴۰]، مقایسه شده‌اند. آزمون فریدمن [۴۱] یک تحلیل آماری برای ارزیابی و مقایسه چندین مجموعه نتیجه است. در این پژوهش، برای آزمون نمنی از k=3 الگوریتم و N=20 گروه آزمایشی استفاده شده است. به طور خلاصه، روش ما شامل مراحل زیر است:

۱) تقسیم‌بندی داده‌ها: داده‌های آزمون برای هر یک از ۲۰ مجموعه‌داده به گروه‌های متوالی ۱۰۰ تایی تقسیم می‌شوند، (همانطور که در شکل ۷ نشان داده شده و در بخش ۳-۱ نیز مفصل بیان شد). برای مثال، اگر یک مجموعه‌داده شامل ۹۰۰۰ کامیت باشد، ۹۰ گروه داده ایجاد خواهد شد.



شکل ۵: مثال عددی برای نشان‌دادن سازوکار معیارهای عملکرد روش CDD



شکل ۶: روش ارزیابی

S1-S100 Group1	S101-S200 Group2	S201-S300 Group3	...	S8901-S9000 Group9

شکل ۷: در گروه اول، نمونه‌های شماره‌گذاری‌شده از ۱ تا ۱۰۰ (S1 - S100) به ترتیب زمانی تا انتها مرتب شده‌اند (این فرآیند به صورت جداگانه برای تمامی مجموعه‌داده‌ها انجام می‌شود). در اینجا S نشان‌دهنده یک نمونه است.

۲) پیش‌پردازش داده‌ها: این مرحله (که در بخش ۴ به تفصیل توضیح داده شده) شامل تغییر مقیاس داده‌ها و بازنمونه‌گیری می‌باشد. این مرحله برای آماده‌سازی داده‌ها و اطمینان از کیفیت جهت تحلیل‌های بعدی ضروری است.

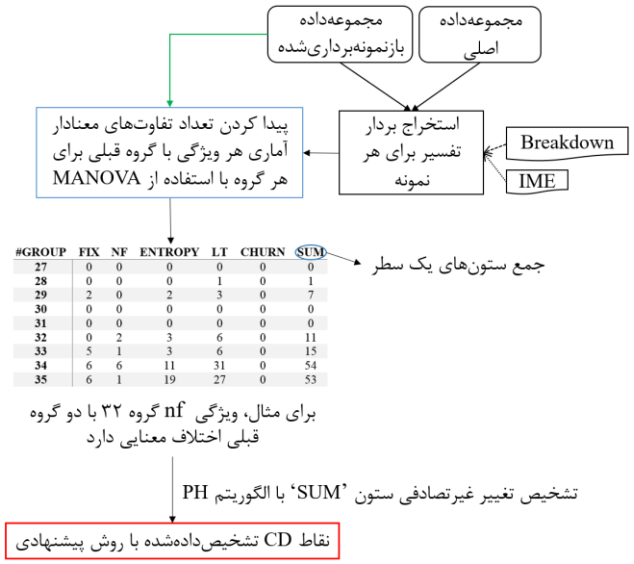
۳) استخراج بردار تفسیر نمونه‌ها: در این مرحله، برای هر نمونه از داده‌های آزمون، یک بردار تفسیر استخراج می‌شود. بردار تفسیر استخراج‌شده، اطلاعات ارزشمندی درباره اهمیت نسبی ویژگی‌های مختلف در پیش‌بینی‌های مدل ارائه می‌دهد و پایه‌ای برای تحلیل تغییرات مفهومی فراهم می‌سازد.

۴) شناسایی تفاوت‌های آماری معنادار: این مرحله، گروه‌های داده آزمونی که از نظر آماری تفاوت معنادار با گروه‌های قبلی خود دارند را شناسایی می‌کند. این کار با استفاده از روش

MANOVA انجام می‌شود که رابطه بین چندین متغیر وابسته (متریک‌های نرم‌افزاری) و یک متغیر مستقل (گروه‌ها) را تحلیل می‌کند. مقدار p-value کمتر از ۰/۰۵ نشان‌دهنده تفاوت معنادار در توزیع داده‌ها بین یک گروه با گروه‌های قبلی است. این فرآیند در شکل ۸ نمایش داده شده است. مقادیر تفاوت‌ها در جدول داخل شکل ۸ ارائه شده‌اند. هر ستون نشان‌دهنده یک ویژگی و هر سطر مربوط به یک گروه داده آزمون است. به عنوان مثال، توزیع آماری بردار ویژگی FIX در گروه ۲۹ با دو گروه قبلی تفاوت معناداری دارد. مجموع مقادیر هر سطر در ستون آخر (ستون مجموع) محاسبه و نمایش داده می‌شود.

۵) تشخیص نقاط CD با استفاده از آزمون PH: در ستون sum شکل ۸، سطری که مقادیر آن به طور معناداری بزرگ‌تر از سطرهای قبلی باشد، نشان‌دهنده وقوع CD است. زمانی که سطرهای متوالی در ستون sum اعداد بزرگ‌تری نسبت به قبل داشته باشند، این تفاوت غیرتصادفی و معنادار در نظر گرفته می‌شود. این نقاط تفاوت فاز با استفاده از الگوریتم تشخیص تغییر PH شناسایی می‌شوند. اگر تنها یک سطر تفاوت قابل توجهی داشته باشد، ممکن است تصادفی باشد. به عنوان مثال، مقادیر گروه ۲۹ در ستون sum که در حال افزایش هستند، نشان‌دهنده تغییر مفهوم است. به جای پیاده‌سازی الگوریتم PH تنها برای تشخیص تغییر معنادار در ستون sum، می‌توان آن را برای هر ویژگی به صورت جداگانه اعمال کرد. این رویکرد امکان بررسی ویژگی‌های خاص تأثیرگذار در وقوع CD را فراهم می‌کند. ما بررسی این ایده را برای پژوهش‌های آینده باقی می‌گذاریم.

۶) شناسایی نقاط CD با روش پیشنهادی: در این مرحله، نقاط CD با استفاده از روش پیشنهادی در دو حالت داده‌های اصلی و بازنمونه‌گیری‌شده برای تمامی ۲۰ مجموعه‌داده، و هم بر روی داده‌های تفسیرنشده و هم داده‌های تفسیرشده، شناسایی می‌شوند. این رویکرد جامع، امکان مقایسه بین حالت‌های مختلف را فراهم می‌کند و به درک بهتر تأثیر متقابل تفسیر داده‌ها و تکنیک بازنمونه‌گیری بر دقت و قابلیت اطمینان تشخیص CD کمک می‌نماید.



