ارائه یک رویکرد ژنتیک–فازی نوع ۲ برای ارزیابی کیفیت سیستم‌های خود انطباق با استفاده از شاخص‌های کیفی نرم افزار

مجید عبدالرزاق نژاد* عشرت زرگری .....مهدی خرد*****

* دانشیار، گروه علوم کامپیوتر، دانشکده مهندسی کامپیوتر و صنایع - دانشگاه صنعتی بیرجند - بیرجند - ایران

** کارشناسی ارشد، گروه مهندسی کامپیوتر، دانشگاه آزاد اسلامی – واحد بیرجند - ایران

*** دانشجوی دکتری، گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی - دانشگاه قم - قم - ایران

تاریخ دریافت: ۰۰/۰۰/۰۰ تاریخ پذیرش: ۰۰/۰۰/۰۰

نوع مقاله: پژوهشی

چکیده

امروزه با گسترش علم در حوزه مهندسی کامپیوتر، امکان توسعه سیستم‌های هوشمند و سازگار با محیط نیز افزایش یافته است. سیستم‌های خود انطباق یکی از انواع سیستم‌های نرم‌افزاری هستند که رفتار خود را با توجه به شرایط محیطی تغییر می‌دهند و خود را با آن سازگار می‌کنند. اگرچه ارزیابی عملکرد این سیستم‌ها در بیشتر تحقیقات تحت پوشش قرار گرفته شده است، ولی ارزیابی کیفیت آنها مغفول مانده است. اولین چالش در این مسیر، عدم قطعیت شاخص‌های کیفی این سیستم‌ها است که پارامترهایی ثابت، مشخص و قطعی ندارند. به عنوان مثال یکی از شاخص‌های کیفی سیستم خودانطباق، توانایی اجرای نرم‌افزار در سیستم‌عامل‌های مختلف است. این شاخص، از نظر افراد خبره مختلف، می‌تواند درجه اهمیت مختلفی داشته باشد. چالش دیگر، برخی شاخص‌های کیفی متغیر ریاضی نیستند و یک متغیر زبانی بین کاربران و کارشناسان می‌باشند، که این امر نشان می‌دهد این کمیت‌های کیفی، متغیرهای فازی بوده و با منطق فازی کاملاً قابل فرموله شدن می‌باشند. در این مقاله به منظور ارزیابی کیفیت این سیستم‌ها براساس شاخص‌های کیفی نرم افزار اقدام به ارائه یک رویکرد ژنتیک-فازی نوع دوم شده است. جهت توصیف شاخص‌های کیفی از منطق فازی نوع دوم و برای تعیین بهینه وزن‌های فازی شاخص‌های کیفی از الگوریتم ژنتیک استفاده شده است. در روش پیشنهادی سعی شده تا سیستم‌های خودانطباق از دو بعد شامل بعد نرم‌افزاری این سیستم‌ها و بعد خود انطباق آنها مورد مقایسه قرار گیرند. این در حالی است که اکثر تحقیقات موجود تنها به یک بعد می‌پردازند. جهت ارزیابی روش پیشنهادی از یک سیستم کنترل ترافیک با نام **InSync** که یک سیستم خودانطباق کنترل ترافیک (**ATCS**) بوده و حاوی شاخص‌های متعدد کیفی می‌باشد استفاده شده است. نتایج بدست آمده باتوجه به عدم نیاز به تولید سناریو، توانایی بسط روش به تمامی شاخص‌های کیفی نرم افزار و سادگی آن تایید کننده کارآمدی روش پیشنهادی می‌باشند. همچنین جامعیت بیشتر روش پیشنهادی را نسبت به سایر مدل‌های ارزیابی موجود نشان می‌دهد.

واژگان کلیدی: سیستم خود انطباق، شاخص‌های کیفی نرم افزار، الگوریتم ژنتیک، فازی نوع دو، سیستم خودانطباق کنترل ترافیک.

نویسنده مسئول: مجید عبدالرزاق نژاد - abdolrazzagh@birjandut.ac.ir

۱. مقدمه

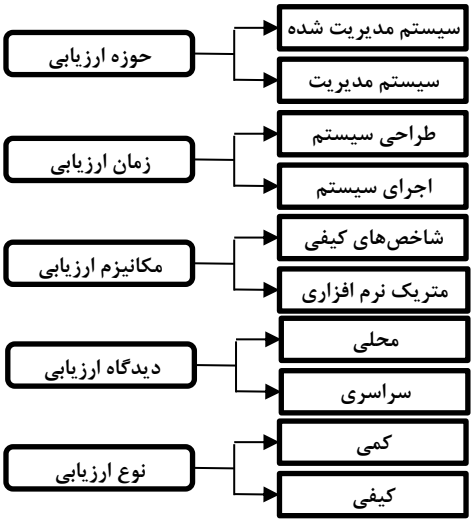
بسیاری از تکنیک‌ها و راه‌حل‌های علوم مهندسی، از طبیعت الهام گرفته‌اند. یکی از این الهامات، خاصیت انطباق‌پذیری است که در آن، رفتارها و سازمان‌بندی‌ها در جهتِ سازگاری و هماهنگی بیشتر، با محیط پیرامون، تغییر می‌کنند. این تغییر ممکن است، هم در ساختار و هم در رفتار اتفاق بیافتد. هدف از تغییرات، سازگاری بیشتر با محیط پیرامون است. تغییر ممکن است به صورت فوری یا در بازه زمانی طولانی رخ دهد و نتیجه آن در طبیعت، تکامل مطلوب است.

خودانطباق درجه‌ای از انطباق سیستم نرم‌افزاری در جهت عملکرد بهینه در مواجه با تغییرات دینامیک محیط و عدم قطعیت است. خود انطباق که کاربرد وسیعی در شبکه‌های اجتماعی، اینترنت اشیاء، سیستم‌های خودرو و ...دارد، اجازه می‌دهد تا سیستم نرم-افزاری بتواند به‌طور خودکار رفتار خود را در زمان اجرا تغییر دهد و خود را سازگار کند [۱]، که این امر سبب کاهش هزینه پیچیدگی ناشی از نگهداری دستی سیستم می‌شود [۲]. به عبارت بهتر، خود انطباقی، قابلیتی است که سیستم، رفتار خود را در پاسخ به محیط سازگار می‌کند، بنابراین سیستم می‌تواند بسته به زمان و محیط، تغییرات مناسب را در خود اجرا کند[۳]. این ویژگی، منجر به حضور یک مکانیزم انطباق با حداقل تعامل و دخالت انسان در اجرا سیستم می‌شود [۴]. برای مثال، یک ترموستات خانه، دمای خانه را اندازه‌گیری می‌کند و آن را با مجموعه نقاط، کنترل سیستم گرما و سرمای خانه کنترل می‌کند. این یک مثال ساده از مدل ترمودینامیکی است. در سیستم بسیار پیچیده‌تر، نیاز به یک مدل صریح‌تر است. برای مثال در سیستم تهویه هوا برای ساختمان‌های بزرگ، که چندین موقعیت را نظارت و کنترل می‌کند، نیاز به یک مدل صریح از موقعیت ساختمان و دماهای آن، برای کنترل کارآمد است که مشخص کند کدام واحدِ سرمایش و در چه زمانی، روشن شود. مثال دیگر سیستم کنترل ترافیک است که یکی از معمول‌ترین انواع سیستم‌های خود انطباق است که تحقیقات بسیاری در حوزه آن انجام گرفته است. لذا باتوجه به آنچه اشاره شد می‌توان مکانیسم این سیستم‌ها را به چهار فرآیند تجزیه کرد: نظارت بر نهادهای نرم افزاری (خودآگاهی) و محیط (آگاه به زمینه)، تجزیه و تحلیل تغییرات مهم، برنامه‌ریزی برای نحوه واکنش، و اجرا، به طوری که تصمیمات موثر براساس سیگنال‌های دریافتی از حسگرها به منظور خودسازگار سازی اخذ شوند [۵].

روش‌های مختلف پیاده‌سازی این سیستم‌های در [۱] بررسی شده‌اند. این روش‌ها شامل استفاده از شبکه‌های عصبی به عنوان یک روش پردازش داده‌های غیرخطی، منطق فازی به عنوان یک روش پردازش داده‌های عددی با مقادیر بین صفر و یک به منظور کمک به اخذ تصمیمات بهتر در این سیستم‌ها در مواجهه با شرایط عدم قطعیت محیط و الگوریتم ژنتیک به عنوان یک روش بهینه‌سازی می‌باشند. محققین در این مقاله همچنین، چالش‌های تحقیقاتی مرتبط با سیستم‌های خودانطباق را مورد بررسی قرار داده‌اند که عبارتند از:

۱- تعریف دقیق مفاهیم مربوط به این سیستم‌ها و نحوه ارزیابی آن‌ها.
۲- ارائه روش‌های جدید برای پیاده‌سازی این سیستم‌ها با توجه به چالش‌های موجود.
۳- تعریف شاخص‌های جدید برای ارزیابی کارایی، اعتمادپذیری و امنیت این سیستم‌های.
٤- پیدا کردن راه‌حل‌های مناسب برای مواجهه با چالش‌های مربوط به تصمیم‌گیری در این سیستم‌ها.
٥- تعریف روش‌های جدید برای مدیریت منابع در آنها.
٦- پیدا کردن راه‌حل‌های مناسب برای مواجهه با چالش‌های مربوط به تغییرات در شرایط محیطی.

باتوجه به چالش‌های شماره ۱ و ۳، می‌توان گفت نیاز به رهیافتی جهت سنجش و ارزیابی این سیستم‌ها یک مسئله ضروری می‌باشد. برای ارزیابی سیستم خود انطباق نگرش‌های مختلفی وجود دارد، که می‌توان آنها را از دیدگاه‌های مختلف دسته‌بندی کرد که در شکل ۱ نشان داده شده است [۶].



شکل(۱). طبقه‌بندی نگرش‌ها برای ارزیابی سیستم خود انطباق [۶]

همانطور که در شکل فوق مشاهده می‌شود، ارزیابی این سیستم‌ها تنها محدود به ارزیابی عمکردی، زمانی، مکانیزمی و محلی/سراسری نیست، بلکه کیفیت آنها را نیز در بر می‌گیرد. روش‌های ارزیابی کیفی این سیستم‌ها در [۱] شامل شاخص‌های مختلف مانند یادگیری، اعتمادپذیری و امنیت، تعادل بین کارایی و هزینه و ... هستند. به عنوان مثال، برای ارزیابی کارایی این سیستم‌ها، شاخص‌هایی مانند زمان پاسخ، دقت و سرعت تصمیم‌گیری مورد بررسی قرار می‌گیرند. همچنین، برای ارزیابی اعتمادپذیری و امنیت،

شاخص‌هایی مانند نفوذپذیری، تحمل به خطا و ... بررسی می‌شوند. همچنین می‌توان چرایی بهره‌گیری از منطق فازی در تعریف شاخص‌های کیفی نرم‌افزار و استفاده از آنها در ارزیابی کیفیت سیستم‌های خودانطباق [۷] را در چهار دلیل ذیل تبیین نمود:

- تعامل با پیچیدگی: سیستم‌های فازی قابلیت مدل‌سازی و مدیریت پیچیدگی سیستم‌های خودانطباق را دارا بوده و قادر به در نظر گرفتن عوامل مبهم و متغیر در فرآیند ارزیابی هستند.
- قابلیت استدلال مبتنی بر قوانین: سیستم‌های فازی از قوانین و قواعد قابل فهم که یا از تجربه و دانش افراد متخصص یا از طریق الگوریتم‌های بهینه‌گر بدست می‌آیند برای استدلال استفاده می‌کنند. این استدلال برای تصمیم‌گیری در مورد انطباق و تغییرات در سیستم مورد استفاده قرار می‌گیرد.
- توانایی مدل‌سازی تعاملات پیچیده: سیستم‌های فازی قادر به مدل‌سازی تعاملات پیچیده و تأثیرات چند ورودی و خروجی هستند. در محیط خودانطباق، ورودی‌ها و خروجی‌ها ممکن است به صورت غیرخطی و غیرقطعی باشند و سیستم‌های فازی قادر به مدل‌سازی این تعاملات پیچیده هستند.
- امکان استفاده از دانش کاربر: سیستم‌های فازی قابلیت استفاده از دانش کاربر را دارند. با در نظر گرفتن ترجیحات و نظرات کاربران در قالب توابع فایده فازی، سیستم‌های خودانطباق می‌توانند تصمیماتی را بر اساس دانش کاربر انجام دهند.

در راستای ارزیابی سیستم‌های خود انطباق، مطالعات و تحقیقات کمی وجود دارد، که بیشتر آنها به عملکرد کلی سیستم و میزان موفق بودن آن در برآورده ساختن تغییرات تمرکز دارند و به کیفیت و شاخص‌های کیفی این دسته از سیستم‌ها توجه خاصی نشده است. از آن جهت که کیفیت سیستم نرم‌افزار صرف‌نظر از اینکه خود انطباق باشد یا خیر، یک چالش و مسئله مهم در مواجهه با آنها است، لذا فراهم ساختن روشی برای ارزیابی سیستم‌های خود انطباق و به خصوص ارزیابی شاخص‌های کیفی یکی از مسائل ضروری است که با گسترش این دسته از سیستم‌ها نیاز به وجود و رسیدگی به آنها پیش از پیش احساس می‌شود.

در این مقاله سعی شده تا یک روش جدید برای ارزیابی سیستم خود انطباق با توصیف شاخص‌های کیفی مناسب براساس منطق فازی نوع دوم و تنظیم بهینه وزن‌های آنها توسط الگوریتم ژنتیک ارائه شود. این روش، می‌تواند به پژوهشگران در جهت ارزیابی و مقایسه کیفی سیستم‌های نرم‌افزاری خود انطباق کمک کند. روش پیشنهادی حاوی دو فاز است. در فاز اول، شاخص‌های مؤثر در عملکرد سیستم‌های نرم‌افزاری و مشخصات نرم‌افزارهای خودانطباق شناسایی می‌گردد و اولویت و اعتبارسنجی این شاخص‌ها بررسی شود. در فاز دوم یک سیستم ژنتیک-فازی شامل تعیین بهینه وزن‌های فازی توابع عضویت، فازی‌ساز، پایگاه قواعد، موتور استنتاج و غیرفازی‌ساز طراحی می‌شود و در نهایت یک روش برای ارزیابی سیستم‌های خودانطباق ارایه می‌شود.

در ادامه این مقاله در بخش ۲، پیشینه تحقیق آورده شده است و در بخش ۳، مسئله پژوهش و مفاهیم مرتبط با آن شامل سیستم‌های خودانطباق، ارزیابی کیفی و فازی نوعی ۲ تشریح شده است. روش پیشنهادی در بخش ۴ و پیاده‌سازی و ارزیابی روش پیشنهادی این مقاله در بخش ۵ ارائه شده است. نهایتا، خلاصه و نتیجه‌گیری رویکرد پیشنهادی در بخش ۶ جمع‌بندی شده است.

