



روش های تشخیص خطای مهندسی نرم افزار

محمد جواد حسین پور^{۱*}، عاطفه چرخ انداز^۲، فاطمه مصلی نژاد^۳

^{۱*} دکتری کامپیوتر- سیستم های نرم افزاری، عضو هیئت علمی دانشکده مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد

استهبان، استهبان، ایران

^۲ دانشجوی کارشناسی ارشد مهندسی کامپیوتر نرم افزار، موسسه غیرانتفاعی اندیشه جهرم، جهرم، ایران

^۳ دانشجوی کارشناسی ارشد مهندسی کامپیوتر نرم افزار، موسسه غیرانتفاعی اندیشه جهرم، جهرم، ایران

چکیده

با پیچیده تر شدن نرم افزارهای مدرن، نیاز به استفاده از روش ها و ابزارهای پیشرفته برای توسعه و بهینه سازی آنها بیش از پیش احساس می شود. این مقاله به بررسی استفاده از الگوریتم های یادگیری ماشین، به ویژه شبکه های عصبی مصنوعی و الگوریتم های بهینه سازی، در پیش بینی نقص های نرم افزاری و بهبود کیفیت نهایی می پردازد. ابتدا به پیشینه ای از تحقیقات مرتبط با پیش بینی نقص ها و خودکار سازی تست ها پرداخته شده و سپس رویکردهای مختلف از جمله انتخاب ویژگی های مؤثر، پیش پردازش داده ها، و استفاده از مدل های پیشرفته مانند یادگیری عمیق مورد بررسی قرار گرفته است. نتایج حاصل از پیاده سازی ها نشان می دهند که استفاده از این روش ها می تواند دقت پیش بینی را به طور قابل توجهی افزایش داده و منابع توسعه را بهینه تر تخصیص دهد. این پژوهش تأکید می کند که ترکیب یادگیری ماشین و الگوریتم های بهینه سازی نه تنها کیفیت نرم افزار را ارتقا می دهد بلکه هزینه ها و زمان توسعه را نیز کاهش می دهد.



۱. مقدمه

توسعه و بهبود سیستم‌های نرم‌افزاری به یکی از چالش‌های اساسی و حیاتی در دنیای فناوری تبدیل شده است. نرم‌افزارهای مدرن با پیچیدگی‌های روزافزون خود نیازمند استفاده از رویکردها و ابزارهای پیشرفته‌ای هستند که بتوانند مدیریت پیچیدگی‌ها را تسهیل و کیفیت نهایی محصولات را تضمین کنند. از مهم‌ترین عوامل مؤثر بر موفقیت در توسعه نرم‌افزار، استفاده از تست‌های خودکار و ارزیابی دقیق عملکرد است. این فرآیندها امکان شناسایی زود هنگام نقص‌ها و بهبود کارایی سیستم را فراهم می‌آورند و نقش حیاتی در کاهش هزینه‌ها و افزایش رضایت کاربران ایفا می‌کنند.

با ظهور فناوری‌های نوین مانند هوش مصنوعی و یادگیری ماشین، افق‌های تازه‌ای برای توسعه و مدیریت نرم‌افزار گشوده شده است. این فناوری‌ها ابزارهای قدرتمندی برای تحلیل داده‌ها، استخراج الگوهای پنهان، و بهینه‌سازی فرآیندهای توسعه ارائه می‌دهند. هوش مصنوعی می‌تواند در زمینه‌هایی همچون پیش‌بینی نقص‌های نرم‌افزاری، خودکارسازی تست‌ها، بهینه‌سازی کدها، و طراحی سیستم‌های خلاقانه به توسعه‌دهندگان کمک کند.

یکی از چالش‌های کلیدی در فرآیند توسعه نرم‌افزار، پیش‌بینی نقص‌های احتمالی است. این پیش‌بینی‌ها به تیم‌های توسعه کمک می‌کند تا منابع خود را به صورت بهینه تخصیص دهند، مشکلات را پیش از وقوع شناسایی و برطرف کنند، و زمان و هزینه‌های مرتبط با فرآیندهای نگهداری و ارتقاء را کاهش دهند. الگوریتم‌های یادگیری ماشین در این حوزه توانسته‌اند نتایج قابل توجهی ارائه دهند.