#GROUP	FIX	NF	ENTROPY	LT	CHURN	SUM
27	0	0	0	0	0	0
28	0	0	0	1	0	1
29	2	0	2	3	0	7
30	0	0	0	0	0	0
31	0	0	0	0	0	0
32	0	2	3	6	0	11
33	5	1	3	6	0	15
34	6	6	11	31	0	54
35	6	1	19	27	0	53

شکل ۸: مراحل CDD با استفاده از الگوریتم‌های MANOVA و PH

۷) جمع‌آوری معیارهای عملکرد مدل RF: در مرحله هفتم، مقادیر معیارهای عملکرد مدل RF (آموزش‌دیده در طول زمان) برای هر مجموعه‌داده جمع‌آوری می‌شود. این معیارها شامل ER، AUC، Gmean، Precision، F-measure، Recall و MCC هستند که به طور معمول در پژوهش‌های حوزه SDP برای ارزیابی پایداری و کارایی مدل مورد استفاده قرار می‌گیرند. این داده‌ها به عنوان معیارهای استانداردی برای سنجش کیفیت و قابلیت اطمینان پیش‌بینی‌های مدل در نظر گرفته می‌شوند.

۸) شناسایی نقاط CD با روش‌های مرجع: این مرحله شامل شناسایی نقاط CD برای هر یک از ۲۰ مجموعه‌داده از طریق پایش هر یک از معیارهای عملکرد ذکر شده و استفاده از آزمون آماری PH است. این روش (که به عنوان رویکرد مرجع شناخته می‌شود)، امکان مقایسه عملکرد روش پیشنهادی را با روش‌های استاندارد فراهم می‌کند. در این چارچوب، تغییرات معنادار آماری در معیارهای عملکرد، به عنوان نشانه‌ای از وقوع CD در نظر گرفته می‌شوند.

۹) محاسبه و ارزیابی معیارهای CDD: در مرحله نهم، معیارهای ارزیابی روش‌های CDD شامل CDD_Accuracy، MTD، MTFA و MTR محاسبه می‌شوند. این معیارها برای ارزیابی نتایج روش پیشنهادی (مرحله ۶) با نتایج روش‌های مرجع (مرحله ۸) مورد استفاده قرار می‌گیرند. محاسبه این شاخص‌ها امکان ارزیابی جامع عملکرد روش پیشنهادی را از جنبه‌های مختلف شامل دقت، سرعت تشخیص و قابلیت اطمینان فراهم می‌کند.

۱۰) مقایسه روش پیشنهادی با روش پایه: در نهایت، در این مرحله، روش پیشنهادی با استفاده از آزمون فریدمن با روش‌های پایه (اعمال شده روی داده‌های تفسیرنشده) مقایسه می‌شود تا برتری یا عدم برتری آن از نظر آماری به طور دقیق مشخص گردد. این ارزیابی به درک بهتر کارایی روش پیشنهادی در تشخیص دقیق تغییرات مفهوم کمک می‌کند.

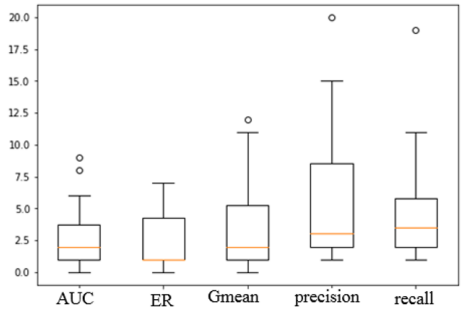
همانطور که پیش‌تر اشاره شد، روش‌های مرجع از طریق پایش معیارهای عملکرد (چه وابسته به آستانه و چه مستقل از آستانه) که در طول زمان محاسبه می‌شوند، به دست می‌آیند. این معیارها از ارزیابی الگوریتم یادگیری ماشین RF حاصل می‌شوند. در این بخش، تأثیر بازنمونه‌گیری داده‌های آموزشی بر این معیارها را بررسی می‌کنیم. ابتدا مقادیر هر معیار عملکرد را از مدل یادگیری ماشین افزایشی RF در طول زمان برای ۲۰ مجموعه‌داده استخراج می‌کنیم، سپس از آزمون ویلکاکسون [۴۲] برای تعیین برتری معنادار آماری عملکرد هر معیار قبل یا بعد از بازنمونه‌گیری استفاده می‌کنیم. این آزمون ناپارامتریک به منظور مقایسه تفاوت‌های زوجی بین دو شرایط مختلف (داده‌های اصلی در مقابل داده‌های بازنمونه‌گیری شده) به کار می‌رود. جدول ۲ تعداد مواردی را نشان می‌دهد که عملکرد طبقه‌بند قبل یا بعد از بازنمونه‌گیری داده‌های آموزشی به‌طور معناداری بهبود یافته است. به‌عنوان مثال، سطر اول نشان می‌دهد که دقت در طول زمان برای ۱۲ مورد از ۲۰ مجموعه‌داده پس از نمونه‌گیری بهتر شده، در حالی که دقت در ۳ مجموعه‌داده نسبت به قبل از بازنمونه‌گیری بهبود یافته است. این نتایج حاکی از آن است که بازنمونه‌گیری موجب افت عملکرد معیارهای مستقل از آستانه (دو سطر اول جدول ۲ که با رنگ آبی مشخص شده‌اند) و افزایش عملکرد معیارهای وابسته به آستانه (سه سطر آخر جدول ۲ که با رنگ سبز نشان داده شده‌اند) می‌شود. در نتیجه، متوازن‌سازی تأثیر مثبتی بر عملکرد معیارهای وابسته به آستانه دارد. با توجه به نامتوازن بودن داده‌ها، بررسی عملکرد روش پیشنهادی بر روی داده‌های بازنمونه‌گیری‌شده ضروری به نظر می‌رسد.

جدول ۲: مقایسه کارایی کلاسبند RF قبل و بعد از بازنمونه‌گیری

	قبل	بعد
Accuracy	۱۲	۳
AUC	۱۳	۳
Recall	۴	۱۴
F Measure	۴	۱۰
Gmean	۵	۱۱

ارزیابی پایداری معیارهای عملکرد مدل آموزشی JIT-SDP

برای سنجش پایداری معیارهای عملکرد مدل آموزشی JIT-SDP، وقوع CD را در خروجی هر معیار عملکرد در طول زمان با استفاده از ۲۰ مجموعه‌داده تحلیل می‌کنیم. این فرآیند برای هر معیار ۲۰ مقدار تولید می‌کند که هر کدام نشان‌دهنده تعداد CD شناسایی‌شده در یک مجموعه‌داده خاص است. تعداد کمتر CD برای هر معیار نشان‌دهنده پایداری بیشتر آن در طول زمان است. برای نمایش بصری، نمودار جعبه‌ای وقوع CD حاصل از مجموعه‌داده‌های مختلف را ترسیم می‌کنیم (شکل ۹). نمودارهای کوچک‌تر و پایین‌تر نشان‌دهنده معیارهای عملکرد پایدارتر هستند. نتایج حاکی از آن است که AUC کم‌ترین میزان ناپایداری را دارد، در حالی که معیار Precision بیشترین میزان ناپایداری را نشان می‌دهد. به عبارت دیگر، معیار Precision در پروژه‌های مختلف متنوع‌ترین تعداد CD را شناسایی کرده است. همچنین تحقیقات پیشین [۲۷] نشان داده‌اند که اعمال تکنیک‌های متوازن‌سازی تأثیر قابل‌توجهی بر مقادیر AUC ندارد. بنابراین نتیجه می‌گیریم که AUC معمولاً از ناپایداری کمی برخوردار است. معیارهای ER و Recall نیز در مقایسه با AUC ناپایداری زیادی ندارند، اگرچه معیار Recall به آستانه نیز وابسته است.