۲. پیشینه تحقیق

برای ارزیابی سیستم‌های نرم‌افزاری مدل‌های کیفی مختلفی ارائه شده است. از جمله مهم‌ترین مدل‌های کیفی می‌توان به McCall [۸]، ISO9126 [۹] و مدل بوهم [۱۰] اشاره کرد. جدول(۱) به طور خلاصه، شاخص‌های کیفی مربوط به این سه مدل کیفی را نشان می‌دهد[۱۱]. خلاصه تحقیقات انجام گرفته شده در حوزه ارزیابی کیفیت نرم‌افزارها از سال ۲۰۰۸ تا کنون در جدول (۲) جهت تسهیل و مقایسه روند آنچه تاکنون ارائه شده، تشریح شده است.

جدول(۱) شاخص‌های کیفی مدل‌های McCall، بوهم و iso9126

معیارها / اهداف	McCall	بوهم	ISO 9126
قابلیت نگهداری	✓	✓	✓
قابلیت انعطاف‌پذیری	✓	✓	
قابلیت آزمایش	✓	✓	قابلیت نگهداری
قابلیت حمل	✓	✓	✓
قابلیت استفاده مجدد	✓	✓	
قابلیت همکاری درونی	✓		
درستی (دقت)	✓	✓	قابلیت نگهداری
قابلیت اعتماد	✓	✓	✓
بهره وری	✓	✓	✓
یکپارچگی	✓	✓	✓
قابلیت استفاده	✓	✓	
قابلیت درک		✓	
قابلیت اصلاح		✓	قابلیت نگهداری

استفاده از منطق فازی برای ارزیابی سیستم‌های نرم‌افزاری، روشی جدید است ولی در برخی از تحقیقات از این روش برای ارزیابی سیستم‌های نرم‌افزار (و نه سیستم‌های خود انطباق) استفاده کرده‌اند. با توجه به مطالعات انجام شده، هیچ استانداردی در خصوص مدل‌های کیفی برای ارزیابی سیستم خود انطباق وجود ندارد. در بیشتر مطالعات با توجه به مقالات و سیستم‌های خود انطباق موجود اقدام به توسعه مدل کیفی دلخواه خود نموده‌اند. مدل‌های کیفی بوهم، McCall و ISO 9126 و دیگر استانداردها

نیز، حیطه عمل گسترده‌ای دارند و اگرچه می‌توانند در ارزیابی سیستم‌های خود انطباق بکار روند، اما برای این امر کافی نیستند.

جدول(۲) تحقیقات و مطالعات انجام شده برای ارزیابی کیفیت نرم‌افزار

مقاله	سال انتشار	توضیحات
[12]	۲۰۰۸	استفاده از منطق فازی، مفهوم فرآیند سلسله مراتبی تحلیل (AHP) جهت ارزیابی کیفیت نرم‌افزار
[13]	۲۰۱۰	استفاده از منطق فازی جهت ارزیابی نرم‌افزار
[14]	۲۰۱۱	استفاده از منطق فازی و مدل iso9126 برای ارزیابی کیفیت سیستم نرم‌افزاری.
[15]	۲۰۱۴	استفاده از منطق فازی، Analytical Network Process(ANP) و Analytical Hierarchical Process(AHP) برای ارزیابی کیفیت نرم‌افزار
[16]	۲۰۱۵	استفاده از منطق فازی و تابع عضویت مثلثی جهت ارزیابی کیفیت سیستم نرم‌افزاری.
[17]	۲۰۱۵	استفاده از منطق فازی برای ارزیابی کیفیت نرم‌افزار بر طبق مدل‌ها و استانداردهای کیفی
[18]	۲۰۱۷	۱)ایجاد یک مدل کیفیت نرم افزار عمومی بر اساس چندین استاندارد کیفیت نرم افزار موجود. ۲) ارائه یک رویکرد ارزیابی سراسری جدید از مدل ارزیابی نرم افزار با استفاده از معیارهای اندازه‌گیری و منطق فازی.۳) پیشنهاد یک الگوریتم نمونه‌سازی برای استخراج مدل ارزیابی نرم‌افزار خاص از مدل های کیفیت نرم افزار عمومی.
[19]	۲۰۱۹	یک مدل فازی سلسله مراتبی چهار سطحی جدید برای ارزیابی کیفیت نرم افزار پیشنهاد می‌کند.
[20]	۲۰۱۸	روش فرآیند تحلیل سلسله مراتبی فازی (FAHP) با روش باکلی (Buckley) برای ارزیابی تأثیر قابلیت نگهداری و زیرمجموعه‌های آن در کیفیت نرم افزار در سطح معماری پیشنهاد شده است.
[21]	۲۰۲۰	طراحی و توسعه یک رویکرد ترکیبی برای تجزیه و تحلیل نرم افزار و برآورد هزینه هر مدل نرم افزاری با ترکیب منطق فازی و مدل کوکومو۲ (COCOMO-II).
[22]	۲۰۲۲	ارائه برخی از معیارهای شباهت جدید سیستم‌های فازی شهودی که با ترکیب تابع نمایی توابع عضویت و تابع منفی توابع غیرعضویت ساخته شده است. همچنین پیشنهاد یک معیار آنتروپی جدید به عنوان یک مرحله برای محاسبه وزن شاخص‌ها در مدل تصمیم‌گیری چند معیاره (MCDM) و استفاده از از این مدل برای ارزیابی کیفیت پروژه‌های نرم افزاری.
[23]	۲۰۲۱	کیفیت محصول نرم افزار بر اساس سیستم استنتاج فازی در استاندارد ISO اندازه‌گیری می‌شود.
[24]	۲۰۲۳	پیشنهاد یک روش جدید ترکیبی برای آزمایش تأثیرگذاری پارامتر (PIT) در توسعه نرم‌افزار با تبدیل داده‌های زبانی به دست آمده از کارشناسان به اعداد فازی مثلثی و استفاده از یک تکنیک فرآیند سلسله مراتبی تحلیلی فازی- تکنیک برای اولویت سفارش با شباهت به راه‌حل ایده آل (AHP-TOPSIS) برای ارزیابی و اولویت‌بندی PIT ها به طور سیستماتیک.

محققان در [۲۵] چالش‌های ارزیابی کیفیت در سیستم های خودانطباق را به صورت زیر بیان می‌کنند:

۱. **مشکلات معماری:** اولین چالش مربوط به تنوع معماری‌های ارائه شده توسط این سیستم‌های است. به دلیل وجود مدل‌های معماری مختلف (به عنوان مثال، متمرکز، توزیع شده)، الگوهای معماری مختلف و همچنان روش‌های مختلف برای انطباق این معماری‌ها در زمان اجرا، توسعه روش‌های ارزیابی کیفیت عمومی کافی برای پشتیبانی از همه ترکیب‌ها، ایجاد مجموعه‌ای از راه‌حل‌های هدایت شده به سمت معماری‌های خاص، چالش برانگیز می‌شود.

۲. **مقادیر آستانه برای شاخص‌های کیفیت:** در مورد خود وظیفه ارزیابی کیفیت، از طریق مطالعه ادبیات مشاهده می‌شود که برخی از تحقیقات توسعه‌یافته بر اساس شاخص‌های کیفیت، مقادیر مرجعی را برای تفسیر نتایج اندازه‌گیری ارائه نمی‌دهند، که ممکن است ارزیابی و تضمین کیفیت را تحت تأثیر قرار دهد.

۳. **پیچیدگی**: پیچیدگی این سیستم‌ها عاملی است که بر چندین راه‌حل ارزیابی کیفیت تأثیر می‌گذارد.

۴. **ابزارهای پشتیبانی:** چالش فعلی ناشی از پیچیدگی این سیستم‌ها، کمبود ابزارهای خودکار برای کمک به وظایف آزمون و ارزیابی کیفیت است.

آن‌ها همچنین برخی از راه‌حل‌های این چالش‌ها را به شرح ذیل بیان کرده‌اند:

- توسعه ابزار برای پشتیبانی از آزمون سیستم‌های خودانطباق و فعالیت‌های ارزیابی کیفیت آنها.
- تعریف حد آستانه برای شاخص‌های کیفیت.
- ارزیابی شاخص‌های ضعیف مطالعه شده (به عنوان مثال، قابل حمل و مناسب بودن عملکرد) و تجزیه و تحلیل تاثیر آنها بر سیستم‌های خودانطباق.
- تعریف ویژگی‌های جدید و شاخص‌های کیفی متناسب با جنبه‌های پویای این سیستم‌ها.
- توسعه رویکردها برای اولویت بندی شاخص‌های کیفیت با توجه به عملیات و حوزه‌های این سیستم‌ها.
- نگاشت تضاد بین شاخص‌های کیفیت.

رایبولت و همکاران [۶] یک طبقه‌بندی برای ارزیابی کیفیت سیستم‌های خودانطباق ارائه می‌دهند. بر اساس این طبقه‌بندی، ارزیابی کیفیت در این سیستم‌ها می‌تواند بر اساس ویژگی‌های کیفی یا معیارهای نرم‌افزاری انجام شود. ارزیابی‌های مبتنی بر ویژگی‌های کیفی برای ارزیابی کل سیستم یا بخشی از آن با در نظر گرفتن معنادار ویژگی‌های کیفی مانند کارایی، بهینه‌سازی و قابلیت اطمینان به کار می‌روند. ارزیابی‌های مبتنی بر معیارهای نرم‌افزاری برای ارزیابی کیفیت سیستم از یک دیدگاه محلی (مانند معیاری

مرتبط با یک جزء یا عامل) یا از یک دیدگاه کلی (مانند معیاری مرتبط با کل سیستم) مورد استفاده قرار می‌گیرند.

کاری که در [۲۶] ارائه شده، رویکردی را برای انجام تطبیق‌پذیری در سیستم‌های خودانطباق مبتنی بر کامپوننت با معماری متمرکز بر اساس تعیین ویژگی‌ها، اندازه‌گیری و بهینه‌سازی خواص و ویژگی‌های کیفی در مدل‌های ویژگی ارائه می‌کند. نویسندگان گزارش می‌دهند که برخورد با یک سیستم توزیع‌شده پیچیده خواهد بود، زیرا مدیریت ویژگی‌های کیفی (مانند امنیت، در دسترس‌بودن، قابلیت مقیاس‌پذیری و مصرف باتری) دشوارتر خواهد شد.

سرال و همکاران [۲۷] ادعا می‌کنند که ارزیابی مکانیزم‌های خودانطباق که تضمین کننده حداکثر رضایت کاربر (مانند کارایی، راحتی، صرفه‌جویی در انرژی و غیره) هستند، به دلیل تفاوت‌های ذاتی بین کاربران و تغییر زمینه عملیاتی سیستم، یک چالش عمده است.

معین فر و همکاران [۲۸] گزارش می‌کنند که یکی از چالش‌های دستیابی به قابلیت اطمینان در سیستم‌های خودانطباق، مربوط به رسیدگی به مسائل محدودیت زمانی و فضایی است که محدودیت‌های مهمی برای این سیستم‌ها به شمار می‌روند. برای ارزیابی سیستم‌های خودانطباق، باید توجه داشت که معیارها و ویژگی‌های قابل ارزیابی متعددی وجود دارد. بر همین اساس، فرصت‌های پژوهشی همچنین می‌تواند به مقادیر مرجع برای معیارهای کیفیت، همانطور که در [۲۹] ذکر شده است، مرتبط باشد. از این منظر، انجام مطالعاتی مانند آزمایش‌ها و کاربرد تکنیک‌های یادگیری ماشین برای تعریف آستانه‌هایی برای معیارهای کیفیت در سیستم‌های خود-انطباق را می توان پیشنهاد کرد.

در مواجه با سیستم‌های خودانطباق، ارزیابی مکانیزم‌های تطبیق‌پذیری می‌تواند به ویژگی‌ها و معیارهای کیفیتی بستگی داشته باشد که در زمان طراحی و زمان اجرا قابل اندازه‌گیری هستند [۳۰]. ارزیابی در زمان طراحی به راه‌حل‌هایی می‌پردازد که به معماری یک سیستم نرم‌افزاری برای پشتیبانی از قابلیت خود-انطباق‌پذیری اشاره می‌کند. ارزیابی ویژگی‌های خود-انطباق‌پذیر در زمان اجرا بیشتر به معیارهایی مرتبط است که به دنبال ارائه ارزیابی کیفی یا کمی از نرم‌افزار هستند. به عنوان مثال، کاری که توسط کادوم و همکاران [۳۱] ارائه شده است، مجموعه‌ای از معیارها و اندازه‌گیری‌ها را برای سیستم‌های خود-انطباق‌پذیر نشان می‌دهد، که از میان آن‌ها می‌توان به تأخیر، معیاری برای سنجش عملکرد در دیدگاه زمان اجرا، اشاره کرد که نشان‌دهنده تغییر بین زمان پاسخ با وجود قابلیت خودانطباقی و بدون وجود مکانیزم‌های خودانطباقی است.

در [۳۲] محققان به بررسی چگونگی تجمیع معیارهایی که برای ارزیابی کیفیت مدل‌های ویژگی پیشنهاد شده‌اند، می‌پردازند. آن‌ها برای تجمیع این معیارها از تئوری منطق فازی استفاده کردند. معیارهای تجمیع‌شده جدید را می‌توان برای ارزیابی جنبه‌های مختلف و پیچیده یک مدل ویژگی، مانند اندازه، پایداری، انعطاف‌پذیری و پویایی، به کار برد. علاوه بر این، برای ارزیابی استفاده از معیارهای تجمیع‌شده جدید، آنها را در مدل‌های ویژگی مختلف به کار بست. یافته‌های آن‌ها نشان می‌دهد که معیارهای تجمیع‌شده می‌توانند به مهندس حوزه‌ی مربوطه در ارزیابی قابلیت نگهداری مدل‌های ویژگی کمک کنند.

ذی‌نفعان انسانی اغلب از تناقضات پنهان بین شاخص‌های کیفی آگاه نیستند. به منظور رفع این مسئله، محققان در [۳۳] یک رویکرد ارائه می‌دهند که از تکنیک‌های یادگیری ماشین (کاهش بعد، خوشه‌بندی و یادگیری درخت تصمیم) برای توضیح دادن استدلال پشت برنامه‌ریزی خودانطباق استفاده می‌کنند. رویکرد آن‌ها بر تناقضات بین شاخص‌های کیفی تمرکز کرده که منجرب به انتخاب وزن‌ها در توابع فایده و تولید برنامه‌های متفاوت می‌شود. آن‌ها به انسان‌ها در درک تناقضات شاخص‌های کیفی کمک می‌کنند.