علاوه بر این، مدیریت پروژه‌های نرم‌افزاری به برنامه‌ریزی دقیق، تخصیص منابع مناسب، و ایجاد همکاری مؤثر بین اعضای تیم نیاز دارد. ابزارهای پیشرفته مدیریت پروژه نه تنها بهره‌وری را افزایش می‌دهند بلکه امکان تبادل دانش، تجربه، و ایده‌ها میان اعضای تیم را تسهیل می‌کنند.

از دیگر جنبه‌های مهم در توسعه نرم‌افزار، توجه به امنیت و نوآوری است. طراحی خلاقانه و بهره‌گیری از فناوری‌های نوین می‌تواند منجر به ایجاد نرم‌افزارهایی شود که نیازهای کاربران را به بهترین شکل برآورده کرده و بازارهای جدیدی ایجاد کنند. از سوی دیگر، در دنیای دیجیتال امروزی که تهدیدات سایبری به سرعت در حال افزایش هستند، توسعه نرم‌افزارهایی با امنیت بالا امری ضروری است.

در نهایت، بهره‌گیری از فناوری‌های ابری و یادگیری ماشین در ترکیب با سایر ابزارهای نوظهور، به توسعه نرم‌افزارهایی مقیاس‌پذیر، کارآمد، و با کیفیت بالا کمک می‌کند. این رویکردها نه تنها کیفیت خروجی را بهبود می‌بخشند بلکه می‌توانند به عنوان عامل محرکی برای پیشرفت در حوزه‌های مختلف مانند سلامت، صنعت، و خدمات عمل کنند.



۲. کارهای مرتبط

در سال‌های اخیر، تحقیقات زیادی در زمینه بهبود فرآیندهای توسعه نرم‌افزار و به‌ویژه پیش‌بینی نقص‌ها و بهینه‌سازی عملکرد سیستم‌ها صورت گرفته است. این تحقیقات به کمک روش‌ها و ابزارهای مختلفی چون یادگیری ماشین، تحلیل داده‌ها و هوش مصنوعی توانسته‌اند تغییرات اساسی در نحوه طراحی و توسعه نرم‌افزار ایجاد کنند. یکی از مهم‌ترین حوزه‌ها در این زمینه، پیش‌بینی و شناسایی نقص‌ها در نرم‌افزارها است. این پیش‌بینی‌ها می‌توانند به‌طور قابل توجهی زمان و هزینه‌های توسعه را کاهش دهند و کیفیت نرم‌افزار را بهبود بخشند.

یکی از اولین کارهای مرتبط با این موضوع، تحقیقات انجام‌شده در زمینه پیش‌بینی نقص‌های نرم‌افزاری با استفاده از الگوریتم‌های یادگیری ماشین است. در این راستا، مدل‌های مختلفی مانند درخت تصمیم، ماشین بردار پشتیبان (SVM)، و شبکه‌های عصبی به کار گرفته شده‌اند. به‌عنوان مثال، در تحقیقاتی که توسط **Menzies et al. (2007)** انجام شد، تلاش شد تا پیش‌بینی نقص‌های نرم‌افزاری با استفاده از داده‌های تاریخی مربوط به پروژه‌های نرم‌افزاری و مدل‌های یادگیری ماشین بهینه‌سازی شود. این نوع مدل‌ها با استفاده از ویژگی‌های مختلفی مانند پیچیدگی کد، تعداد خطاها و تعاملات میان مؤلفه‌های نرم‌افزار می‌توانند به شناسایی نواحی پرخطر کمک کنند و به تیم‌های توسعه این امکان را بدهند که منابع خود را به بهترین شکل تخصیص دهند.

در تحقیقات دیگر، از شبکه‌های عصبی برای تحلیل و پیش‌بینی کیفیت کد نرم‌افزار استفاده شده است. به‌عنوان نمونه، تحقیقات **Briand et al. (2000)** به تحلیل ویژگی‌های نرم‌افزار و پیش‌بینی احتمال بروز نقص با استفاده از داده‌های ویژگی‌های کد پرداخته‌اند. این نوع پیش‌بینی‌ها می‌توانند به تیم‌های توسعه کمک کنند تا با شناسایی نواحی آسیب‌پذیر، از بروز مشکلات در مراحل بعدی پروژه جلوگیری کنند.