شکل ۹: تحلیل مقادیر معیارهای کارایی در طول زمان و روی ۲۰ مجموعه داده

۵-۱- RQ1: عملکرد روش CDD پیشنهادی با استفاده از الگوریتم‌های تفسیر مختلف در مقایسه با روش‌های مرجع چگونه است؟

در این بخش، نقاط CD را با استفاده از حالت‌های مختلف روش پیشنهادی در ۲۰ مجموعه‌داده شناسایی کرده و نتایج را با روش‌های مرجع مبتنی بر معیارهای وابسته و مستقل از آستانه ارزیابی می‌کنیم. به‌عنوان مثال، جدول ۳ دقت میانگین (CDD_Accuracy) حالت‌های مختلف روش پیشنهادی را در مقایسه با روش‌های مرجع نشان می‌دهد. نام حالت‌های مختلف

روش پیشنهادی در سطر اول و نام روش‌های مرجع در ستون اول آورده شده‌اند. همانطور که در جدول ۳ مشاهده می‌شود، دقت متوسط حالت مبتنی بر تفسیر نمونه با الگوریتم IME (با نام IME_base) و همچنین روش پایه (Baseline) از حالت BD_base پیشی گرفته‌است. علاوه‌براین، تمامی حالت‌ها به دقت رضایت‌بخشی بیش از ٪۹۰ دست یافته‌اند. جدول ۴ معیار MTD را برای حالت‌های مختلف گزارش می‌کند. این معیار فاصله بین نقاط CD شناسایی‌شده توسط روش‌های پیشنهادی و نقاط تولیدشده توسط روش‌های مرجع را محاسبه می‌کند. مقدار کمتر MTD نشان‌دهنده عملکرد بهتر و تشخیص سریع‌تر است. در این مطالعه، این فاصله برای تمام نقاط CD شناسایی‌شده محاسبه شده و از آزمون فریدمن به همراه آزمون تعقیبی نمنی استفاده شده تا مشخص شود کدام حالت‌ها از نظر معیار MTD به طور معناداری بهتر از سایرین عمل می‌کنند ($P < 0.05$). در جدول ۴، تفاوت‌های معنادار با علامت ستاره (*) نشان داده شده‌اند که تعداد بیشتر ستاره‌ها بیانگر عملکرد بهتر آن حالت در مقایسه با حالت‌های دیگر است. همانطور که در جدول ۴ مشاهده می‌شود، حالت‌های مبتنی بر تفسیر نمونه (BD_base و IME_base) از نظر آماری عملکرد بهتری نسبت به روش پایه دارند، به استثنای اینکه BD_base فقط در مورد AUC عملکرد ضعیف‌تری نشان می‌دهد. علاوه بر این، حالت مبتنی بر IME به طور سازگار عملکرد بهتری نسبت به BD_base از خود نشان می‌دهد. جدول ۵ مقادیر میانگین معیار MTFA را برای تمامی حالت‌های ارزیابی‌شده ارائه می‌دهد. این معیار فاصله زمانی بین هشدارهای کاذب متوالی را نشان می‌دهد؛ بنابراین مقدار بالاتر آن بیانگر عملکرد بهتر مدل مربوطه است. نکته قابل توجه این است که دقت CDD و MTD رابطه معکوسی با MTFA دارند. بهبود در دقت CDD یا کاهش MTD نشان‌دهنده توانایی بهتر حالت در تشخیص نقاط تغییر است. اما این بهبود ممکن است به طور ناخواسته منجر به افزایش هشدارهای کاذب شده و در نتیجه MTFA را کاهش دهد. بنابراین، حالت‌هایی که در دقت CDD و MTD عملکرد قوی‌تری دارند، لزوماً نتایج بهتری در MTFA کسب نمی‌کنند و برعکس. با این وجود، حتی در این معیار نیز حالت‌های مبتنی بر تفسیر نمونه عملکرد بهتری نسبت به روش پایه از خود نشان می‌دهند. این یافته حاکی از آن است که روش پیشنهادی نه تنها دقت بالاتری دارد، بلکه در مقایسه با روش پایه (که مبتنی بر توزیع داده‌های تفسیرنشده است)، هشدارهای کاذب کمتری نیز تولید می‌کند. جدول ۶ مقادیر میانگین MTR را برای حالت‌های ارزیابی‌شده نشان می‌دهد. MTR که با استفاده از معادله (۲) محاسبه می‌شود، بیانگر تعادل میان سایر معیارهای عملکرد است. بر اساس معادله (۲)، مقادیر بالاتر MTR نشان‌دهنده عملکرد کلی بهتر و تعادل مطلوب‌تر بین حساسیت تشخیص و دقت است. نتایج، حاکی از آن است که روش مبتنی بر IME مقادیر MTR بالاتری نسبت به هر دو روش پایه و حالت BD_base در تمام سناریوهای آزمایشی کسب می‌کند که نشانگر توانایی برتر آن در حفظ عملکرد بهینه در معیارهای ارزیابی مختلف می‌باشد.

جدول ۳: نتایج ارزیابی معیار CDD_Accuracy(%) حالت‌های مختلف روش پیشنهادی و روش پایه(Baseline) در مقایسه با روش‌های مرجع (Benchmark)

Benchmark method	Baseline method	IME_base	BD_base
ER PH-Er PH	۱۰۰	۱۰۰	۹۳
AUC-Er PH	۱۰۰	۹۹	۹۳
Gmean-Er PH	۹۹	۹۸	۹۰
Precision-Er PH	۹۶	۹۳	۹۱
Recall-Er PH	۱۰۰	۱۰۰	۹۱
MCC-Er PH	۱۰۰	۹۹	۹۱
F Measure-Er PH	۹۲	۹۰	۸۲

جدول ۴: نتایج ارزیابی معیار MTD حالت‌های مختلف روش پیشنهادی و روش پایه(Baseline) در مقایسه با روش‌های مرجع (Benchmark)

Benchmark method	Baseline method	IME_base	BD_base
ER PH	*۷	**۵	*۷
AUC-Er PH	*۸	*۸	۱۰
Gmean-Er PH	*۷	**۵	*۷
Precision-Er PH	۷	۷	۷
Recall-Er PH	۵	*۲	۲
MCC-Er PH	۴	۲	۲
F Measure-Er PH	۱۱	۹	۱۳