در [۳۴] یک رویکرد جدید به نام CHESS ارائه شده که با استفاده از مهندسی آشوب، سیستم‌های خودانطباق را ارزیابی می‌کند. معماری ارائه شده در این مقاله یک نمای کامل از استفاده از شطرنج از طریق دو مورد مطالعه مبتنی بر میکروسرویس را ارائه می‌دهد: یک مورد مطالعه مربوط به دفتر هوشمند و یک برنامه نمونه موجود به نام Yelb. این معماری شامل یک سرویس مدیریت سامانه، یک سرویس خودنظارتی و پنج سناریوی تزریق خطا که شامل خطاهای زیرساختی و خطاهای عملکردی است، می‌باشد. هر یک از این اجزا قابلیت گسترش یا جایگزینی آسان جهت استفاده از رویکرد CHESS در مطالعه موردی جدید را دارند.

همانطور که مشاهده می‌شود نگرش‌های ارزیابی متنوعی تاکنون ارائه شده اند که باید گفت، روش ارزیابی اولیه از جمله روش‌های مبتنی بر سناریو نمی‌توانند از سیستم خود انطباق پشتیبانی کنند. با توجه به روش‌ها و مطالعات موجود نیز، اگرچه، روش‌های مختلفی، با توجه به جنبه‌های مختلفی از سیستم خود انطباق اقدام به ارزیابی اینگونه از سیستم‌ها کردند، اما هیچ یک از آنها مختص این سیستم‌ها نیستند[۳۵]. از طرف دیگر، بیشتر نگرش‌های ارزیابی استفاده شده برای سیستم‌های خود انطباق در تحقیقات موجود، به عملکرد سیستم توجه دارند و عملکرد سیستم را مورد ارزیابی قرار می‌دهند و روشی برای ارزیابی کیفیت سیستم نرم‌افزاری خود انطباق ارائه نمی‌دهند. با این اوصاف، تلاش برای ارزیابی کیفیت سیستم خود انطباق و همچنین مشخص کردن یک

مدل کیفی برای ارزیابی این دسته از سیستم‌ها امری ضروری و لازم است.

دلیل اصلی استفاده از منطق فازی، سهولت و انعطاف‌پذیری آن است. منطق فازی می‌تواند مشکلات ناشی از اطلاعات مبهم و ناتمام را حل کند و توابع غیرخطی پیچیده را مدل نماید. منظور از اطلاعات مبهم درواقع عدم قطعیت و نسبی بودن مقادیر کیفی است. منطق کلاسیک درک بالایی از سیستم را نیاز دارد. در حالی که منطق فازی برای مدلسازی یک سیستم پیچیده توسط سطح بالایی از انتزاع از دانش و اطلاعات کلی بدون درک عمیق از سیستم مورد استفاده قرار می‌گیرد. در مدل‌های ارزیابی، برخی از داده‌های ورودی به طور معقول تعیین می‌شوند (به عنوان مثال ارزیابی بر طبق شاخص‌های کیفی، توابع ارزیابی جزئی برای شاخص‌های کمی، انتخاب یک نوع مناسب تجمیع، وزن‌دهی به شاخص‌ها یا در نهایت، توصیف مبتنی بر قانون بین مقادیر شاخص‌ها و ارزیابی کلی آنها). از آنجا که عدم قطعیت ویژگی متداول هرگونه اطلاعات خبره است، نظریه مجموعه فازی یک ابزار ریاضی مناسب برای ایجاد چنین مدل‌هایی می‌باشد.

محدودیت‌های متعددی از تحقیقات کنونی در ارزیابی سیستم‌های خودانطباق وجود دارد. اول اینکه، به مدل نرم‌افزاری نیاز است، در حالی که در سیستم‌های نرم‌افزاری دقیق مدل‌سازی تقریبا غیرممکن است. دوم اینکه، طراحی کنترل‌کننده‌ها و استراتژی‌های کنترل نیاز به دانش بسیار زیاد دارد که این امر برای محققان و کارکنان علوم کامپیوتر بسیار سخت است[۳۶]. لذا با توجه به این امر که کیفیت نرم‌افزار کمیتی کیفی است، در نتیجه منطق فازی می‌تواند یک راه‌حل بسیار مناسب برای این مسئله باشد. به همین دلیل تحقیقاتی نیز در این خصوص فراهم آمده است که در جدول (۲) به آن اشاره شده است. اما تاکنون تحقیق و مطالعه‌ای برای ارزیابی سیستم خود انطباق توسط منطق فازی نوع دوم صورت نگرفته است و تحقیقات فوق برای سیستم‌هایی غیر از سیستم‌های خود انطباق مورد استفاده قرار گرفتند. روش ارزیابی فازی می‌تواند مشکل عامل‌های چندگانه، مشکلات سیستماتیک، علمی، عملیاتی و غیره را حل کند. بنابراین، یک نوع روش ارزیابی پیشرفته جامع را فراهم می‌آورد [۳۷]. منطق فازی در کنار مزیت‌های فراوانی که به همراه می‌آورد، معایبی نیز دارد. همین امر سبب ایجاد فازی نوع ۲ شده است. استفاده از فازی نوع ۲ در ارزیابی سیستم‌های نرم‌افزاری می‌تواند با رفع مشکل عدم قطعیت در سیستم فازی نوع ۱، به دقیق‌تر شدن ارزیابی کمک بسیاری کند.

شایان ذکر است که باتوجه به عدم دسترسی به جزئیات سیستم‌های خودانطباق موجود در تحقیقات مختلف، در این مقاله از سیستم کنترل ترافیک با نام **InSync** که یک سیستم خودانطباق کنترل ترافیک (**ATCS**) بوده و حاوی شاخص‌های متعدد کیفی می‌باشد استفاده شده است. این سیستم حاوی شاخص‌ها و متریک‌های کیفی متعددی است که با کمک فرد خبره از مستندات موجود برای این سیستم [۳۸-۴۱] جمع‌آوری شده که این اطلاعات به اختصار در جدول (۳) بیان شده‌اند. از این مستندات ۳۲ شاخص شامل پایداری، زمان تنظیم کوتاه، پایان دهی، مقیاس پذیری، دلنه‌بندی، پیشگویی، سرعت پاسخ خودمختار، هزینه، درجه خودمختاری، خودپیکربندی، خودبهینگی، خود حفاظتی، بازبودن، قابلیت حمل، قابلیت نگهداری، قابلیت عملیاتی، دقت، تخطی کم، سازگاری، حساسیت، پرهیز از شکست، استحکام، امنیت، دقت تصمیم‌گیری، سطح خودمختاری، خود بهبودی، خودآگاهی، آگاهی به زمینه، ثبات، رضایت، قابلیت استفاده و قابلیت اعتماد بدست آمده است. همچنین ۴۷ متریک شامل حسگر فعال، زمان پاسخ (اجرا)، زمان انتظار، استفاده از منابع (بهره‌وری منابع)، سادگی، قابلیت گسترش، یکپارچگی، مناسب بودن، قابلیت همکاری درونی، تاخیر، توان عملیاتی، ظرفیت، رفع شکست و نقص، تشخیص شکست و نقص، پرهیز از شکست و نقص، قابلیت دسترسی، وفق‌پذیری ساختار داده، وفق‌پذیری محیطی سخت افزا، وفق‌پذیری محیط اجرایی، وفق‌پذیری محیط نرم افزاری، درگاه‌های کاربر پسند نصب آسان، آسانی تنظیمات مجدد، همزیستی قابل دسترس، زمان تصمیم، پیچیدگی رفتار، دلنه‌بندی ریز/درشت، ثبات، رضایت، فراگیری عملیات، دانش، قدرت اتصال، پیچیدگی تعاملات، محرمانگی، ساختار دهی مجدد، عدم انکار، سرعت تصمیم گیری، زمان شکست، زمان بازیابی، سودمندی، قابلیت بازسازی و شروع مجدد، قابلیت تحلیل، قابلیت تغییر، قابلیت آزمایش، قابلیت یادگیری، قابلیت یادسپاری و قابلیت درک از این مستندات استخراج می‌شوند.

۳. مسئله پژوهش

هر مسئلهٔ تحقیق، ادبیات و مفاهیم منحصر بفرد خود را دارا می‌باشد و در این بخش به تشریح مفاهیم مرتبط با مسئله شامل سیستم‌های خودانطباق، ارزیابی کیفیت نرم‌افزار و فازی نوع ۲ می‌پردازیم.

۳-۱. سیستم‌های خودانطباق

بر طبق تعریف آژانس DARPA نرم‌افزار خود انطباق، رفتار و تغییرات رفتار خود را برای بهبود کارایی و عملکرد بهتر، ارزیابی کرده و آنرا برای پاسخ به تغییرات در محیط عملیاتی اصلاح می‌کند. محیط عملیاتی می‌تواند شامل ورودی کاربر نهایی، حسگرها و دستگاه‌های سخت‌افزاری خارجی، یا برنامه دیگر شود [۴۲]. محققان معمولاً از تعاریف زیر برای سیستم‌های خود انطباق استفاده می‌کنند:

جدول (۳). مستندات و اطلاعات منتشر شده درباره ارزیابی و عملکرد سیستم InSync

نام مستند	سال انتشار	اطلاعات
Criteria For the Selection and Application of Advanced Traffic Signal Systems [۳۸]	۲۰۱۳	سیستم های کنترل ترافیک مختلفی توسعه یافتند و اکنون در حال استفاده هستند از جمله آنها می‌توان به سیستم های زیر اشاره کرد: • SCATS: در دهه ۱۹۹۰ تولید شد. این سیستم یک سیستم انطباق ترافیک است، که ترافیک را در جاده ها شناسایی می کند. • ACS-Lite: در سال ۲۰۰۶ برای کنترل و بهبود ترافیک چهار راه ها معرفی شد. • InSync: در سال ۲۰۰۹ کاربردی شد. از پردازش تصویر و پردازش فیلم استفاده می کند تا ترافیک را شناسایی کند. این مستند سیستم های کنترل ترافیک را که تاکنون طراحی، ساخته و اجرایی شدند مورد بررسی و ارزیابی قرار داده است. این مستند، بر طبق نوع عملکرد و همچنین برخی از معیارهای کیفی این سیستم ها را مورد مقایسه و تحلیل قرار داده است. شاخص پایداری، ثبات، هزینه را از این مستند استخراج کرد.
Comparative Evaluation of Insync and Time-Of-Day Signal Timing Plans Under Normal and Varied Traffic Conditions [۳۹]	۲۰۱۲	این مستند به سیستم insync پرداخته است. این گزارش ابتدا به بررسی عملکرد سیستم insync می پردازد و سپس عملکرد آن را مورد ارزیابی قرار می دهد. متریک-هایی مانند کارایی، رضایت از سیستم، را از این مستند استخراج کرد.
InSync Adaptive Traffic Control System for the Veterans Memorial Hwy Corridor on Long Island, NY [43]	۲۰۱۲	این مستند به بررسی سیستم insyc و تشریح عملکرد و موارد کاربرد آن می پردازد. همچنین متریک های حسگر فعال، استفاده از منابع، قابلیت استفاده را استخراج کرد.
Adaptive-Traffic-Signal-Control Systems in the United States: updated summery and comparison [۴۱]	۲۰۱۰	این مستند یک ارزیابی کلی از عملکرد سیستم های کنترل ترافیک ارائه می دهد. از آن می توان مقادیر شاخصها و یا متریک های هزینه، قابلیت اطمینان، قابلیت دسترسی، قابلیت نگهداری، قابلیت نصب، قابلیت پیکر بندی (مجدد)، تاخیر را اسخراج کرد.
Analysis of Adaptive Traffic Control Systems and design of a Decision Support System for better choice [۴۰]	۲۰۱۳	این مستند نیز به بررسی سیستم های کنترل ترافیک از جمله inSync و Scats می پردازد. این مستند علاوه بر تشریح عملکرد سیستم به ارزیابی آن نیز می پردازد. از جمله شاخصها یا متریک های قابل استخراج در این سیستم می توان به ایمنی، قابلیت یکپارچگی، سودمندی سیستم اشاره کرد.

- خود انطباقی، توانایی سیستم برای تنظیم رفتار خود در پاسخ به محیط است. پیشوند "خود" نشان می‌دهد که سیستم‌ها به طور مستقل و خود مختار (یعنی، با حداقل دخالت یا بدون دخالت) تصمیم می‌گیرند چگونه بر طبق زمینه و محیط، خود را انطباق و یا سازماندهی کنند[۴۴].
- در چارچوب سیستم‌های چند مدله، انطباق یک رویه یا روش برای تعویض بین مدل‌ها است. بنابراین انطباق می-تواند به عنوان توانایی یک سیستم برای دستیابی به اهداف خود در یک محیط متغیر تعریف شده و با انتخاب مجدد و تغییر بین مدل‌ها، تعریف شود[۴۵].
- یک سیستم خود انطباق شامل یک سیستم حلقه بسته است (یعنی خود را در زمان اجرا با استفاده از بازخورد ناشی از تغییرات مداوم سیستم اصلاح می‌کند). مفهوم سیستم خود انطباق به نیازمندی‌های کاربر، خصوصیات سیستم و خصوصیات محیطی بستگی دارد. نرم‌افزار خودانطباق مستلزم قابلیت اطمینان بالا، استحکام زیاد، انطباق پذیری زیاد و قابلیت دسترس‌پذیری بالا است[۴۶]
- سیستم خود انطباق، رفتار خود را ارزیابی می‌کند و عملکرد خود را بر طبق ارزیابی‌ها تغییر می‌دهد[۴۲].

از نرم‌افزار خود انطباق، انتظار می‌رود که نیازمندی‌های زمان اجرا (یعنی اهداف) خود را انجام دهند، حتی زمانی که با تغییرات داخلی و خارجی رو به رو می‌شود. از یک سیستم خود انطباق انتظار می‌رود که تغییراتی بر اهدافش اثر دارند،کشف کند و فعالیت مناسب را با هزینه و زمان قابل قبول انجام دهد. این اهداف می‌توانند از طریق نظارت سیستم بر محیط برای کشف تغییرات جهت گرفتن تصمیم مناسب، تحقق یابد[۴۷].

۳-۲. ارزیابی کیفیت نرم‌افزار

کیفیت نرم‌افزار یک جنبه مهم برای توسعه‌دهندگان، کاربران و مدیران پروژه است. محققان بسیاری سعی در طراحی مدلی مناسب برای تعریف کیفیت نرم‌افزار از دیدگاه‌های مختلف داشتند. مطالعات کیفیت نرم‌افزار یک مجموعه سیستماتیک و برنامه‌ریزی شده از فعالیت‌ها برای اطمینان از سودمند بودن نرم‌افزار را فراهم می‌کند و

شـــامل زیر عنوان‌های مختلف مانند تامین کیفیت نرم‌افزار[۱]، کنترل کیفیت[۲] و مهندسی کیفیت[۳] است. بر طبق IEEE، کیفیت نرم‌افزار مجموعه‌ای از شـــاخص‌های نرم‌افزار اســـت و به صـــورت زیر تعریف می‌گردد[۱۴]:

۱- درجه‌ای که سیستم، مؤلفه‌ها یا محصول، نیازمندی‌ها را تامین می‌کنند.
۲- درجه‌ای که ســـیســـتم، مؤلفه‌ها یا فرآیند، نیازهای مشتریان و کاربران را تامین می‌کند.
۳- کیفیت شـــامل هدایت شـــاخصـــها برای رضـــایت نیازمندی‌هایشان می‌شود.