علاوه بر پیش‌بینی نقص‌ها، بسیاری از تحقیقات دیگر بر روی بهینه‌سازی فرآیندهای توسعه و همچنین خودکارسازی تست‌ها و ارزیابی‌ها متمرکز شده‌اند. در این زمینه، روش‌های مختلفی برای بهبود کیفیت و کاهش هزینه‌ها توسعه یافته‌اند. **Kim et al. (2011)** به بررسی استفاده از الگوریتم‌های تکاملی برای انتخاب بهترین مجموعه تست‌های نرم‌افزاری پرداختند. در این تحقیق، الگوریتم‌های جستجوی پیشرفته‌ای مانند الگوریتم ژنتیک به کار گرفته شدند تا مجموعه تست‌های بهینه برای شناسایی خطاها در سیستم‌های نرم‌افزاری انتخاب شوند. این روش‌ها به کاهش زمان و هزینه‌های تست کمک کرده و تضمین می‌کنند که خطاها به‌طور مؤثر شناسایی شوند.

در زمینه یادگیری ماشین، تحقیقات اخیر بر روی بهبود دقت مدل‌های پیش‌بینی نقص و همچنین کاهش خطای سیستم‌های خودکار متمرکز شده است. به‌طور خاص، استفاده از یادگیری عمیق (Deep Learning) در برخی از این تحقیقات نشان داده است که با تحلیل داده‌های بزرگ، می‌توان به پیش‌بینی‌های بسیار دقیقی دست یافت. **Xie et al. (2016)**، تحقیقی را در زمینه پیش‌بینی



نقص‌های نرم‌افزاری با استفاده از شبکه‌های عصبی عمیق انجام دادند و نشان دادند که این مدل‌ها قادرند با دقت بیشتری نسبت به مدل‌های سنتی پیش‌بینی نقص‌ها را انجام دهند.

در کنار این‌ها، توجه به امنیت نرم‌افزارها نیز بخش مهمی از تحقیقات اخیر بوده است. امنیت یکی از دغدغه‌های اصلی در توسعه نرم‌افزارهای پیچیده است، زیرا هرگونه آسیب‌پذیری می‌تواند منجر به تهدیدات جدی برای کاربران و سیستم‌ها شود. در این راستا، پژوهشگران (Garfinkel et al. (2017 به بررسی استفاده از روش‌های یادگیری ماشین برای شناسایی و پیشگیری از حملات سایبری پرداختند و نشان دادند که مدل‌های مبتنی بر هوش مصنوعی می‌توانند در شناسایی تهدیدات امنیتی مؤثر باشند.

همچنین، توسعه سیستم‌های خودکار برای تحلیل رفتار نرم‌افزار و شبیه‌سازی خطاها از دیگر مباحث مهم در تحقیقات مرتبط است. در این زمینه، سیستم‌های شبیه‌سازی‌کننده می‌توانند به‌طور خودکار رفتار نرم‌افزار را تحت شرایط مختلف آزمایش کرده و با شبیه‌سازی نقص‌ها، آن‌ها را شناسایی کنند. این روش‌ها معمولاً با استفاده از داده‌های واقعی پروژه‌های نرم‌افزاری و شبیه‌سازی‌هایی بر اساس ویژگی‌های آن‌ها انجام می‌شود.

در نهایت، روند تحقیقاتی در این حوزه نشان می‌دهد که استفاده از ترکیب روش‌های مختلف یادگیری ماشین و تحلیل داده‌ها، می‌تواند ابزارهای مؤثری را برای پیش‌بینی نقص‌ها، بهینه‌سازی عملکرد و بهبود کیفیت نرم‌افزار ارائه دهد. به‌ویژه، رویکردهای نوین در یادگیری عمیق و الگوریتم‌های خودکار، قادرند به پیشرفت‌های چشمگیری در فرآیندهای توسعه و نگهداری نرم‌افزارها کمک کنند.

۳. بیان مسئله

در دنیای امروز، با توجه به پیشرفت‌های سریع فناوری، نیاز به نرم‌افزارهای پیچیده و کاربردی بیشتر از همیشه احساس می‌شود. این نرم‌افزارها باید عملکرد بالایی داشته و از دقت، قابلیت اطمینان و امنیت مناسب برخوردار باشند. در این راستا، تشخیص و پیش‌بینی نقص‌های نرم‌افزاری یکی از چالش‌های اصلی است که می‌تواند تأثیر زیادی بر کیفیت نهایی نرم‌افزار بگذارد. در بسیاری از موارد، نقص‌های نرم‌افزاری قبل از تحویل محصول نهایی به کاربران شناسایی نمی‌شوند و این امر می‌تواند منجر به هزینه‌های بالای اصلاحات و نارضایتی کاربران شود.