جدول ۵: نتایج ارزیابی معیار MTFA حالت‌های مختلف روش پیشنهادی و روش پایه(Baseline) در مقایسه با روش‌های مرجع (Benchmark)

Benchmark method	Baseline method	IME_base	BD_base
ER PH	۶۶۴	۷۵۳/۲۲	۷۸۱
AUC-Er PH	۵۰۲/۲۱	۵۳۵/۸۳	۵۶۱/۷۹
Gmean-Er PH	۷۳۶/۵۳	۸۳۴/۹۴	۷۹۴/۰۵
Precision-Er PH	۸۸۰/۵	۹۷۲/۲۲	۹۴۴/۴۴
Recall-Er PH	۸۳۳/۳۳	۹۴۴/۴۴	۸۸۸/۸۹
MCC-Er PH	۷۵۱/۶۱	۸۳۴/۹۴	۸۰۷/۱۷
F Measure-Er PH	۸۶۱/۱۱	۹۴۴/۴۴	۹۱۶/۶۷

جدول ۶: نتایج ارزیابی معیار MTR حالت‌های مختلف روش پیشنهادی و روش پایه(Baseline) در مقایسه با روش‌های مرجع (Benchmark)

Benchmark method	Baseline method	IME_base	BD_base
ER PH-Er PH	۹۴/۸۶	۱۵۰/۶۴	۱۰۳/۷۹
AUC-Er PH	۶۲/۸۴	۶۶/۳۱	۵۲/۲۵
Gmean-Er PH	۱۰۳/۹۲	۱۶۲/۸۶	۱۰۲/۲۳
Precision-Er PH	۱۲۰/۲	۱۲۹/۶۳	۱۲۲/۹۳
Recall-Er PH	۱۶۶/۶۷	۴۷۲/۲۲	۴۰۶/۰۴
MCC-Er PH	۱۸۷/۹	۴۱۲/۱۹	۳۶۷/۸۲
F Measure-Er PH	۷۲/۰۲	۹۴/۴۴	۵۷/۸۲

نتیجه‌گیری RQ1: با توجه به اینکه یکی از مزایای کلیدی روش پیشنهادی، توانایی آن در تشخیص CD روی داده‌های بدون برچسب است، یک مطالعه آزمایشی بر اساس دو الگوریتم تفسیر نمونه طراحی کرده و با ۷ روش مرجع مبتنی بر پایش معیارهای عملکرد ارزیابی نمودیم. نتایج نشان می‌دهد روش پیشنهادی، روش پایه مبتنی بر پایش توزیع داده‌ها را از نظر معیارهای MTD و MTR بهبود می‌بخشد. تحلیل‌ها نشانگر رابطه معکوس ذاتی بین معیارهای دقت تشخیص و نرخ هشدارهای کاذب است. نکته قابل توجه اینجاست که روش پیشنهادی بهبود مستمری در تمام معیارها از جمله MTFA نشان می‌دهد. همچنین نتایج نشان می‌دهد در میان الگوریتم‌های تفسیر موجود در تحقیقات، استفاده از الگوریتم IME کارایی روش CDD را در مقایسه با الگوریتم تفسیر BD بهبود می‌بخشد که به طور میانگین به دقت ٪۹۸ دست یافته است.

۵-۲- RQ2 : عملکرد روش پیشنهادی مبتنی بر داده‌های بازنمونه‌گیری‌شده در مقایسه با روش‌های مرجع وابسته و مستقل از آستانه چگونه است؟

در این بخش، نتایج روش پیشنهادی مبتنی بر داده‌های بازنمونه‌گیری‌شده را با خروجی‌های روش‌های مرجع (که خود به دو دسته وابسته به آستانه و مستقل از آستانه تقسیم می‌شوند) مقایسه می‌کنیم. یافته‌های جالبی از رتبه‌بندی عملکرد روش‌ها روی ۲۰ مجموعه‌داده ارائه می‌دهیم. همانطور که در بخش قبل اشاره شد، مطالعات نشان داده‌اند که عملکرد و پایداری این دو دسته از معیارها در مدل‌های افزایشی می‌تواند متفاوت باشد. از سوی دیگر، بررسی پایداری مدل روی داده‌های بازنمونه‌گیری‌شده نیز چالشی در این حوزه محسوب می‌شود. برای مواجهه با این دو چالش، RQ2 را مطرح می‌کنیم. با توجه به یافته‌های قبلی که نشان داد روش پیشنهادی عملکرد بهتری نسبت به روش پایه دارد، حال ارزیابی خود را به داده‌های بازنمونه‌گیری‌شده نیز گسترش می‌دهیم. همچنین با توجه به بهبود عملکرد روش مبتنی بر تفسیر نمونه با الگوریتم IME، یک حالت جدید با نام Model_base (متشکل از تفسیر مدل مبتنی بر الگوریتم IME روی داده‌های اصلی) را برای تحلیل مقایسه‌ای دقیق‌تر معرفی می‌کنیم.

برای پاسخ جامع به سوال تحقیق دوم، تشخیص CD را با اعمال انواع حالت‌های روش پیشنهادی روی هر دو مجموعه داده‌های آموزشی اصلی و بازنمونه‌گیری‌شده انجام می‌دهیم. جدول ۷ میانگین دقت را برای این پیاده‌سازی‌های مختلف روش پیشنهادی نشان می‌دهد. نتایج به وضوح نشان می‌دهد که حالت‌های IME_base و rIME_base به طور مداوم و معناداری عملکرد بهتری نسبت به حالت‌های مبتنی بر BD دارند. به‌طور خاص، حالت rIME_base به دقت بسیار بالایی (بیش از ٪۹۵) دست یافته است. این سطح از دقت نه تنها برتری روش‌های مبتنی بر تفسیر نمونه را نشان می‌دهد، بلکه مؤثر بودن استفاده از تکنیک بازنمونه‌گیری (مانند SMOTE) را برای بهبود تشخیص دقیق و قابل اطمینان CD تأیید می‌کند. نتایج ارائه شده در جدول ۸ که معیار عملکرد MTD را برای تمامی حالت‌ها نشان می‌دهد، به وضوح برتری روش پیشنهادی مبتنی بر تفسیر نمونه را تأیید می‌کند. مطابق با بحث‌های قبلی، مقادیر کمتر MTD نشان‌دهنده عملکرد تشخیص به‌موقع‌تر و اثربخشی بالاتر مدل است. با این معیار، فواصل تأخیر تشخیص برای تمامی نقاط CD شناسایی‌شده محاسبه گردیده و با استفاده از آزمون فریدمن مقایسه‌های آماری دقیقی انجام شده است. عملکرد برتر آماری معنادار ($p > 0.05$) با نماد ستاره (*) نشان داده شده و تعداد ستاره‌ها بیانگر میزان برتری عملکرد است. حالت‌هایی با چندین ستاره، قابلیت‌های تشخیص سریع‌تر و پایدارتری را نشان می‌دهند. این یافته مزیت عملی روش ما را در کاربردهای تشخیص وابسته به زمان اثبات می‌کند. به طور خاص، حالت‌های مبتنی بر تفسیر نمونه به‌طور معناداری از هر دو روش پایه و حالتModel_base عملکرد بهتری دارند، به‌جز حالت BD_base که فقط در معیارهای AUC عملکرد ضعیف‌تری نشان داد. پیاده‌سازی‌های بازنمونه‌گیری‌شده (rIME_base و rBD_base) نتایج بهتری نسبت به حالت‌های IME_base و BD_base و روش پایه و حالت Model_base در تمام مقایسه‌های روش‌های مرجع کسب کردند. نکته قابل توجه این که حالت‌های مبتنی بر IME، هم در داده‌های اصلی و هم بازنمونه‌گیری‌شده، عموماً عملکرد تشخیص بهتری نسبت به حالت‌های مبتنی بر BD نشان دادند که مزیت‌های ذاتی الگوریتم