در IEEE ارزیابی اینگونه توصـیف شـده اسـت: «فرآیند تحلیل یک آیتم نرم‌افزار برای تعیین اختلاف بین شـرایط مورد نیاز و وضعیت موجود آیتم (برای تعیین خطاها) و تست مؤلفه‌های آیتم نرم‌افزاری اســـت». از این توصـــیف ما می‌توانیم ببینیم ارزیابی را می‌توانیم رویکردی در ارزیـابی و ســـنجش نرم‌افزارهـا و ازجملـه نرم‌افزارهای خودانطباق در نظر بگیریم.

نیازمندی‌های کیفی مانند کارایی و بهینه‌سازی، در معماری و نرم‌افزار سیستم‌های خود انطباق بسیار مهم هستند. سیستم‌های خود انطباق خواهان تکامل مداوم هســـتند و هدف آن‌ها اطمینان از کیفیت انجام وظایف زمان اجرا و پیکربندی مجدد است. در نتیجه، ارزیابی کیفی از انطباق‌ها و موفقیت در انجام تغییرات زمان اجرا مسئله مهمی است که باید در این سیستم‌ها در نظر گرفته شود[۶]. در این راسـتا، مهندسـی نرم‌افزار چندین مکانیزم و اسـتاندارد برای ارزیابی کیفیت سـیسـتم‌ها از مرحله طراحی معماری تا پیاده‌سـازی راه‌حل، فراهم کرده اسـت[۶]. مکانیزم‌های ارزیابی در سـیسـتم‌های خود انطباق می‌توانند وابسته به موارد زیر باشند:

۱. **شـــاخص‌های کیفی:** برای ارزیابی کیفیت ســـیســـتم یا بخش‌هایی از آن اســـتفاده می‌شـــود. که به طور خاص بر ویژگی‌های کیفی معنی‌دار برای سیستم‌های خود انطباق تمرکز می‌کند. از جمله این ویژگی‌های کیفی می‌توان به کارایی و بهینه‌سازی اشاره کرد.

متریـک‌هـای نرم‌افزار: برای ارزیـابی کیفیـت انطبـاق از دیدگاه محلی یا سـراسـری اسـتفاده می‌شـود. دیدگاه محلی می‌تواند شامل انطباق یک محصول باشد و دیدگاه سراسری می‌تولند به زمان برای انطباق یا درجه عدم تمرکز اشـــاره کند. دو مرحله مهم در مکانیزم‌های انطباق وجود دارد:

الف) در زمان طراحی، در طول توسعه سیستم که مکانیزم انتخاب می‌شود.

ب) در زمان اجرا، که سیستم در حالت تمام عملیاتی است و کیفیت انطباق باید به طور پیوسته نظارت شود.

۳-۳. فازی نوع ۲

دکتر لطفی زاده ابداع کننده منطق فازی[۴] معتقد به لزوم در نظر گرفتن ابهام در ماهیت علم بود. برخلاف دیگران که معتقد بودند که باید تقریب‌ها را دقیق‌تر کرد تا بهره‌وری افزایش یابد، ایشـان معتقد به لزوم سـاخت مدل‌هایی که ابهام را به عنوان بخشـی از سـیسـتم مدل نماید بودند. برخلاف منطق ارسطویی که بر درست یا نادرست بودن گزاره‌ها اسـتوار اسـت، در منطق فازی، جملاتی هسـتند که مقداری درست و مقداری نادرست هستند. برای مثال، جمله "هوا سرد است" یک گزاره منطقی فازی می‌باشد که درستی آن گاهی کم و گاهی زیاد اسـت. گاهی همیشـه درسـت و گاهی همیشـه نادرست و گاهی تا حدودی درست است.

پایگاه قواعد فازی، مجموعه‌ای از قواعد اگر-آنگاه فازی را نشـان می‌دهد. موتور اسـتنتاج فازی این قواعد را به یک نگاشـت از مجموعه‌های فازی در فضای ورودی به مجموعه‌های فازی در فضای خروجی بر اساس اصول منطق فازی ترکیب می‌کند. در سیستم‌های فازی با فازی‌ساز ابتدا در ورودی سیستم متغیرهای عددی تبدیل به متغیرهای فازی می‌شـوند و به موتور اسـتنتاج وارد می‌شـوند و خروجی موتور اسـتنتاج که یک کمیت فازی اسـت توسـط غیر فازی‌سـاز به یک عدد ثابت تبدیل می‌شـود. توابع عضـویت فازی بیان‌کننده میزان تعلق یک عدد به یک متغیر زبانی می‌باشـد و این میزان تعلق همواره بین ۰ تا ۱ می‌باشـد. یک مجموعه فازی نوع ۱، یک متغیر منفرد $x\epsilon X$ است که می‌تواند به صورت زیر ارائه شود:

$$A = \{x, \mu_A(x) |\ \forall x\epsilon X\} \qquad (۱)$$

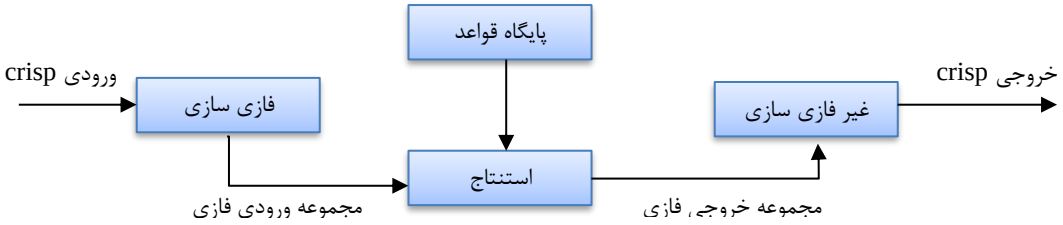
A می‌تواند به صورت زیر نیز تعریف شود:

$$A = \int_{x\epsilon X} \mu_A(x)/x \qquad (۲)$$

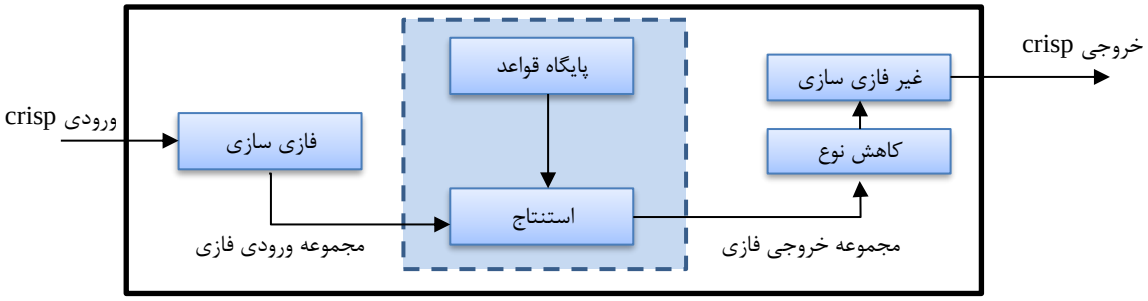
توابع عضویت انواع مختلفی دارند که به تابع عضویت مثلثی، ذوزنقه ای، گوســی و ســیگموئید می توان اشـــاره کرد. روش کار سیسـتم فازی نوع ۱ در شـکل ۲ نشـان داده شـده اسـت. پروفسـور لطفی‌زاده در سـال ۱۹۷۵ مجموعه‌های فازی نوع ۲ را به عنوان توسعه‌ای از مجموعه‌های فازی معرفی کرد. مجموعه‌های فازی نوع ۲ دارای درجه عضـویت‌های فازی می‌باشـند؛ ازاین رو، به آنها مجموعه‌های فازی -فازی نیز می‌گویند که در برخورد با عدم قطعیت‌ها توانایی کاهش اثر و مدل کردن آنها را دارند. یک مجموعه فازی نوع ۲ در مواجهه با عدم قطعیت‌های زبانی برای ارائه‌ی اطلاعاتی بیشتر از یک درجه عضویت ساده، اندازه‌ای از

---

۱ Software quality assurance
۲ Quality control
۳ Quality engineering
۴ Fuzzy Logic



شکل (۲). روش کار سیستم فازی نوع ۱[۴۸]



شکل (۳). سیستم فازی نوع ۲[۴۹]

پراکندگی را حول یک عدم قطعیت زبانی در نظر می‌گیرد. درحقیقت، مجموعه فازی نوع۱ تقریب درجه اول عدم قطعیت و مجموعه فازی نوع۲ تقریب درجه دوم عدم قطعیت هستند. به این ترتیب، مجموعه‌های فازی نوع ۲ در کاهش اثر عدم قطعیت در قوانین فازی بهتر عمل می‌کنند، علاوه بر کاهش اثر عدم قطعیت در قوانین فازی، به علت فازی بودن توابع عضویت در مجموعه‌های فازی نوع۲ امکان مدل کردن عدم قطعیت‌های زبانی و داده‌ها به نحو مؤثری بهبود پیدا می‌کند[۵۰]. مشابه با فازی نوع ۱، سیستم فازی نوع ۲ نیز شامل فازی‌ساز، پایگاه قوانین، موتور واسط فازی، و پردازش‌گر خروجی است. شکل ۳ نمای یک سیستم فازی نوع ۲ را نشان می‌دهد.

پردازش‌گر خروجی، شامل کاهش دهنده نوع و غیر فازی‌ساز است و مجموعه خروجی فازی نوع ۱ یک عدد درست را (از غیرفازی‌کننده) تولید می‌کند. فازی نوع ۲ توسط قوانین if-then مشخص می‌شود. اما مجموعه‌های استنتاجی خود را دارد. مجموعه فاصله[۱]، تا حد زیادی توابع عضویت ثانویه را ساده می‌کند. مجموعه فازی نوع ۲ که با $\tilde{A}$ مشخص می‌شود، توسط توابع عضویت نوع ۲ یعنی $\mu_{\tilde{A}}(x,u)$ مشخص می‌شوند، که $x\epsilon X$ و $u\epsilon J_x \subseteq (0,1)$ است یعنی

$$\mu_{\tilde{A}}(x,u): X \to [0,1] \qquad (۳)$$

همچنین J_x تابع عضویت اولیه در فاصله (۰ و ۱) است و u مقدار تابع عضویت است. درنتیجه:

$$0 \le \mu_{\tilde{A}}(x,u) \le 1 \qquad (۴)$$

علاوه براین $\tilde{A}$ به صورت زیر نیز بیان می‌شود:

$$\tilde{A} = \int_{x\in X}\int_{u\in J_x} \mu_{\tilde{A}}(x,u)/(x,u) \quad J_x \subseteq (0,1) \qquad (۵)$$

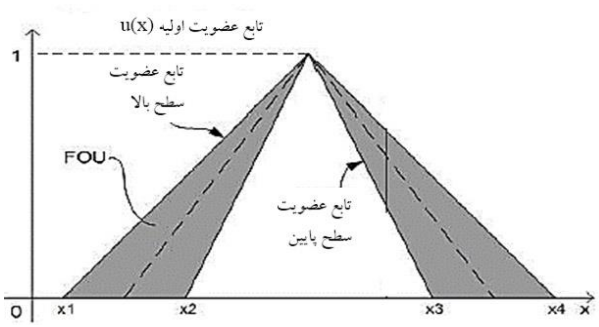
که $\iint$ اجتماع روی همه x و u های قابل قبول را مشخص می‌کند.

$$\tilde{A} = \{((x,u),\mu_{\tilde{A}}(x,u)) \; \forall x \in X, \forall u \in J_x \subseteq (0,1)\} \qquad (۶)$$

هنگامی‌که همه توابع عضویت ثانویه، از مجموعه فازی نوع ۲ یکی شدند، مجموعه فاصله فازی نوع ۲ به صورت زیر مشخص می‌شود.

$$\tilde{A} = \int_{x\in X}\int_{u\in J_x} 1/(x,u) \quad J_x \subseteq (0,1) \qquad (۷)$$

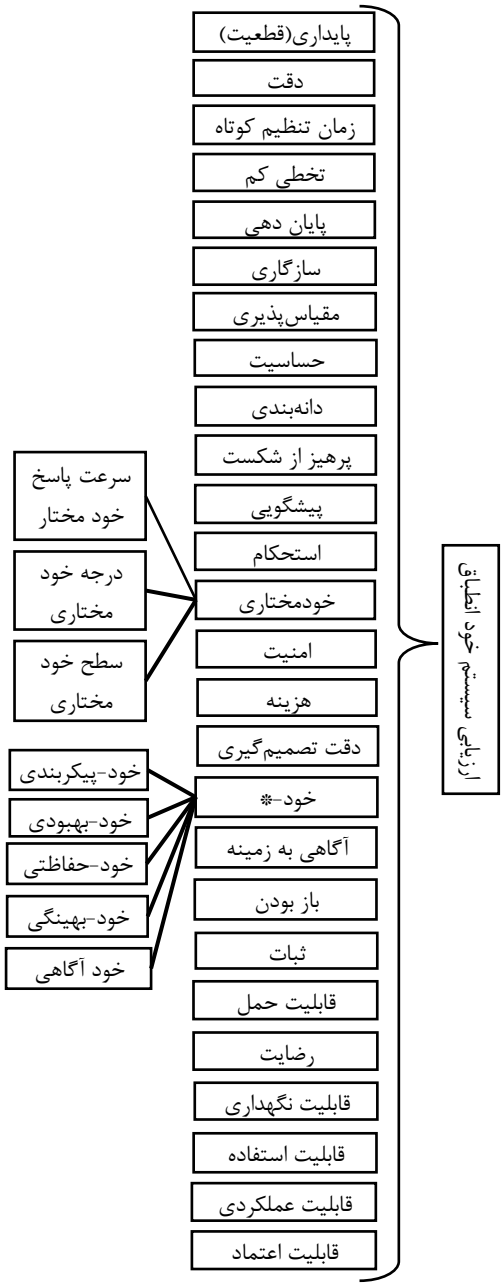
عدم قطعیت در تابع عضویت اولیه متشکل از یک منطقه محدود که به عنوان FOU[۲] شناخته می‌شود، که اجتماع همه توابع عضویت اولیه است[۵۱]. شکل ۴ نمایی از تابع عضویت مثلثی در منطق فازی نوع ۲ را نمایش می‌دهد.