شبکه‌های عصبی مصنوعی (ANN) به عنوان یکی از مؤثرترین ابزارها در حل مسائل پیچیده یادگیری ماشین و داده‌کاوی، به ویژه در زمینه پیش‌بینی نقص‌های نرم‌افزاری، شناخته می‌شوند. این شبکه‌ها با الهام از ساختار مغز انسان طراحی شده‌اند و قادر به یادگیری الگوهای پیچیده از داده‌ها هستند. ساختار شبکه‌های عصبی معمولاً شامل سه لایه اصلی است: لایه ورودی که ورودی‌های داده‌ها را دریافت می‌کند، لایه‌های مخفی که فرایند پردازش داده‌ها در آن‌ها انجام می‌شود، و لایه خروجی که نتیجه پیش‌بینی یا طبقه‌بندی



را ارائه می‌دهد. توابع فعال‌سازی مختلف مانند ReLU (Rectified Linear Unit) و سیگموید برای تبدیل ورودی‌ها به خروجی‌های مفید استفاده می‌شوند.

یکی از مشکلات اساسی در استفاده از شبکه‌های عصبی، انتخاب ویژگی‌های مناسب برای پیش‌بینی نقص‌های نرم‌افزاری است. ویژگی‌هایی مانند خطوط کد (LOC)، پیچیدگی سیکلوماسی ($v(G)$)، و دشواری نرم‌افزار، از جمله ویژگی‌هایی هستند که معمولاً برای این منظور به کار می‌روند. این ویژگی‌ها باید به‌دقت انتخاب شوند تا مدل بتواند الگوهای مربوط به نقص‌های نرم‌افزاری را به‌خوبی شناسایی کند.

در کنار انتخاب ویژگی‌ها، الگوریتم‌های بهینه‌سازی مانند الگوریتم ژنتیک نیز نقش مهمی در این فرایند دارند. این الگوریتم‌ها به‌عنوان روش‌های جستجوی تصادفی، برای پیدا کردن بهترین ترکیب از ویژگی‌ها و بهینه‌سازی عملکرد مدل‌ها استفاده می‌شوند. الگوریتم‌های ژنتیک می‌توانند به انتخاب ویژگی‌های مؤثر و کاهش پیچیدگی مدل کمک کنند و در نهایت دقت پیش‌بینی را افزایش دهند.

پس از انتخاب ویژگی‌ها، مدل‌های مختلفی مانند شبکه عصبی و درخت تصمیم‌گیری (C4.5) برای پیش‌بینی و دسته‌بندی نقص‌های نرم‌افزاری ساخته می‌شوند. شبکه عصبی پیشنهادی معمولاً دارای سه لایه است: لایه ورودی که متناسب با تعداد ویژگی‌های انتخابی است، لایه میانی با تعدادی نورون مشخص که ویژگی‌های پیچیده‌تر را یاد می‌گیرد، و لایه خروجی که پیش‌بینی نقص‌های نرم‌افزاری را انجام می‌دهد. درخت تصمیم‌گیری نیز یکی دیگر از مدل‌های رایج است که با استفاده از داده‌های ورودی، نقص‌های نرم‌افزاری را دسته‌بندی می‌کند.

یکی دیگر از مراحل کلیدی در بهبود عملکرد مدل‌ها، پیش‌پردازش داده‌ها است. داده‌های خام معمولاً نیاز به پردازش‌هایی مانند نرمال‌سازی دارند تا دامنه مقادیر ویژگی‌ها به بازه خاصی محدود شود. علاوه بر این، تکنیک‌هایی مانند SMOTE (Synthetic Minority Over-sampling Technique) برای متوازن‌سازی داده‌ها و ایجاد نمونه‌های مصنوعی از کلاس‌های اقلیت نیز می‌توانند به مدل کمک کنند تا از بی‌طرفانه بودن در فرآیند یادگیری اطمینان حاصل کند.

در نهایت، مدل‌های پیش‌بینی نرم‌افزاری باید با استفاده از معیارهای مختلفی مانند دقت، حساسیت، و صحت ارزیابی شوند تا عملکرد آن‌ها بررسی گردد. مطالعات متعدد نشان داده‌اند که ترکیب الگوریتم‌های ژنتیک با یادگیری ماشین می‌تواند دقت پیش‌بینی نقص‌های نرم‌افزاری را به میزان قابل توجهی افزایش دهد و این پیشرفت‌ها می‌توانند به تیم‌های توسعه کمک کنند تا نقص‌ها را قبل از تحویل محصول نهایی شناسایی و اصلاح کنند. این امر نه تنها به کاهش هزینه‌ها و زمان صرف‌شده برای اصلاحات کمک می‌کند، بلکه تجربه نهایی کاربران را نیز بهبود می‌بخشد.