جدول ۷: نتایج ارزیابی معیار CDD_Accuracy (%) برای روش‌های مبتنی بر داده ساده و متوازن‌شده و روش پایه

	Baseline method	IME_ base	BD_ base	rIME _base	rBD_ base	Model_ base
۱	۱۰۰	۱۰۰	۹۳	۹۹	۹۵	۹۹
۲	۱۰۰	۹۹	۹۳	۱۰۰	۹۷	۹۷
۳	۹۹	۹۸	۹۰	۹۷	۹۴	۹۸
۴	۹۶	۹۳	۹۱	۹۵	۹۳	۹۷
۵	۱۰۰	۱۰۰	۹۱	۹۸	۹۶	۹۹
۶	۱۰۰	۹۹	۹۱	۱۰۰	۹۶	۹۹
۷	۹۲	۹۰	۸۲	۹۶	۹۲	۹۶

جدول ۸: نتایج ارزیابی معیار MTD برای روش‌های مبتنی بر داده ساده و متوازن‌شده و روش پایه

	Baseline method	IME_ base	BD_ base	rIME _base	rBD_ base	Model _base
۱	۷*	۵**	۷*	۲***	۶*	۱۱
۲	۸*	۸*	۱۰	۳**	۱***	۶**
۳	۷*	۵**	۷*	۳***	۳**	۷
۴	۷	۷	۷	۶*	۶	۹
۵	۵	۲*	۲	۲*	۴	۹
۶	۴	۲	۲	۱*	۴	۷
۷	۱۱	۹	۱۳	۴***	۴***	۱۱

IME را برجسته می‌سازد. این یافته‌ها به‌طور جمعی دو بینش کلیدی را تأکید می‌کنند: اثربخشی بازنمونه‌گیری و استحکام معماری‌های مبتنی بر IME برای وظایف تشخیص تغییر مبتنی بر تفسیر، که با مقایسه‌های آماری دقیق ارائه‌شده در جدول ۸ نیز تأیید می‌شود. جدول ۹ عملکرد معیار MTFA را برای تمامی حالت‌های ارزیابی‌شده نمایش می‌دهد، که مقادیر بالاتر در آن نشان‌دهنده قابلیت بهتر در سرکوب هشدارهای کاذب است. نتایج حاکی از آن است که حالت‌های مبتنی بر تفسیر نمونه، هم در پیکربندی ساده (IME_base) و هم در حالت بازنمونه‌گیری‌شده (rIME_base) مقادیر MTFA به‌مراتب بالاتری نسبت به روش پایه و حالت Model_base کسب می‌کنند. این یافته‌ها تأیید می‌کنند که روش پیشنهادی دارای دو مزیت عملیاتی است: کاهش فراوانی هشدارهای کاذب و در عین حال حفظ حساسیت تشخیص بالا که در تحلیل‌های قبلی نشان داده شد. جدول ۱۰ تحلیل مقایسه‌ای عملکرد مدل‌ها را با استفاده از معیار MTR ارائه می‌دهد که بر اساس معادله ۲ محاسبه شده و مقادیر بالاتر آن نشان‌دهنده عملکرد کلی بهتر و تعادل مطلوب‌تر بین معیارهای مختلف است. نتایج نشان می‌دهد هر دو پیاده‌سازی ساده (IME_base) و بازنمونه‌گیری‌شده (rIME_base) روش پیشنهادی به طور مداوم عملکرد بهتری نسبت به حالت‌های مبتنی بر BD دارند (به استثنای روش مبتنی بر AUC)، نکته برجسته‌تر این که حالت IME_base بهبود عملکرد مضاعفی را نسبت به هر دو حالت IME_base و Model_base نشان می‌دهد.

جدول ۹. نتایج ارزیابی معیار MTFA برای روش‌های مبتنی بر داده ساده و متوازن‌شده و روش پایه

	Baseline method	IME_ base	BD_ base	rIME _base	rBD_ base	Model _base
۱	۶۶۴	۷۵۳	۷۸۱	۸۹۰	۸۳۶	۶۸۸
۲	۵۰۲	۵۳۵	۵۶۱	۶۶۴	۷۲۷	۴۸۴
۳	۷۳۶	۸۳۴	۷۹۴	۹۷۲	۹۷۲	۷۷۶
۴	۸۸۰	۹۷۲	۹۴۴	۹۷۲	۹۴۴	۹۴۴
۵	۸۳۳	۹۴۴	۸۸۸	۹۱۶	۸۸۸	۸۶۱
۶	۷۵۱	۸۳۴	۸۰۷	۸۶۱	۸۶۱	۷۵۱
۷	۸۶۱	۹۴۴	۹۱۶	۱۰۰۰	۹۷۲	۹۱۶

جدول ۱۰: نتایج ارزیابی معیار MTR برای روش‌های مبتنی بر داده ساده و متوازن‌شده و روش پایه

	Baseline method	IME_ base	BD_ base	rIME_ base	rBD_ base	Model_ base
۱	۹۴/۸۶	۱۵۰/۶	۱۰۳	۴۴۰/۸	۱۳/۴	۶۱/۹۹
۲	۶۲/۸۴	۶۶/۳۱	۵۲/۲	۲۲/۳۳	۷۰/۲	۷۸/۲۸
۳	۱۰۳/۹۲	۱۶۲/۸	۱۰۲	۳۱۲/۹	۳۰/۴	۱۰۸/۱۲
۴	۱۲۰/۲	۱۲۹/۶	۱۲۲	۱۵/۱۳	۱۴۷	۹۶/۱
۵	۱۶۶/۶۷	۴۷/۲	۴۰۶	۴۴/۰۲	۲۱/۹	۹۴/۵
۶	۱۸۷/۹	۴۱/۱	۳۶۷	۸۶۱/۱۱	۲۰/۱	۱۰۶/۰۱
۷	۷۲/۰۲	۹/۴۴	۵۷/۸	۲۴۰	۲۲۳/۶	۸۰