شکل (۴). نمایی از تابع عضویت مثلثی در فازی نوع ۲ [۴۸]

[1] interval sets

[2] Foot-print of uncertainty



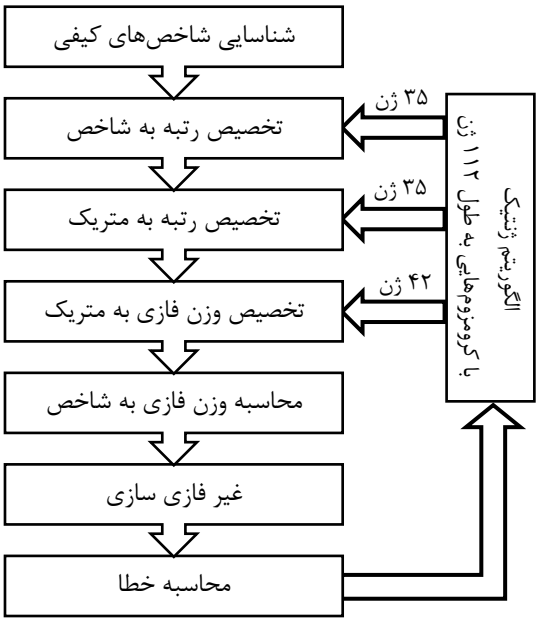
شکل (۵). شاخص‌های مؤثر در ارزیابی کیفی سیستم خودانطباق

۳-۴. شاخص‌های کیفی سیستم‌های خود انطباق

شـــاخص‌های کیفی که برای ارزیابی نرم‌افزارهای خود انطباق در نظرگرفته شـــده‌اند، مؤلفه‌های انتزاعی مانند ســـودمندی، عملکرد، خود انطباق، استحکام و مؤلفه پشتیبانی می‌باشند که نمایش‌دهنده مفاهیمی سـطح بالا هســتند و به راحتی قابل اندازه‌گیری نیســتند. برای ارزیابی این مولفه‌ها نیاز اســت که آن‌ها به کمیت‌های ســطح پایین و قابل اندازه‌گیری نگاشت شوند، که در این نگاشت هر مؤلفه به چند متریک قابل اندازه‌گیری تبدیل می‌شـــود. شـــکل (۵) نمای کلی از شاخص‌های مهم در ارزیابی ســیستم خود انطباق را نشان می‌دهد.

۴. روش پیشنهادی

روند عملکرد رویکرد پیشـنهادی که ترکیب جدید ژنتیک-فازی نوع ۲ اسـت در شـکل ۶ ترسـیم شـده و مشـتمل بر هفت مرحله اصـلی می‌باشـــد. الگوریتم ژنتیک وظیفه تنظیم پارامترهای ۳ مرحله را برعهده داشـته و نتیجه مرحله شـشـم یعنی محاسـبه خطا مقدار تخمینی ارزیابی سـیسـتم توسـط رویکرد پیشـنهادی از مقدار واقعی بعنوان تابع برازش و تعیین کیفیت پارامترهای پیشـنهادی به منظور اعمال در فرایند بهبود الگوریتم ژنتیک در نظر گرفته می‌شـــود. در ادامه این بخش مراحل شـشـگانه رویکرد فازی پیشـنهادی تشـریح شده است.



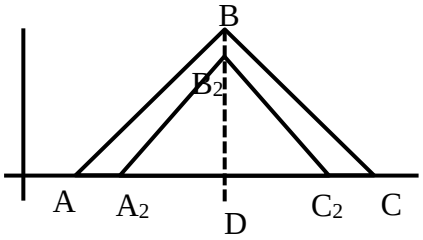
شکل (۶). روند عملکرد رویکرد پیشنهادی

مرحله اول: شناسایی شاخص‌های کیفی

در نخستین مرحله رویکرد پیشنهادی می‌بایست تمامی نیازمندی‌ها و شـاخص‌های کیفی برای ارزیابی سـیسـتم خود انطباق شـناسـایی شـوند. این شـاخص‌های کیفی از مدل‌های مک‌کال، بوهم و سـایر اسـتانداردهای موجود در این زمینه و همچنین از مقالات ارائه‌دهنده سیستم خود انطباق جمع‌آوری شده است. سپس متریک‌های مؤثر در بدست آوردن مقدار این شاخص‌ها مشخص می‌شوند. هر شاخص می‌تواند توسـط چند متریک مورد ارزیابی و محاسـبه قرار گیرد. از آن جهت که شـاخص‌ها و متریک‌های بسـیاری جهت ارزیابی کیفی وجود دارد، شاخص‌ها و متریک‌های موثر بر شاخص‌های کیفی مورد اســـتفاده در این مقاله باتوجه به جدول (۳) از مقالات ارائه‌دهنده سیستم خود انطباق جمع‌آوری شده است.

به منظور تعریف و نمایش شـــاخص‌ها و متریک‌ها از اعداد فازی نوع ۲ در رویکرد پیشـنهادی اسـتفاده می‌شـود. همانطور که در

شکل ۷ مشاهده می‌شود و قبلا نیز ذکر شده، تابع عضویت فازی نوع ۲ از دو سطح بالایی و پایینی تشکیل شده است. مقادیر سطح بالا و پایین از چهار مقدار تشکیل شده‌اند. مطابق با شکل، مقدار تابع عضویت سطح بالا برابر است با: (A,D,C,B) و مقدار سطح پایین برابر است با (A_2,D,C_2,B_2). همانطور که در شکل ۶ مشاهده می‌شود مقادیر مربوط به پارامترهای توابع عضویت فازی با استفاده از الگوریتم ژنتیک بدست می‌آیند.



شکل (۷). نمونه تابع عضویت مثلثی برای مقادیر فازی شاخص‌ها و متریک‌ها

مرحله دوم: تخصیص رتبه به شاخص

در این مرحله باید به شاخص‌های انتخاب شده یک رتبه اختصاص داده شود که میزان اهمیت آن را نمایش می‌دهد. بدین‌ترتیب، شاخص‌های کیفی مهم‌تر شناسایی می‌شوند. میزان اهمیت هر شاخص توسط مقادیر خیلی کم، کم، متوسط، زیاد و خیلی زیاد مشخص می‌شود که تنظیم بهینه پارامترهای توابع عضویت آنها توسط الگوریتم ژنتیک انجام می‌پذیرد. برای این مهم هر یک از پارامترهای توابع عضویت ۵ مقدار کیفی فوق را به عنوان یک ژن در ساختار کروموزم در نظر می‌گیریم. باتوجه به شکل ۷، برای کد کردن این توابع عضویت نیاز به ۳۵ ژن می‌باشد. هر ژن نیز با مقدار حقیقی کد می‌شود. نتیجه این مرحله، دو بردار ۴ بیتی است که مقدار رتبه شاخص را به شکل فازی بیان کرده و به صورت زیر نمایش داده می‌شود:

$$rateLow = [rlow1, rlow2, rlow3, rlow4]$$
$$rateUp = [rup1, rup2, rup3, rup4]$$

مرحله سوم: تخصیص رتبه به متریک

برای ارزیابی هر شاخص کیفی ممکن است چند متریک وجود داشته باشد، سیستم ممکن است با توجه به نوع عملیات خود به برخی از متریک‌ها اهمیت بیشتری دهد. به همین جهت وزن و تأثیر هر متریک برای ارزیابی شاخص، باید مشخص باشد. لذا در این مرحله به منظور تعیین میزان اهمیت متریک‌هایی که برای ارزیابی شاخص وجود دارند، به آنها رتبه تخصیص داده می‌شود. این امر می‌تواند در کاهش پیچیدگی محاسبات کمک نماید. در صورتی که یک شاخص تنها از یک متریک جهت ارزیابی استفاده کند، نیازی به تعیین رتبه نیست. جهت رتبه‌دهی به متریک‌ها نیز مانند مرحله قبل عمل می‌شود. یعنی هر متریک با توجه به مقادیر خیلی کم، کم، متوسط، زیاد و خیلی زیاد رتبه‌دهی می‌شوند. ۳۵ ژن بخش دوم یک کروموزم به تعیین مقادیر پارامترهای توابع عضویت این ۵ مقدار کیفی براساس نمایش عدد فازی نوع ۲ اختصاص می‌یابد.

نتیجه این مرحله دو ماتریس با چهار ستون و m سطر است. که m نشان دهنده تعداد متریک‌های مورد استفاده برای ارزیابی شاخص مذکور می‌باشد. یکی از این ماتریس‌ها به مقادیر سطح بالا و دیگری به مقادیر سطح پایین اشاره دارد که به ترتیب با rateup و ratelow نشان داده می‌شوند.

$$\text{rateup} = \begin{bmatrix} rup_{1,1} & rup_{1,2} & rup_{1,3} & rup_{1,4} \\ \vdots & \vdots & \vdots & \vdots \\ rup_{m,1} & rup_{m,2} & rup_{m,3} & rup_{m,4} \end{bmatrix}$$

$$\text{ratelow} = \begin{bmatrix} rlow_{1,1} & rlow_{1,2} & rlow_{1,3} & rlow_{1,4} \\ \vdots & \vdots & \vdots & \vdots \\ rlow_{m,1} & rlow_{m,2} & rlow_{m,3} & rlow_{m,4} \end{bmatrix}$$

مرحله چهارم: تخصیص وزن فازی به متریک

این مرحله مربوط به انتخاب وزن فازی مناسب برای هر متریک است. این وزن‌ها نیز توسط الگوریتم ژنتیک محاسبه می‌شوند. هر متریک دارای نوع وزن‌دهی مخصوص به خود است. اما به طور کلی می‌تواند شامل مقادیر برآورده نشدن، خیلی کم، کم، متوسط، زیاد و خیلی زیاد قرار گیرد. به منظور کد کردن مقادیر پارامترهای توابع عضویت این ۶ مقدار کیفی براساس ساختار فازی نوع ۲ به تعداد ۴۲ ژن در قالب بخش سوم یک کروموزم در نظر گرفته می‌شود.

خروجی این مرحله دو ماتریس است که مانند ماتریس رتبه حاوی ۴ ستون و mسطر است که m تعداد متریک‌های مؤثر در ارزیابی شاخص را نمایش می‌دهد. ماتریس upperWeightValue به مقادیر سطح بالا و ماتریس lowerWeightValue به مقادیر سطح پایین اشاره دارد.

$$upperWeightValue = \begin{bmatrix} wup_{1,1} & wup_{1,2} & wup_{1,3} & wup_{1,4} \\ \vdots & \vdots & \vdots & \vdots \\ wup_{m,1} & wup_{m.2} & wup_{m,3} & wup_{m,4} \end{bmatrix}$$

$$lowerWeightValue = \begin{bmatrix} wlow_{1,1} & wlow_{1,2} & wlow_{1,3} & wlow_{1,4} \\ \vdots & \vdots & \vdots & \vdots \\ wlow_{m,1} & wlow_{m.2} & wlow_{m,3} & wlow_{m,4} \end{bmatrix}$$

مرحله پنجم: محاسبه وزن فازی شاخص

با توجه به مقادیر بدست آمده در مرحله چهارم، برای هر شاخص یک وزن فازی نوع ۲ می‌شود. در این مرحله متریک‌ها و مقادیر آنها به عنوان ورودی و شاخص کیفی به عنوان خروجی در نظر گرفته شده است و مقادیر داده شده برای متریک‌ها، تعیین‌کننده مقدار شاخص کیفی هستند. مقدار بدست آمده برای شاخص کیفی نیز می‌تواند در گسترده برآورده نشده، خیلی کم، کم، متوسط، زیاد و خیلی زیاد باشد.

$$weightFactorUp = rateup.upperWeightValue \quad (۸)$$

برای ماتریس سطح پایین نیز به صورت زیر عمل می‌شود:

$$weightFactorLow = ratelow.lowerWeightValue \quad (۹)$$

لازم به ذکر است که در هر دو فرمول بالا از ضرب نقطه‌ای استفاده شده است. یعنی درآیه‌های متناظر در هر دو ماتریس با هم ترکیب می‌شوند.

مرحله ششم: غیرفازی‌سازی

در این مرحله میزان کیفی شاخص به مقدار کمی تبدیل می‌شود و بدین‌ترتیب مشخص می‌شود که تا چه اندازه سیستم توانسته شاخص کیفی مورد نظر را برآورده سازد. برای غیرفازی سازی در فازی نوع ۲، دو مرحله اساسی وجود دارد:

۱. کاهش نوع: تبدیل فازی نوع ۲ به فازی نوع ۱
۲. غیرفازی‌سازی توسط یکی از روش‌های مربوط به فازی نوع ۱

برای کاهش نوع مینیمم مقدار هر ستون از ماتریس‌های مربوط به وزن فازی بدست می‌آید و در نهایت یک ماتریس با ۱ سطر و چهار ستون ایجاد می‌شود. هر درآیه از این ماتریس نشان دهنده کمترین مقدار در ستون متناظر است. این عمل برای هر دو ماتریس weightFactorUp و weightFactorLow انجام می‌شود. بدین ترتیب وزن فازی شاخص مشخص می‌شود. سپس رتبه و میزان اهمیت این شاخص در این وزن ضرب می‌شود:

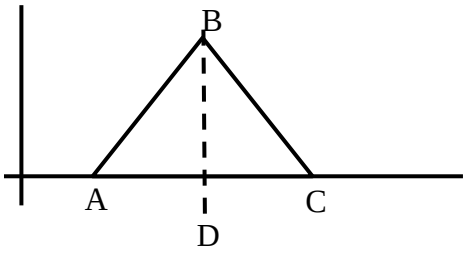
```
FactorUpValue=min (factorlowervalue, [], 1).*ratefactorLow;   (۱۰)
```

```
FactorLowValue =min (factoruppervalue, [], 1).*ratefactorUp;   (۱۱)
```

سپس از دو بردار فوق برای هر ستون مقدار ماکزیمم محاسبه می‌شود. در نهایت یک بردار با یک سطر و چهار ستون تولید می‌گردد. مانند:

$$Value = [A\ D\ C\ B] \quad (۱۲)$$

این بردار، یک مثلث را در نمودار نمایش می‌دهد که در شکل ۸ نشان داده شده است. مقدار سطح زیر این نمودار، مشخص کننده مقدار غیرفازی است که برای بدست آوردن مقدار آن، از انتگرال استفاده می‌شود.



شکل (۸) حاصل عملیات کاهش نوع

$$main\ value = \frac{\int_A^B \left(\frac{D-A}{B-A}\right) z\, dz + \int_B^C \left(\frac{C-D}{C-B}\right) z\, dz}{\int_A^B \left(\frac{D-A}{B-A}\right) dz + \int_B^C \left(\frac{C-D}{C-B}\right) dz} \quad (۱۳)$$

بدین ترتیب مقدار حاصل از عملیات غیرفازی سازی بدست می‌آید. این مقدار بین ۰ تا ۱.۲ است و هرچه مقدار بیشتر باشد به معنای آن است که شاخص مورد نظر بیشتر برآورده شده است.