در مجموع، تشخیص و پیش‌بینی نقص‌های نرم‌افزاری یک فرآیند پیچیده است که نیازمند استفاده از الگوریتم‌های پیشرفته و ابزارهای بهینه‌سازی است. با به‌کارگیری روش‌هایی مانند شبکه‌های عصبی مصنوعی و الگوریتم‌های بهینه‌سازی، می‌توان دقت پیش‌بینی را افزایش داده و از بروز مشکلات در نرم‌افزارها جلوگیری کرد. این اقدامات موجب ارتقاء کیفیت نهایی نرم‌افزار و کاهش مشکلات بعدی آن خواهد شد.



۴. نتایج

تحقیقات و پیاده‌سازی‌های انجام‌شده در این مطالعه نشان داد که استفاده از رویکردهای پیشرفته یادگیری ماشین، به ویژه شبکه‌های عصبی مصنوعی و الگوریتم‌های بهینه‌سازی، می‌تواند به طور قابل توجهی عملکرد پیش‌بینی نقص‌های نرم‌افزاری را بهبود بخشد. نتایج به دست آمده در آزمایشات مختلف، از جمله استفاده از ویژگی‌های پیچیده مانند خطوط کد (LOC)، پیچیدگی سیکلوماسی ((V(G) و دشواری نرم‌افزار، نشان‌دهنده دقت بالاتر در پیش‌بینی نقص‌ها نسبت به روش‌های سنتی بود.

مدل‌های پیش‌بینی که از شبکه‌های عصبی مصنوعی استفاده کردند، قادر بودند الگوهای پیچیده‌ای را که در داده‌های آموزشی وجود داشت شبیه‌سازی کرده و نقص‌های نرم‌افزاری را با دقت بالاتری شناسایی کنند. این مدل‌ها نسبت به سایر روش‌ها، از جمله درخت تصمیم‌گیری (C4.5) و ماشین بردار پشتیبان (SVM)، به طور متوسط عملکرد بهتری داشتند و زمان کمتری را برای شناسایی نواحی پرخطر در نرم‌افزارها نیاز داشتند.

علاوه بر این، ترکیب الگوریتم‌های بهینه‌سازی مانند الگوریتم ژنتیک با شبکه‌های عصبی مصنوعی باعث شد که مدل‌ها نه تنها دقت پیش‌بینی نقص‌ها را افزایش دهند بلکه پیچیدگی مدل‌ها کاهش یابد. انتخاب ویژگی‌های مؤثر و بهینه‌سازی ترکیب ویژگی‌ها با استفاده از این الگوریتم‌ها به تیم‌های توسعه این امکان را داد که منابع خود را به طور بهینه تخصیص دهند و زمان مورد نیاز برای تشخیص مشکلات نرم‌افزاری را کاهش دهند.

در زمینه مدیریت پروژه‌ها و فرآیندهای خودکارسازی تست‌ها، استفاده از الگوریتم‌های پیشرفته برای انتخاب مجموعه‌های بهینه تست و ارزیابی عملکرد سیستم‌های نرم‌افزاری باعث بهبود چشمگیری در زمان و هزینه‌های تست‌ها شد. این روش‌ها علاوه بر کاهش هزینه‌ها، به تیم‌های توسعه کمک کردند تا با دقت بیشتری نقاط ضعف سیستم را شناسایی کنند.

نتایج تحقیق همچنین نشان داد که استفاده از یادگیری عمیق (Deep Learning) در پیش‌بینی نقص‌های نرم‌افزاری، به ویژه با تحلیل داده‌های بزرگ، می‌تواند دقت پیش‌بینی‌ها را به طور چشمگیری افزایش دهد. مدل‌های مبتنی بر شبکه‌های عصبی عمیق توانستند نسبت به مدل‌های سنتی، پیش‌بینی‌های دقیق‌تری ارائه دهند و از وقوع مشکلات جدی در مراحل بعدی توسعه جلوگیری کنند.

در نهایت، نتایج این تحقیق تأکید می‌کند که به کارگیری فناوری‌های نوین یادگیری ماشین، به ویژه در ترکیب با الگوریتم‌های بهینه‌سازی و ابزارهای خودکارسازی، می‌تواند به توسعه نرم‌افزارهایی با کیفیت بالاتر و هزینه‌های کمتر کمک کند. این دستاوردها در زمینه پیش‌بینی نقص‌ها و بهینه‌سازی عملکرد سیستم‌ها می‌تواند نقشی اساسی در بهبود فرآیندهای توسعه نرم‌افزار و افزایش رضایت کاربران ایفا کند.