نتیجه‌گیری RQ2: این پژوهش ارزیابی جامعی از روش پیشنهادی ارائه می‌دهد و عملکرد آن را بر روی هر دو نوع مجموعه‌داده اصلی و بازنمونه‌گیری‌شده با استفاده از دو الگوریتم متمایز تفسیر نمونه تحلیل می‌کند. مقایسه‌ها با روش پایه و روش مبتنی بر تفسیر مدل (Model_base) سه یافته کلیدی را نشان می‌دهد: (۱) روش ما به دقت برتری (بهبود ٪۹۷) دست یافته است؛ به‌ویژه با عملکرد بهبودیافته در داده‌های بازنمونه‌گیری‌شده؛ (۲) به‌طور معناداری از روش پایه در معیارهای عملیاتی حیاتی شامل MTD و MTR پیشی می‌گیرد، که بازنمونه‌گیری این معیارها را بیشتر بهبود می‌بخشد؛ و (۳) علیرغم تقابل ذاتی بین عملکرد تشخیص و سرکوب هشدارهای کاذب (که با MTFA سنجیده می‌شود)، روش پیشنهادی عملکرد MTFA را به طور قوی حفظ می‌کند. اعتبارسنجی با استفاده از هر دو پارادایم ارزیابی وابسته و مستقل از آستانه، برتری قابل اطمینان روش

پیشنهادی نسبت به هفت روش مرجع را تأیید می‌کند، که پیاده‌سازی‌های بازنمونه‌گیری‌شده بهبود عملکرد مداوم را نشان می‌دهند. این نتایج به‌طور جمعی روش پیشنهادی مبتنی بر تفسیر را به‌عنوان راه‌حلی قوی تثبیت می‌کند که به‌طور مؤثری دقت تشخیص، به‌موقع بودن و قابلیت اطمینان را در شرایط عملیاتی متنوع متعادل می‌سازد.

۶- نتیجه‌گیری و کارهای آینده

مطالعات پیشین نشان داده‌اند که عملکرد مدل‌های SDP یا توزیع داده‌ها ممکن است به مرور زمان به دلیل بروز CD و عدم تطابق بین توزیع داده‌های آموزشی و داده‌های جدید بدون برچسب، کاهش یابد. این واگرایی به تدریج موجب کاهش اثربخشی مدل می‌شود، چرا که الگوهای یادگرفته‌شده اولیه، کمتر نمایانگر ویژگی‌های داده‌های در حال ورود خواهند بود. در نتیجه، مدل‌هایی که بر اساس داده‌های تاریخی آموزش دیده‌اند ممکن است در حفظ دقت پیش‌بینی هنگام اعمال بر داده‌های جدیدتر ناتوان باشند. برای غلبه بر این چالش، رویکردهای مرسوم تشخیص CD بر پایش مستمر عملکرد متمرکز است که با ردیابی مستقیم خطای دقت پیش‌بینی، معیاری قابل اعتماد برای ارزیابی افت مدل ارائه می‌دهد. با این حال، وابستگی این روش‌ها به داده‌های ورودی برچسب‌دار باعث ایجاد تأخیر در تشخیص می‌شود که با وجود مفید بودن به عنوان معیار ارزیابی، کاربرد عملی آن‌ها را در محیط‌های واقعی محدود می‌کند. این پژوهش روشی نوین برای پیش‌بینی CD در زمان واقعی ارائه می‌دهد که بدون نیاز به داده‌های آزمون برچسب‌دار عمل می‌کند. رویکرد پیشنهادی با بهره‌گیری از تکنیک‌های تفسیر نمونه، تغییرات مفهوم را از طریق تحلیل ناسازگاری‌های (تغییرات معنادار) تفسیر در طول زمان شناسایی می‌کند. این روش با حذف نیاز به برچسب‌های داده‌های جدید، امکان تشخیص به موقع تغییرات را فراهم می‌سازد و محدودیت‌های روش‌های مبتنی بر پایش عملکرد را مرتفع می‌کند. تفسیرها بر اساس برچسب‌های پیش‌بینی‌شده (و نه برچسب‌های واقعی) تولید می‌شوند و در نتیجه وابستگی به داده‌های برچسب‌دار را حذف می‌کنند. ارزیابی مقایسه‌ای در برابر هفت روش مرجع نشان می‌دهد که روش پیشنهادی با توجه به امکان تشخیص به‌موقع (MTD پایین)، دقت بالا (میانگین CDD_Accuracy ٪۹۷)، قابلیت اطمینان (افزایش فاصله بین هشدارهای کاذب – MTFA بالا) و استحکام (MTR بالا) نسبت به روش‌های مرسوم برتری قابل توجهی دارد. این مزیت‌ها، روش پیشنهادی را به گزینه‌ای مناسب برای سیستم‌های تشخیص تغییر در محیط‌های عملیاتی تبدیل می‌کند. نتایج نشان‌دهنده بهبودهای چشمگیر در مقایسه با روش‌های مبتنی بر توزیع نمونه و تفسیر مدل است. این روش با امکان‌پذیر ساختن تشخیص به‌موقع رانش در جریان‌های داده کاملاً بدون برچسب، محدودیت‌های اساسی روش‌های موجود را مرتفع می‌سازد و گامی مهم برای استقرار در سناریوهای دنیای واقعی محسوب می‌شود که در آن‌ها معمولاً داده برچسب‌دار در دسترس نیست. از سوی دیگر، برای حل مشکل عدم توازن ذاتی برچسب‌ها در مجموعه‌داده‌های SDP، روش پیشنهادی را بر روی داده‌های بازنمونه‌گیری‌شده ارزیابی کرده و مقایسه‌های جامعی با هفت روش مرجع انجام دادیم که هم معیارهای وابسته به آستانه و هم مستقل از آستانه را در بر می‌گرفت. نتایج نشان می‌دهد که به‌کارگیری تکنیک بازنمونه‌گیری منجر به عملکردی به‌مراتب بهتر معیارهای عملیاتی حیاتی از جمله MTD و MTR می‌شود. علیرغم تعادل ذاتی بین عملکرد تشخیص و سرکوب FA (که با MTFA سنجیده می‌شود)، روش پیشنهادی، با افزایش عملکرد MTFA پایداری را حفظ می‌کند. این یافته‌ها تأیید می‌کنند که مدیریت صحیح عدم توازن برچسب‌ها از طریق تکنیک بازنمونه‌گیری، برای توسعه سیستم‌های CDD مقاوم در کاربردهای SDP حیاتی است. زیرا توزیع‌های نامتوازن داده رواج داشته و در صورت عدم رسیدگی مناسب، می‌توانند تأثیر قابل توجهی بر عملکرد مدل بگذارند. از سوی دیگر، نتایج پژوهش‌های پیشین نشان داده‌اند که الگوریتم تفسیر BD عملکرد بهتری نسبت به الگوریتم LIME در تطبیق مدل‌های SDP دارد، در حالی که مطالعات دیگر عملکرد رضایت‌بخش الگوریتم IME را تأیید کرده‌اند. بر این اساس، مطالعه حاضر به ارزیابی جامع هر دو الگوریتم تفسیر BD و IME پرداخته است. روش CDD پیشنهادی در مقایسه با روش‌های مرجع، هنگام استفاده از هر یک از این الگوریتم‌ها بهبود عملکرد نشان می‌دهد؛ با این حال، رویکرد مبتنی بر IME در تمام معیارهای ارزیابی نتایج به‌مراتب بهتری ارائه می‌کند. این یافته‌ها حاکی از آن است که مکانیزم مبتنی بر تعامل[۳۹] در الگوریتم IME با در نظرگیری وابستگی‌های بین ویژگی‌ها، تفسیرهای دقیق‌تر و پایدارتری تولید می‌کند که منجر به عملکرد برتر در تشخیص تغییرات مفهوم می‌شود. پیشنهادهای زیر برای کارهای آتی ارائه گردیده است:

۱. این آزمایش‌ها را می‌توان در کارهای آینده روی مجموعه داده‌های متوازن یا نامتوازن غیر JIT-SDP تکرار کرد.

۲. برای بهبود نتایج، می‌توانیم داده‌های مرتبط و مناسب برای آموزش یک طبقه‌بندی‌کننده آنلاین را پس از تشخیص CD از سایر مجموعه داده‌های برچسب‌گذاری‌شده استخراج کنیم.

۳. به جای استفاده از تفسیر مدل، که نیاز به برچسب‌گذاری داده‌های آزمایشی را برطرف می‌سازد، می‌توانیم از روش‌های بدون نظارت دیگری که نیازی به برچسب‌گذاری داده‌ها ندارند، مانند تشخیص ناهنجاری، استفاده کنیم.

۴. کارهای اخیر از مدل‌های پیچیده یادگیری عمیق برای SDP استفاده کرده‌اند. در کارهای آینده، این روش‌ها را می‌توان برای JIT-SDP نیز بررسی کرد.

مراجع

[1] X. Chen, Y. Zhao, Q. Wang, and Z. Yuan, "MULTI: Multi-objective effort-aware just-in-time software defect prediction," *Information and Software Technology*, vol. 93, pp. 1-13, 2018, doi: 10.1016/j.infsof.2017.08.004.

[2] G. G. Cabral, L. L. Minku, E. Shihab, and S. Mujahid, "Class imbalance evolution and verification latency in just-in-time software defect prediction," in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, 2019: IEEE, pp. 666-676, doi: 10.1109/ICSE.2019.00076.

[3] A. K. Gangwar, S. Kumar, and A. Mishra, "A Paired Learner-Based Approach for Concept Drift Detection and Adaptation in Software Defect Prediction," *Applied Sciences*, vol. 11, no. 14, p. 6663, 2021, doi: 10.3390/app11146663.

[4] F. Bayram, B. S. Ahmed, and A. Kassler, "From concept drift to model degradation: An overview on performance-aware drift detectors," *Knowledge-Based Systems*, vol. 245, p. 108632, 2022, doi: 10.1016/j.knosys.2022.108632.

[5] F. Hinder, V. Vaquet, and B. Hammer, "One or two things we know about concept drift—a survey on monitoring in evolving environments. Part A: detecting concept drift," *Frontiers in Artificial Intelligence*, vol. 7, p. 1330257, 2024, doi: 10.3389/frai.2024.1330257.

[6] S. Agrahari and A. K. Singh, "Concept drift detection in data stream mining: A literature review," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, pp. 9523-9540, 2022, doi: 10.1016/j.jksuci.2021.11.006.

[7] M. Jain, G. Kaur, and V. Saxena, "A K-Means clustering and SVM based hybrid concept drift detection technique for network anomaly detection," *Expert Systems with Applications*, vol. 193, p. 116510, 2022, doi: 10.1016/j.eswa.2022.116510.

[8] S. McIntosh and Y. Kamei, "Are fix-inducing changes a moving target? a longitudinal case study of just-in-time defect prediction," in *Proceedings of the 40th International Conference on Software Engineering*, 2018, pp. 560-560, doi: 10.1145/3180155.3182514.

[9] D. Lin, C. Tantithamthavorn, and A. E. Hassan, "The impact of data merging on the interpretation of cross-project just-in-time defect models," *IEEE Transactions on Software Engineering*, vol. 48, no. 8, pp. 2969-2986, 2021, doi: 10.1109/TSE.2021.3073920.

[10] G. K. Rajbahadur, S. Wang, G. A. Oliva, Y. Kamei, and A. E. Hassan, "The impact of feature importance methods on the interpretation of defect classifiers," *IEEE Transactions on Software Engineering*, vol. 48, no. 7, pp. 2245-2261, 2021, doi: 10.1109/TSE.2021.3056941.

[11] W. Zheng, T. Shen, X. Chen, and P. Deng, "Interpretability application of the Just-in-Time software defect prediction model," *Journal of Systems and Software*, vol. 188, p. 111245, 2022, doi: 10.1016/j.jss.2022.111245.

[12] K. Fathi, M. Sadurski, T. Kleinert, and H. W. van de Venn, "Source component shift detection & classification for improved remaining useful life estimation in alarm-based predictive maintenance," in *2023 23rd international conference on control, automation and systems (iccas)*, 2023: IEEE, pp. 975-980, doi: 10.23919/ICCAS59377.2023.10316874.

[13] B. Turhan, "On the dataset shift problem in software engineering prediction models," *Empirical Software Engineering*, vol. 17, pp. 62-74, 2012, doi: 10.1007/s10664-011-9182-8.

[14] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE transactions on knowledge and data engineering*, vol. 31, no. 12, pp. 2346-2363, 2018, doi: 10.1109/TKDE.2018.2876857.

[15] F. Dong, J. Lu, K. Li, and G. Zhang, "Concept drift region identification via competence-based discrepancy distribution estimation," in *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, 2017: IEEE, pp. 1-7, doi: 10.1109/ISKE.2017.8258734.

[16] R. F. Haase and M. V. Ellis, "Multivariate analysis of variance," *Journal of Counseling Psychology*, vol. 34, no. 4, p. 404, 1987.

[17] J. Gama, *Knowledge discovery from data streams*. CRC Press, 2010.

[18] O. A. Mahdi, E. Pardede, N. Ali, and J. Cao, "Fast reaction to sudden concept drift in the absence of class labels," *Applied Sciences*, vol. 10, no. 2, p. 606, 2020, doi: 10.3390/app10020606.