مرحله هفتم: محاسبه خطا

از آنجایی که الگوریتم ژنتیک وظیفه انتخاب بهینه پارامترهای توابع عضویت مقادیر فازی رتبه‌های شاخص، رتبه‌های متریک و وزن‌های فازی متریک‌ها را توسط یک کرومزوم به طول ۱۱۲ ژن برعهده دارد، لازم است کیفیت هر کرومزوم در فرایند الگوریتم ژنتیک توسط یک تابع برازش محاسبه گردد. لذا خطای هر شاخص را از طریق تفاضل مقدار بدست آمده از عملیات غیرفازی سازی (رابطه ۱۳) بعنوان مقدار تخمین زده شده هر شاخص از مقدار واقعی بدست آورده و کمینه‌سازی مجموع آنها ($\sum|e|$) را بعنوان تابع برازش در نظر می‌گیریم. یعنی هرچه مقدار خروجی تخمین‌زده شده توسط سیستم فازی پیشنهادی به مقدار واقعی که توسط فرد خبره داده می‌شود، نزدیک‌تر باشد، برازش بیشتر خواهد بود. برای این منظور از مقادیری که قبلا توسط فرد خبره داده شده استفاده می‌شود.

۵. پیاده سازی و ارزیابی

این بخش به منظور پیاده‌سازی و ارزیابی روش پیشنهادی مشتمل بر سه بخش اصلی می‌باشد. بخش اول به معرفی داده مورد استفاده یعنی یک سیستم کنترل ترافیک می‌پردازد. بخش دوم به تشریح پیاده‌سازی روش پیشنهادی و ابزارهای مورد استفاده در پیاده‌سازی آن اشاره دارد. بخش سوم به ارزیابی روش پیشنهادی بر روی سیستم کنترل ترافیک تمرکز دارد.

۵-۱. داده مورد استفاده

سیستم کنترل ترافیک یکی از معمول‌ترین انواع سیستم های خود انطباق است که تحقیقات بسیاری در حوزه آن انجام گرفته است. جزئیات اطلاعات سایر سیستم‌های خود انطباق موجود در تحقیقات دیگر یا در دسترس نبودند یا اطلاعات کمی در مورد آنها وجود داشت و یا اینکه نمی‌توانستند شاخص‌های کیفی زیادی را شامل شوند. به همین دلیل، در خصوص ارزیابی، یک سیستم کنترل ترافیک به نام InSync انتخاب شده است. این سیستم، حاوی چند دوربین است که در مکان‌های خاصی از خیابان نصب می‌شوند. دوربین‌ها به عنوان حسگر یا سنسور عمل می‌کنند. اطلاعات ترافیک و ازدحام خودروها را جمع آوری کرده و سپس این اطلاعات را برای سیستم مرکزی ارسال می‌کنند. با توجه به تصاویر ارسال شده و دیگر اطلاعات موجود، سیستم مرکزی تصمیم می‌گیرد که چراغ راهنمایی و رانندگی را برای کدام مسیر سبز و برای کدام مسیر قرمز کند. در نتیجه سیستم فوق، یک سیستم حاوی بخش‌های سخت افزاری و نرم افزاری بسیاری است. برای ارزیابی این سیستم از مستنداتی که حاوی اطلاعات جامع به همراه ارزیابی‌ها و عملکردهای مربوط به آن بوده استفاده شده است. InSync دو جنبه مهم دارد:

- به طور خودکار توسط سیگنال‌های محلی تنظیم می‌شود.
- با توجه به اطلاعات سیگنال‌ها رانندگان را نسبت به ترافیک و مسائل ترافیکی در جاده آگاه می‌سازد.

۵-۲. پیاده‌سازی

بر طبق روش پیشنهادی که بیان شد، پرسشنامه یکی از موارد مهم و کلیدی در این روش محسوب می‌گردد. با توجه به شاخص‌ها و متریک‌ها، می‌توان یک دسته از شاخص‌های کیفی را به عنوان متریک‌های مهم در ارزیابی سیستم‌های خود انطباق مطرح کرد. با این حال، بسته به نوع سیستم و نیازمندی‌های آن، برخی از شاخص‌های کیفی دارای اهمیت بیشتری هستند. بنابراین مشخص کردن رتبه شاخص‌های کیفی می‌تواند به منزله اولویت و اهمیت آن شاخص در سیستم باشد.

رتبه و مقدار فازی هر شاخص کیفی توسط پرسشنامه مشخص می‌گردد. جداول (۱ پیوست) و (۲ پیوست) لیست ۳۲ شاخص کیفی و ۴۷ متریک موثر در ارزیابی سیستم و نحوه رتبه‌بندی آنها نمایش داده شده و پرسشنامه‌های مرتبط میزان رتبه این شاخص‌ها و متریک‌ها نیز در جداول (۳ پیوست) و (۴ پیوست) ارائه شده‌اند. در این جداول ستون No به معنی بی اهمیت بودن، ستون VL به معنای خیلی کم اهمیت، ستون L به معنای کم اهمیت است. همچنین ستون M اهمیت متوسط، ستون H اهمیت زیاد و ستون VH اهمیت خیلی زیاد را نشان می‌دهد.

در ادامه به منظور محاسبه وزن فازی شاخص‌ها، لازم است به ۴۷ متریک موثر در ارزیابی سیستم وزن‌های فازی اختصاص یابد که در جدول (۵ پیوست) پرسشنامه انجام این مهم نمایش داده شده است. در این جدول ستون No به معنی برآورده نشدن، ستون VL به معنای خیلی کم برآورده شدن، ستون L به معنای کم برآورده شدن است. همچنین ستون M برآورده شدن در حد متوسط، ستون H برآورده شدن در حد زیاد و ستون VH برآورده شدن در حد خیلی زیاد را نشان می‌دهد.

جدول (۴). مقادیر رتبه بندی شاخص‌های کیفی

مقدار کیفی	محدوده در جدول	
	سطح بالایی	سطح پایینی
خیلی کم	[0 0 0.32 0.95]	[0 0 0.2 0.45]
کم	[0.1 0.44 0.75 1]	[0.32 0.44 0.61 0.5]
متوسط	[0.4 0.67 0.89 1]	[0.49 0.67 0.8 0.5]
زیاد	[0.81 0.9 1.1 1]	[0.78 0.9 1 0.5]
خیلی زیاد	[0.9 1 1.2 1]	[0.96 1 1.06 0.5]

جدول (۵). مقادیر رتبه بندی متریک‌ها

مقدار کیفی	محدوده در جدول	
	سطح بالایی	سطح پایینی
خیلی کم	[0 0 0.49 0.92]	[0 0 0.27 0.44]
کم	[0.12 0.39 0.61 0.97]	[0.29 0.39 0.55 0.5]
متوسط	[0.47 0.7 0.89 1]	[0.45 0.7 0.82 0.54]
زیاد	[0.73 0.9 1.12 1]	[0.81 0.9 1 0.54]
خیلی زیاد	[0.9 1 1.2 1]	[0.93 1 1.1 0.5]

جدول (۶). مقادیر وزن فازی برای متریک‌ها

وزن فازی	محدوده در جدول	
	سطح بالایی	سطح پایینی
برآورده نشدن	[0 0 0.09 1]	[0 0 0.04 0.5]
خیلی کم	[0 0.1 0.41 1]	[0.04 0.1 0.36 0.5]
کم	[0.12 0.4 0.72 1]	[0.3 0.4 0.57 0.5]
متوسط	[0.44 0.7 0.9 1]	[0.54 0.7 0.81 0.5]
زیاد	[0.72 0.9 1.12 1]	[0.82 0.9 1 0.5]
خیلی زیاد	[0.9 1 1.2 1]	[0.95 1 1.075 0.5]

جهت پیاده‌سازی سیستم پیشنهادی از نرم‌افزار متلب ۲۰۲۰ استفاده شده است. برای اجرای برنامه نیز از سیستمی با ویندوز ۷، core i5 و RAM 8GB استفاده شده است. به منظور تنظیم بهینه مقادیر رتبه‌بندی شاخص‌های کیفی، متریک‌ها و وزن‌های فازی برای متریک‌ها از نسخه کلاسیک الگوریتم ژنتیک استفاده شد که تعداد جمعیت اولیه ۱۰۰ کرومزوم، تعداد تکرار ۱۰۰۰ نسل و نرخ جهش و بازترکیب به ترتیب ۰.۲ و ۰.۸ استفاده شده است. در جدول (۴) پارامترهای توابع عضویت فازی بدست آمده توسط الگوریتم ژنتیک برای رتبه بندی شاخص‌ها نشان داده شده است. همچنین برای رتبه‌دهی به متریک‌ها مقادیر جدول (۵) و مقادیر وزن فازی برای هر متریک نیز در جدول (۶) بدست آمده است.

۲-۵. ارزیابی

روش پیشنهادی را می‌توان از دو جهت با سایر روش‌ها و مقالات موجود مورد مقایسه قرار داد:

۱- شاخص‌های کیفی و متریک‌های مورد استفاده

۲- چارچوب و روش کار

از جهت شاخص‌های کیفی مورد استفاده، این شاخص‌ها با سایر مقالات و مدل‌های کیفی در جدول (۷) مورد مقایسه قرار گرفته است. بطور آشکار این جدول نشان می‌دهد که روش پیشنهادی جامعیت بیشتری نسبت به سایر مدل‌ها و مقالات موجود دارد و طی آن سعی شده تا سیستم‌های خودانطباق از دو بُعد شامل بُعد نرم‌افزاری این سیستم و بُعد خود انطباق آنها مورد مقایسه قرار گیرد. این در حالی است که اکثر تحقیقات موجود تنها به یک بعد می‌پردازند.

از طرفی اگر سیستم Insync را تنها با مدل‌های کیفی مطرح در جدول (۷) مورد ارزیابی قرار دهیم، یا به عبارت بهتر از شاخص‌های کیفی موجود استفاده کنیم، نمی‌توان به طور جامع این سیستم را مورد ارزیابی قرار داد. لذا به منظور جامعیت بخشی بیشتر به فرایند ارزیابی روش پیشنهادی از نمودار رادار استفاده کرده‌ایم. برای این منظور اطلاعات سیستم Insync با توجه به تنها شاخص‌های کیفی مطرح شده در یک مقاله، مورد ارزیابی قرار می‌دهیم و روش پیشنهادی را تنها با همان شاخص‌های کیفی اجر می‌کنیم. برای اینکار سه مقاله [۵۲]، [۲] و [۵۳] انتخاب شدند. نتیجه کار در نمودارهای رادار در شکل (۹) تا (۱۱) قابل مشاهده است. نمودار رادار روش پیشنهادی نیز در شکل (۱۲) آمده است.

همانطور که نمودارهای رادار (۹) تا (۱۲) و جدول (۷) نمایش می‌دهند، روش پیشنهادی از جامعیت بیشتری برخوردار است و شاخص‌هاس کیفی متعددی را در بر می‌گیرد. اما نکته حائز اهمیت در ارزیابی کیفیت نرم‌افزار (چه خود انطباق باشد و چه نباشد) این است که تمام معیارها و شاخص‌های کیفی برای ارزیابی دارای اهمیت نیستند. به عبارت بهتر، همه این معیارها برای ارزیابی کیفیت نرم افزار به کار گرفته نمی‌شوند و در سیستم‌های مختلف دارای اولویت و اهمیت متفاوتی هستند. درواقع، شاخص‌های کیفی، دارای قطعیت نیستند و مهم بودن یک شاخص کیفی در یک یا چند سیستم نرم‌افزاری به منزله اهمیت آن در تمام سیستم‌های نرم‌افزاری نیست. اگرچه، در سیستم‌های خود انطباق مجموعه‌ای از شاخص‌های کیفی هستند که ویژگی و اساس این گونه از سیستم‌ها را فراهم می‌آورند (مانند پایداری، شاخص‌های خود-*، حساسیت، قابلیت پیشگویی، کارایی، هزینه، نفوذپذیری، تحمل به خطا و ...)، اما این شاخص‌ها بسته به نوع و عملکرد سیستم می‌توانند اهمیت متفاوتی در سیستم‌های مختلف داشته باشند. درنتیجه لازم است مکانیزم و راهکاری برای در نظر گرفتن میزان اهمیت شاخص‌های کیفی در چارچوب و روش کار در نظر گرفت.

از دیدگاه چارچوب و روش کار نیز، جدول شماره (۸) روش‌های بکار گرفته شده در مقالات موجود در حیطه ارزیابی کیفیت سیستم نرم‌افزاری خود انطباق نشان می‌دهد. با توجه به جدول (۸)، مشاهده می‌شود که بیشتر مقالات توجهی به موضوع میزان اهمیت و اولویت شاخص‌های کیفی نسبت به یکدیگر نداشتند. استفاده از یک الگوی ثابت در ارزیابی سیستم‌های نرم‌افزاری می‌تواند سبب نادقیق شدن نتایج ارزیابی شود.

روش پیشنهادی ابتدا یک چک لیست از شاخص‌های کیفی موثر در ارزیابی سیستم خود انطباق فراهم می‌آورد. سپس این امکان را می‌دهد تا ارزیاب با توجه به عملکرد سیستم یا نظر ذینفعان سیستم، شاخص‌های کیفی مهم را مشخص کند و اولویت آنها را تعیین نماید. این امر می‌تواند در ارزیابی بهتر سیستم تاثیر بسزایی داشته باشد.

علاوه‌براین، اکثر مقالات از فرمول‌های مشخص شده برای شاخص‌های کیفی استفاده کرده‌اند. حال اینکه، تمامی شاخص‌های کیفی را نمی‌توان با استفاده از فرمول و مدل‌های ریاضی بدست آورد. در بسیاری از موارد یافتن یک فرمول برای شاخص کیفی بسیار دشوار و گاه ناممکن است. برای مثال برای تعیین مقدار شاخص کیفی قابلیت استفاده، بدست آوردن فرمول تقریباً ناممکن است.