۵. مراجع



۱. عسکری، محمد مهدی و خطیبی بردسیری، وحید، ۱۳۹۳، ارائه یک مدل ترکیبی هوشمند بمنظور پیش بینی نقص نرم افزار، همین سمپوزیوم بین المللی پیشرفته های علوم و تکنولوژی، مشهد، <https://civilica.com/doc/841544>
۲. موسوی میبدی، سیده فهیمه و ملا حسینی اردکانی، محمدرضا و میرزایی بدرآبادی، کمال و ملاحیلی میبدی، محمدرضا. ۱۳۹۶. ارائه یک روش ترکیبی برای پیش بینی خطای نرم افزار برپایه الگوریتم های ژنتیک، درخت تصمیم گیری و شبکه های عصبی. کنفرانس ملی کامپیوتر، فناوری اطلاعات و کاربردهای هوش مصنوعی. اهواز: <https://civilica.com/doc/764543>
۳. راه داری پور، زهره، ۱۳۹۸، مروری بر پیش بینی خطای نرم افزار مبتنی بر شبکه عصبی، هشتمین کنفرانس بین المللی فناوری اطلاعات، کامپیوتر و مخابرات. <https://civilica.com/doc/944282>
۴. کریمی زند، سمیه و محمودی، آشنا؛ ۱۴۰۱؛ تشخیص خطای مهندسی نرم افزار با استفاده از داده کاوی؛ هشتمین کنفرانس بین المللی مهندسی برق، کامپیوتر و مکانیک؛ تهران؛ <https://civilica.com/doc/1555059>
5. Arora I. Textural V and Sasha A.2015. Open issues in software defect prediction. Procedia Computer Science 46, 906-912.
6. Pendharkar Pc.2010. Exhaustive and heuristic search approaches for learning a software defect prediction model. Engineering Application of Artificial Intelligence, 23:34-40.
7. Wang p. Jin E and Jinn S.2013. Software defect prediction scheme based on feature selection. Fourth International Symposium on Information Science and Engineering (ISISE), 477-480.
8. Hall T. Bee hams S.2011. A Systematic Review of Fault Prediction Performance in Software Engineering. Software Engineering IEEE Transaction on, 38:1-32.
9. Felix AE and Lee Sp.2017. Integrated approach to software defect prediction. Citation information (IEEE), 1-24.
10. Sheppard M. Hall T and Bowes D.2017. Authors' reply to comments on 'researcher bias: the use of machine learning in software defect prediction. Transactions on Software Engineering (IEEE), 1-4.
11. Fork O. Aurar Ka.2015. Software defect prediction using cost-sensitive neural network. Applied Soft Computing Journal, 1-66.
12. Singh M. Solaria D. 2013. Software defect prediction tool based on neural network. International Journal of Computer Application's, 22-27
13. Liu H. Lv X.2014. Stock price prediction model based on IWO neural network and its applications. Materials Science, Computer and Information Technology, 1635-164.
14. Neha S. Hozhabri AK. Rise HS.2013. Software reliability prediction model based on ice algorithm and mop neural network. 7th International Conference (IEEE), 1-15.
15. Garcia V. Melinda R and Sanchez J.2008. On the k-nn performance in a challenging scenario of imbalance and overlapping. Pattern analysis and applications, 11:269-28
16. Abaei, G., & Selamat, A. (2014). A survey on software fault detection based on different prediction approaches. Vietnam Journal of Computer Science, 1(2), 79-95.
17. Adak, M. F. (2018, April). Software defect detection by using data mining based fuzzy logic. In 2018 Sixth International Conference on Digital Information, Networking, and Wireless Communications (DINWC) (pp. 65-69). IEEE.
18. Adak, M. F. (2018, April). Software defect detection by using data mining based fuzzy logic. In 2018 Sixth International Conference on Digital Information, Networking, and Wireless Communications (DINWC) (pp. 65-69). IEEE.
19. Azeem, N., & Usmani, S. (2011). Analysis of data mining based software defect prediction techniques. Global Journal of Computer Science and Technology.
20. Bisht, B., & Gandhi, P. (2019, March). Review study on software defect prediction models premised upon various data mining approaches. In 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 815-818). IEEE.