[19] J. Jiarpakdee, C. K. Tantithamthavorn, H. K. Dam, and J. Grundy, "An empirical study of model-agnostic techniques for defect prediction models," *IEEE Transactions on Software Engineering*, vol. 48, no. 1, pp. 166-185, 2020, doi: 10.1109/TSE.2020.2982385.

[20] E. Štrumbelj, I. Kononenko, and M. R. Šikonja, "Explaining instance classifications with interactions of subsets of feature values," *Data & Knowledge Engineering*, vol. 68, no. 10, pp. 886-904, 2009, doi: 10.1016/j.datak.2009.01.004.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin, "" Why should i trust you?" Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135-1144, doi: 10.1145/2939672.2939778.

[22] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[23] D. Vreš and M. Robnik-Šikonja, "Preventing deception with explanation methods using focused sampling," *Data Mining and Knowledge Discovery*, vol. 38, no. 5, pp. 3262-3307, 2024, doi: 10.1007/s10618-022-00900-w.

[24] Y. Kamei, T. Fukushima, S. McIntosh, K. Yamashita, N. Ubayashi, and A. E. Hassan, "Studying just-in-time defect prediction using cross-project models," *Empirical Software Engineering*, vol. 21, pp. 2072-2106, 2016, doi: 10.1007/s10664-015-9400-x.

[25] W. Li, W. Zhang, X. Jia, and Z. Huang, "Effort-aware semi-supervised just-in-time defect prediction," *Information and Software Technology*, vol. 126, p. 106364, 2020, doi: 10.1016/j.infsof.2020.106364.

[26] G. J. Ross, N. M. Adams, D. K. Tasoulis, and D. J. Hand, "Exponentially weighted moving average charts for detecting concept drift," *Pattern recognition letters*, vol. 33, no. 2, pp. 191-198, 2012.

[27] C. Tantithamthavorn, A. E. Hassan, and K. Matsumoto, "The impact of class rebalancing techniques on the performance and interpretation of defect prediction models," *IEEE Transactions on Software Engineering*, vol. 46, no. 11, pp. 1200-1219, 2020, doi: 10.1109/TSE.2018.2876537.

[28] L. Pelayo and S. Dick, "Applying novel resampling strategies to software defect prediction," in *NAFIPS 2007-2007 Annual meeting of the North American fuzzy information processing society*, 2007: IEEE, pp. 69-72, doi: 10.1109/NAFIPS.2007.383813.

[29] L. Chen, B. Fang, Z. Shang, and Y. Tang, "Tackling class overlap and imbalance problems in software defect prediction," *Software Quality Journal*, vol. 26, pp. 97-125, 2018, doi: 10.1007/s11219-016-9342-6.

[30] O. I. Sheluhin and S. A. Sekretarev, "Concept drift detection in streaming classification of mobile application traffic," *Automatic Control and Computer Sciences*, vol. 55, pp. 253-262, 2021, doi: 10.1016/j.procs.2017.11.440.

[31] S. Tabassum, L. L. Minku, and D. Feng, "Cross-project online just-in-time software defect prediction," *IEEE Transactions on Software Engineering*, vol. 49, no. 1, pp. 268-287, 2022, doi: 10.1109/TSE.2022.3150153.

[32] Y. Gao, Y. Zhu, and Y. Zhao, "Dealing with imbalanced data for interpretable defect prediction," *Information and software technology*, vol. 151, p. 107016, 2022, doi: 10.1016/j.infsof.2022.107016.

[33] A. Mockus and D. M. Weiss, "Predicting risk of software changes," *Bell Labs Technical Journal*, vol. 5, no. 2, pp. 169-180, 2000, doi: 10.1002/bltj.2229.

[34] A. E. Hassan, "Predicting faults using the complexity of code changes," in *2009 IEEE 31st international conference on software engineering*, 2009: IEEE, pp. 78-88, doi: 10.1109/ICSE.2009.5070510.

[35] R. Purushothaman and D. E. Perry, "Toward understanding the rhetoric of small source code changes," *IEEE Transactions on Software Engineering*, vol. 31, no. 6, pp. 511-526, 2005, doi: 10.1109/TSE.2005.74.

[36] P. J. Guo, T. Zimmermann, N. Nagappan, and B. Murphy, "Characterizing and predicting which bugs get fixed: an empirical study of microsoft windows," in *Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering-Volume 1*, 2010, pp. 495-504, doi: 10.1145/1806799.1806871.

[37] L. Torgo and M. L. Torgo, "Package 'dmwr'," *Comprehensive R archive network*, 2013.

[38] A. L. Suárez-Cetrulo, D. Quintana, and A. Cervantes, "A survey on machine learning for recurring concept drifting data streams," *Expert Systems with Applications*, vol. 213, p. 118934, 2023, 10.1016/j.eswa.2022.118934.

[39] A. Bifet, J. Read, B. Pfahringer, G. Holmes, and I. Žliobaitė, "CD-MOA: Change detection framework for massive online analysis," in *International Symposium on Intelligent Data Analysis*, 2013: Springer, pp. 92-103, doi: 10.1007/978-3-642-41398-8_9.

[40] J. Demšar and Z. Bosnić, "Detecting concept drift in data streams using model explanation," *Expert Systems with Applications*, vol. 92, pp. 546-559, 2018, doi: 10.1016/j.eswa.2017.10.003.

[41] W. J. Conover and R. L. Iman, "Rank transformations as a bridge between parametric and nonparametric statistics," *The American Statistician*, vol. 35, no. 3, pp. 124-129, 1981.

[42] R. C. Blair and J. J. Higgins, "The power of t and Wilcoxon statistics: A comparison," *Evaluation Review*, vol. 4, no. 5, pp. 645-656, 1980, doi: 10.1177/0193841X8000400506.

پاورقی‌ها:

[1] Just-In-Time Software Defect Prediction
[2] Concept Drift
[3] Concept Drift Detection
[4] Error Rate
[5] Offline
[6] Source Component Shift
[7] Interpretation Inconsistency Detection in JIT-SDP
[8] Resampled
[9] Multivariate ANalysis Of VAriance
[10] Area Under the Curve
[11] Matthews Correlation Coefficient
[12] Benchmark
[13] Commit
[14] Interaction-based Method for Explanation
[15] Local Interpretable Model-agnostic Explanations
[16] SHapley Additive exPlanations
[17] Over-sampling
[18] Under-sampling
[19] Synthetic Minority Over-sampling Technique
[20] Overfitting
[21] Page Hinkley test
[22] Random Forest
[23] Verification Latency
[24] Baseline method
[25] Induced defects
[26] Number of modified Subsystems
[27] Number of modified Directories
[28] Number of modified Files
[29] Lines Added
[30] Lines Deleted
[31] Lines Total
[32] Model-agnostic
[33] Mean Time to Detection
[34] Mean Time between False Alarms
[35] False Alarm
[36] Mean Time Ratio
[37] Friedman
[38] Nemenyi
[39] Interaction-based