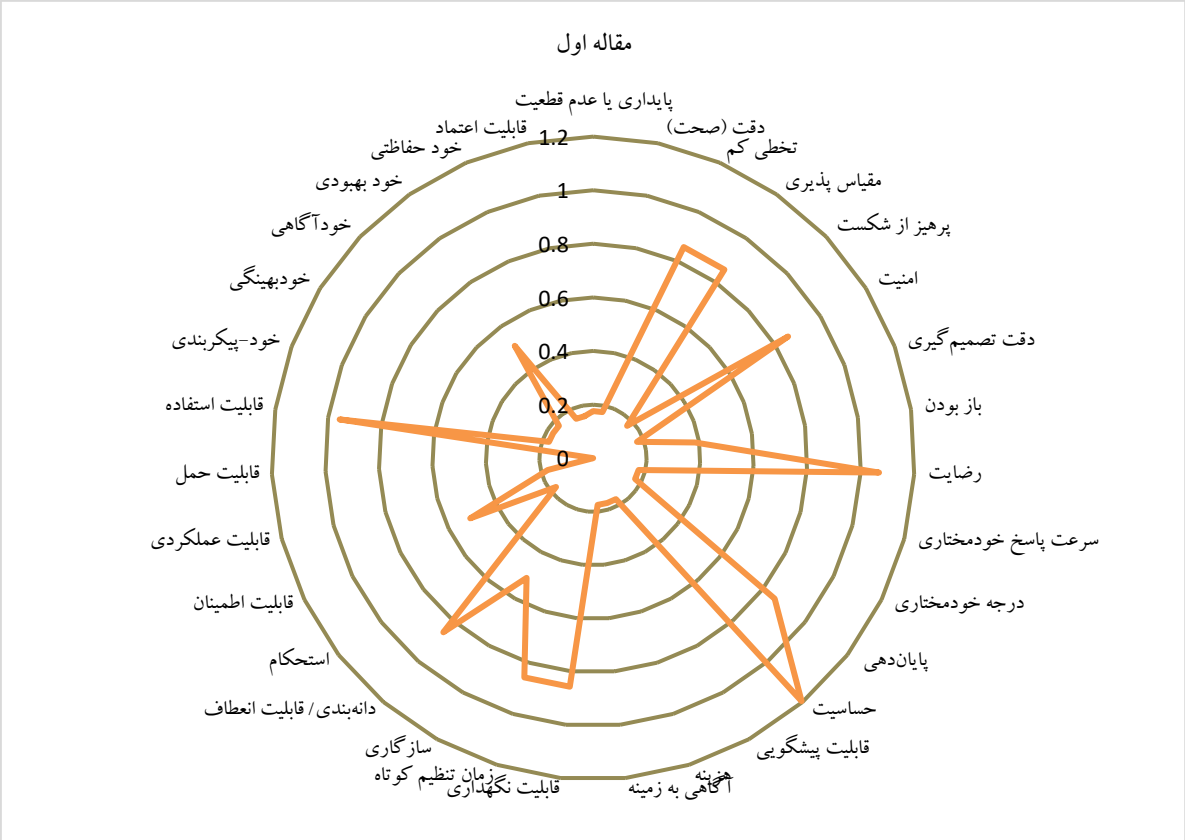
از طرف دیگر، گاه بدست آوردن مقادیر لازم جهت انجام محاسبات دشوار و یا زمان‌بر است. روش مبتنی بر سناریو نیز یکی از روش‌های پرکاربرد در ارزیابی است، اما زمانی که کیفیت و شاخص‌های کیفی مطرح می‌شوند، این روش نیز نمی‌تواند کارایی مناسبی داشته باشد، زیرا تولید سناریو برای ارزیابی کیفی دشوار و یا گاه ناممکن است. روش لایه‌ای را نیز نمی‌توان برای تمام شاخص‌های کیفی بسط داد.

جدول (۷) مقایسه شاخص‌های کیفی در روش پیشنهادی با سایر مقالات و مدل‌های کیفی

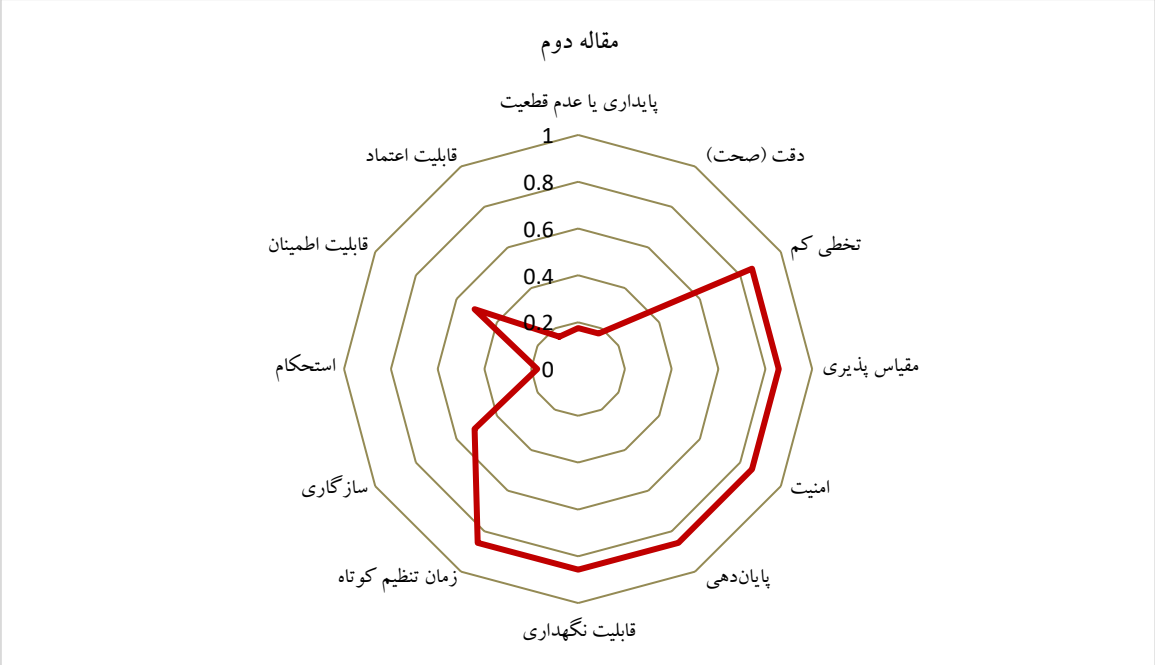
مدل‌های کیفی/ تحقیقات / شاخص‌های کیفی/ معیارها	MCCall	بوهم	ISO 9126	[۵۴]	[۳]	[۴۲]	[۵۵]	[۵۳]	[۳]	[۵۲]	روش پیشنهادی
پایداری یا عدم قطعیت							✓	✓		✓	✓
دقت (صحت)	✓	✓	قابلیت نگهداری	✓			✓			✓	✓
قابلیت اصلاح		✓	قابلیت نگهداری								✓
تخطی کم							✓			✓	✓
مقیاس‌پذیری				✓			✓			✓	✓
پرهیز از شکست								✓	✓	✓	✓
امنیت				✓			✓			✓	✓
دقت تصمیم‌گیری										✓	✓
باز بودن										✓	✓
رضایت					✓				✓	✓	✓
سطح خودمختاری											✓
سرعت پاسخ خودمختاری										✓	✓
درجه خودمختاری								✓	✓	✓	✓
پایان‌دهی							✓			✓	✓
حساسیت								✓		✓	✓
قابلیت پیشگویی										✓	✓
هزینه								✓	✓	✓	✓
آگاهی به زمینه						✓				✓	✓
ثبات											✓
قابلیت نگهداری	✓	✓	✓	✓	✓		✓			✓	✓
زمان تنظیم کوتاه							✓			✓	✓
سازگاری							✓			✓	✓
دانه‌بندی/ قابلیت انعطاف	✓	✓		✓				✓		✓	✓
استحکام							✓			✓	✓
قابلیت اطمینان							✓			✓	✓
قابلیت عملکردی			✓	✓	✓					✓	✓
قابلیت حمل	✓	✓	✓	✓	✓					✓	✓
قابلیت استفاده	✓	✓		✓	✓					✓	✓
خود-پیکربندی						✓				✓	✓
خودبهینگی						✓				✓	✓
خودآگاهی						✓				✓	✓
خود بهبودی						✓				✓	✓
خود حفاظتی						✓				✓	✓
قابلیت اعتماد	✓	✓	✓	✓	✓		✓			✓	✓
قابلیت آزمایش	✓	✓	قابلیت نگهداری							✓	قابلیت نگهداری
قابلیت استفاده مجدد	✓	✓								✓	قابلیت اصلاح
قابلیت همکاری درونی	✓									✓	قابلیت عملکردی
بهره‌وری	✓	✓	✓	✓	✓					✓	قابلیت استفاده / قابلیت پیشگویی/ خودبهبودی

جدول ادامه دارد ...

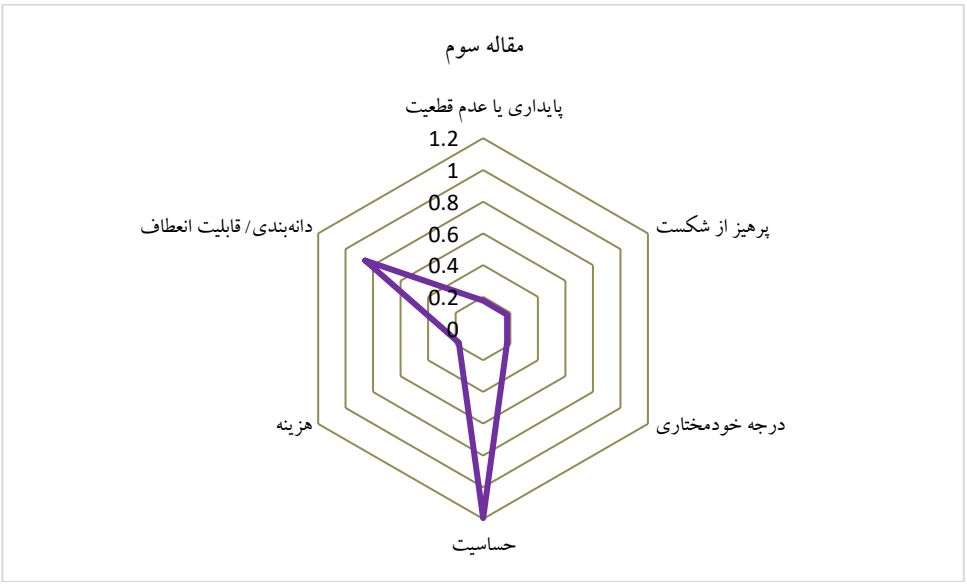
مدل‌های کیفی/ تحقیقات / شاخص‌های کیفی/ معیارها	MCall	بوهم	ISO 9126	[۵۴]	[۳]	[۴۲]	[۵۵]	[۵۳]	[۲]	[۵۲]	روش پیشنهادی
یکپارچگی	✓	✓	✓				✓			✓	امنیت / سازگاری/ پایداری
قابلیت درک		✓								✓	قابلیت استفاده
کارایی				✓			✓			✓	هزینه/ صحت /مقیاس‌پذیری/ پایان دهی/ تخطی کم / زمان تنظیم کوتاه / پایداری
قابلیت دسترسی				✓			✓			✓	خودبهبودی/ قابلیت اطمینان/ امنیت / استحکام
سودمندی					✓					✓	قابلیت استفاده
ایمنی							✓		✓	✓	استحکام/ پرهیز از شکست/ پایداری
جداناپذیری							✓				
کیفیت خدمات								✓			
قابلیت انطباق								✓			
زمان انطباق/ زمان واکنش								✓	✓	✓	کارایی
انطباق بهینه									✓		سودمندی



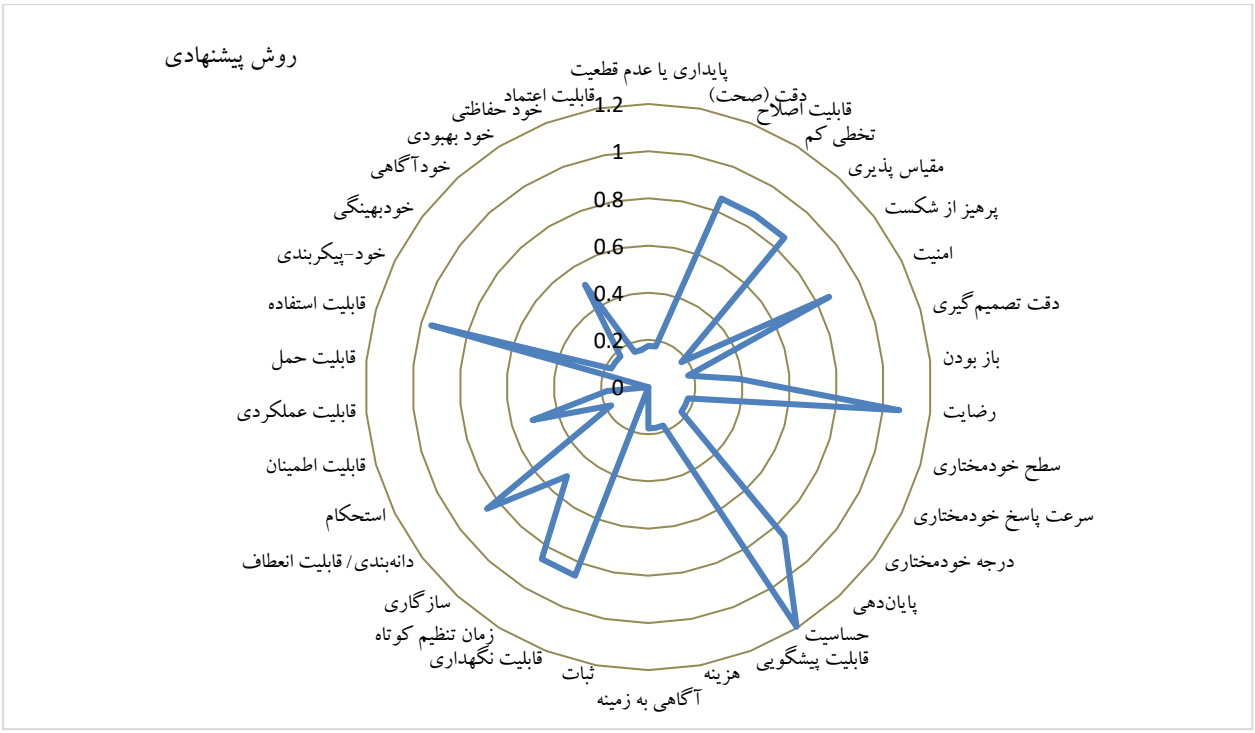
شکل(۹) نمودار رادار شاخص‌های کیفی در [۵۲]



شکل(۱۰) نمودار رادار شاخص‌های کیفی در [۲]



شکل(۱۱) نمودار رادار شاخص‌های کیفی در [۵۳]



شکل(۱۲) نمودار رادار شاخص‌های کیفی در روش پیشنهادی

جدول(۸) مقایسه روش کار مقالات حوزه ارزیابی کیفیت سیستم خود انطباق با روش پیشنهادی

معیارها / مقالات	میزان شاخص‌های کیفی مورد استفاده	استفاده از رتبه و اولویت	روش کار
[55]	تعداد محدودی از شاخص‌های کیفی را در نظر گرفته است.	×	استفاده از فرمول
[۲]	شش شاخص هزینه، پرهیز از شکست، درجه خودمختار بودن، انطباق بهینه، زمان انطباق/واکنش، ایمنی و پایداری	×	استفاده از فرمول
[52]	مجموعه کاملی از شاخصهای کیفی در سیستم های خود انطباق	✓	استفاده از فرمول
[۳۵]	قابلیت دسترسی	×	استفاده از سناریو
[۵۶]	اعتبار و قابلیت اعتماد داده های ورودی، واسط‌های مورد استفاده با کاربر و محیط، تصمیمات انطباق، تراکنش‌های سیستم	×	استفاده از روش لایه‌ای
[۶]	مجموعه ای از شاخص‌های کیفی	×	استفاده از فرمول
[۳۱]	مجموعه ای از شاخص‌های کیفی	×	استفاده از فرمول
روش پیشنهادی	مجموعه ای از شاخص‌های کیفی	✓	استفاده از منطق فازی

* در جدول ۸ علامت ✓ به معنای وجود مکانیزم اولویت‌دهی و × به معنای عدم وجود این مکانیزم است.

استفاده از منطق فازی جهت ارزیابی کیفیت نرم‌افزار می‌تواند فرآیند ارزیابی را تسهیل نماید. همانطور هم که قبلاً ذکر شد، شاخص‌های کیفی نرم‌افزار و همچنین میزان اهمیت آنها یک مقدار کیفی است که در زمان ارزیابی کیفیت نرم‌افزار باید کمی‌سازی شود. هدف از منطق فازی، کمّی‌سازی مقادیر کیفی است. از این‌رو، روش پیشنهادی می‌تواند بهتر از سایر روش‌ها عمل کند.

۶. نتیجه گیری و پیشنهادات

در این مقاله هدف ارائه یک رویکرد برای ارزیابی کیفیت سیستم‌های خود انطباق است. سیستم‌های خود انطباق به سیستم‌هایی اطلاق می‌گردد که رفتار خود را بر طبق اطلاعات محیطی و یا زمینه خود تغییر می‌دهند. این دسته از سیستم‌ها، عموماً بدون دخالت انسان و یا با حداقل دخالت انسان کار می‌کنند. مقاله حاضر به بررسی شاخص‌های کیفی و در واقع کیفیت نرم‌افزار خود انطباق در یک سیستم کنترل ترافیک به نام InSync می‌پردازد. به همین دلیل، ابتدا، شاخصهای اصلی جهت ارزیابی سیستم‌های خود انطباق از مقالات و منابع مختلف دردسترس برای این سیستم استخراج شده و سپس شیوه اندازه‌گیری این شاخصها معرفی گردید.

شاخص‌های کیفی، دارای مقادیر کیفی هستند، بنابراین اندازه‌گیری آنها دشوار و کمی‌سازی آنها سخت است. منطق فازی به علت توانایی کار با مقادیر کیفی، و مقادیر دارای ابهام بعنوان گزینه مناسب برای اندازه‌گیری کیفیت نرم‌افزار انتخاب شده و فازی نوع ۲ به علت ویژگی‌های متمایزی که نسبت به فازی نوع ۱ داشته و توانایی رفع عدم قطعیت را دارد، بعنوان نوآوری این پژوهش در ارزیابی کیفیت نرم‌افزار خودانطباق مد نظر قرار گرفت.

این مقاله، علاوه بر طرح یک روش برای ارزیابی سیستم خود انطباق مبتنی بر الگوریتم ژنتیک و فازی نوع ۲، یک مجموعه از شاخص‌های کیفی مؤثر در آن را نیز جمع‌آوری کرده است که می‌تواند به عنوان یک چک لیست در زمان طراحی این دسته از سیستم‌ها مورد استفاده قرار گیرد. با این حال هنوز مراحل و روش‌هایی جهت بهبود طرح پیشنهادی جاری وجود دارد. از جمله اینکه، روش پیشنهادی، تنها کیفیت سیستم پیشنهادی را می‌سنجد و بر عملکرد این دسته از سیستم‌ها توجه ندارد. برخلاف تحقیقات و مقالات دیگر که بر عملکرد سیستم توجه دارند تا کیفیت آن. بعنوان یکی از مسیرهای تحقیقاتی باز در پیشروی این مقاله می‌توان به طراحی یک مکانیزم ارزیابی پیشنهادی که هم به کیفیت و هم به عملکرد سیستم‌های خود انطباق بپردازد، اشاره نمود.

دوم اینکه، روش پیشنهادی برای ارزیابی کیفیت سیستم خود انطباق از تابع عضویت مثلثی استفاده کرده است، اما بررسی دیگر توابع عضویت در این راستا بعنوان دومین مسیر باز تحقیقات آتی می‌تواند مشخص نماید که آیا تابع عضویت مثلثی بهتر انتخاب بوده یا خیر.

۷. مراجع

[1] F. D. Macías-Escrivá, R. Haber, R. Del Toro, and V. Hernandez, "Self-adaptive systems: A survey of current approaches, research challenges and applications," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7267-7279, 2013.

[2] F. Kneer and E. Kamsties, "A Framework for Prototyping and Evaluating Self-adaptive Systems - A Research Preview," 2016.

[3] S. Sucipto and R. S. Wahono, "A Systematic Literature Review of Requirements Engineering for Self-Adaptive Systems," *Software Engineering & Applications*, vol. 1, no. 1, 2015.

[4] M. Abufouda, "A Framework for Enhancing Performance And Handling Run-Time Uncertainty in Self-Adaptive Systems," *Software Engineering & Applications*, vol. 5, no. 1, 2014.

[5] M. Salehie and L. Tahvildari, "Towards a goal-driven approach to action selection in self-adaptive software," *Software: Practice and Experience*, vol. 42, no. 2, pp. 211-233, 2012.

[6] C. Raibulet, F. Arcelli Fontana, R. Capilla, and C. Carrillo, "An Overview on Quality Evaluation of Self-Adaptive Systems," 2016.

[7] J. D. Paraiba and L. E. G. Martins, "PERSA: a requirements specification process for self-adaptive systems based on fuzzy logic and NFR-framework," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 25, no. 01, pp. 145-178, 2017.

[8] J. A. McCall, P. K. Richards, and G. F. Walters, "Factors in software quality. volume i. concepts and definitions of software quality," DTIC Document, 1977.

[9] S. Valenti, A. Cucchiarelli, and M. Panti, "Computer based assessment systems evaluation via the ISO9126 quality model," *Journal of Information Technology Education*, vol. 1, no. 3, pp. 157-175, 2002.

[10] B. W. Boehm, J. R. Brown, and M. Lipow, "Quantitative evaluation of software quality," in *Proceedings of the 2nd international conference on Software engineering*, 1976: IEEE Computer Society Press, pp. 592-605.

[11] P. Berander *et al.*, "Software quality attributes and trade-offs," *Blekinge Institute of Technology*, 2005.

[12] C.-W. Chang, C.-R. Wu, and H.-L. Lin, "Integrating fuzzy theory and hierarchy concepts to evaluate software quality," *Software Quality Journal*, vol. 16, no. 2, pp. 263-276, 2008.

[13] P. HOLECEK and J. TALAŠOVÁ, "FuzzME: a new software for multiple-criteria fuzzy evaluation," *Acta Universitatis Matthiae Belii ser. Mathematics*, vol. 16, pp. 35-51, 2010.

[14] J. S. Challa, A. Paul, Y. Dada, V. Nerella, P. R. Srivastava, and A. P. Singh, "Integrated Software Quality Evaluation: A Fuzzy Multi-Criteria Approach," *JIPS*, vol. 7, no. 3, pp. 473-518, 2011.

[15] E. Letier, D. Stefan, and E. T. Barr, "Uncertainty, risk, and information value in software requirements and architecture," in *Proceedings of the 36th International Conference on Software Engineering*, 2014, pp. 883-894.

[16] F. Haryana, "Software Quality Evaluation using Fuzzy Multi Criteria Decision Method," 2015.

[17] A. Mansoor, D. Streitferdt, and F.-F. Füßl, "Fuzzy Based Evaluation of Software Quality Using Quality Models and Goal Models."

[18] M. Kara, O. Lamouchi, and A. Ramdane-Cherif, "Ontology software quality model for fuzzy logic evaluation approach," *Procedia Computer Science*, vol. 83, pp. 637-641, 2016.

[19] A. S. Abdygalievich, A. Barlybayev, and K. B. Amanzholovich, "Quality evaluation fuzzy method of automated control systems on the LMS example," *IEEE Access*, vol. 7, pp. 138000-138010, 2019.

[20] U. Dayanandan and V. Kalimuthu, "A fuzzy analytical hierarchy process (FAHP) based software quality assessment model: maintainability analysis," *International Journal of Intelligent Engineering and Systems*, vol. 11, no. 4, pp. 88-96, 2018.

[21] D. Manikavelan and R. Ponnusamy, "Software quality analysis based on cost and error using fuzzy combined COCOMO model," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-11, 2020.

[22] N. X. Thao and S.-Y. Chou, "Novel similarity measures, entropy of intuitionistic fuzzy sets and their application in software quality evaluation," *Soft Computing*, pp. 1-12, 2022.

[23] A. Barzegar, "Measuring software quality Product based on Fuzzy Inference System techniques in ISO standard," 2021.

[24] V. Singh, V. Kumar, and V. Singh, "A hybrid novel fuzzy AHP-TOPSIS technique for selecting parameter-influencing testing in software development," *Decision Analytics Journal*, vol. 6, p. 100159, 2023.

[25] A. O. de Sousa, C. I. Bezerra, R. M. Andrade, and J. M. Filho, "Quality evaluation of self-adaptive systems: Challenges and opportunities," in *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, 2019, pp. 213-218.

[26] L. E. Sanchez, J. A. Diaz-Pace, A. Zunino, S. Moisan, and J.-P. Rigault, "An approach based on feature models and quality criteria for adapting component-based systems," *Journal of Software Engineering Research and Development*, vol. 3, pp. 1-30, 2015.

[27] E. Serral, P. Sernani, and F. Dalpiaz, "Personalized adaptation in pervasive systems via non-functional requirements," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 6, pp. 1729-1743, 2018.

[28] R. MoeinFar and A. A. Barforoush, "Using models at run-time to measure quality of SAS in the large-scale software systems," in *2017 9th International Conference on Information and Knowledge Technology (IKT)*, 2017: IEEE, pp. 99-103.

[29] R. Edwards and N. Bencomo, "DeSiRE: further understanding nuances of degrees of satisfaction of non-functional requirements trade-off," in *Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems*, 2018, pp. 12-18.

[30] C. Raibulet, "Hints on quality evaluation of self-systems," in *2014 IEEE Eighth International Conference on Self-Adaptive and Self-Organizing Systems*, 2014: IEEE, pp. 185-186.

[31] E. Kaddoum, C. Raibulet, J.-P. Georgé, G. Picard, and M.-P. Gleizes, "Criteria for the evaluation of self-* systems," in *Proceedings of the 2010 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems*, 2010, pp. 29-38.

[32] C. I. Bezerra, R. M. Andrade, J. M. Monteiro, and D. Cedraz, "Aggregating measures using fuzzy logic for evaluating feature models," in *Proceedings of the 12th International Workshop on Variability Modelling of Software-Intensive Systems*, 2018, pp. 35-42.

[33] R. Wohlrab, J. Cámara, D. Garlan, and B. Schmerl, "Explaining quality attribute tradeoffs in automated planning for self-adaptive systems," *Journal of Systems and Software*, vol. 198, p. 111538, 2023.

[34] S. Malik, M. A. Naqvi, and L. Moonen, "CHESS: A Framework for Evaluation of Self-adaptive Systems based on Chaos Engineering," *arXiv preprint arXiv:2303.07283*, 2023.

[35] A. Parvizi-Mosaed, S. Moaven, J. Habibi, and A. Heydarnoori, "Towards a Tactic-Based Evaluation of Self-Adaptive Software Architecture Availability," in *SEKE*, 2014, pp. 168-173.

[36] Q. Yang, J. Lü, J. Li, X. Ma, W. Song, and Y. Zou, "Toward a fuzzy control-based approach to design of self-adaptive software," in *Proceedings of the Second Asia-Pacific Symposium on Internetware*, 2010: ACM, p. 15.

[37] W. Min, Z. Jun, and Z. Wei, "The application of fuzzy comprehensive evaluation method in the software project risk assessment," in *Proceedings of the 2017 International Conference on Management Engineering, Software Engineering and Service Sciences*, 2017: ACM, pp. 76-79.

[38] Y. Wang, J. Corey, Y. Lao, K. Henrickson, and X. Xin, "Criteria for the Selection and Application of Advanced Traffic Signal Systems," 2013.

[39] A. Stevanovic and P. M. Zlatkovic, "Comparative Evaluation of InSync and Time-of-Day Signal Timing Plans under Normal and Varied Traffic Conditions," ed: February, 2013.

[40] M. KETABDARI, "Analysis of adaptive traffic control systems and design of a decision support system for better choice," 2013.

[41] M. Selinger and L. Schmidt, "Adaptive traffic control systems in the united states: Updated summary and comparison," *HDR Engineering*, 2010.

[42] M. Salehie and L. Tahvildari, "Self-Adaptive Software: Landscape and Research Challenges," *ACM Transactions on Autonomous and Adaptive Systems*, 2009.

[43] S. Elkins and G. Niehus, "Insync adaptive traffic control system for the veterans memorial hwy corridor on long island, ny," 2012.

[44] Y. Brun *et al.*, "Engineering Self-Adaptive Systems through Feedback Loops," *Software engineering for self-adaptive systems*, vol. 5525, pp. 48-70, 2009.

[45] M. Ravindranathan and R. Leitch, "Heterogeneous intelligent control systems," *IEE Proceedings-Control Theory and Applications*, vol. 145, no. 6, pp. 551-558, 1998.

[46] M. Naqvi, "Claims and supporting evidence for self-adaptive systems–A literature review," ed, 2012.

[47] R. Almeida and M. Vieira, "Changeloads for resilience benchmarking of self-adaptive systems: a risk-based approach," in *Dependable Computing Conference (EDCC), 2012 Ninth European*, 2012: IEEE, pp. 173-184.

[48] S. M. Abuelenin, "Decomposed interval Type-2 fuzzy systems with application to inverted pendulum," in *Engineering and Technology (ICET), 2014 International Conference on*, 2014: IEEE, pp. 1-5.

[49] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 fuzzy logic systems," *IEEE transactions on Fuzzy Systems*, vol. 7, no. 6, pp. 643-658, 1999.

[50] م. ذكري, "مروري بر and ح. م. فراهانى, ج. عسگري منطق فازي نوع- ۲: از پيدايش تا كاربرد," محاسبات نرم, vol. 3, 1392.

[51] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems," in *Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on*, 2000, vol. 1: IEEE, pp. 328-333.

[52] A. Farahani, E. Nazemi, G. Cabri, and A. Rafizadeh, "An evaluation method for Self-Adaptive systems," in *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*, 2016: IEEE, pp. 002814-002820.

[53] J. A. McCann and M. C. Huebscher, "Evaluation issues in autonomic computing," in *International Conference on Grid and Cooperative Computing*, 2004: Springer, pp. 597-608.

[54] Danny Weyns, M. Usman Iftikhar, Didac Gil de la Iglesia, and T. Ahmad, "A Survey of Formal Methods in Self-Adaptive Systems," 2012.

[55] N. M. Villegas, H. A. Müller, G. Tamura, L. Duchien, and R. Casallas, "A Framework for Evaluating Quality-Driven Self-Adaptive Software Systems," 2011.

[56] S. Weibelzahl and G. Weber, "Advantages, opportunities and limits of empirical evaluations: Evaluating adaptive systems," *KI*, vol. 16, no. 3, pp. 17-20, 2002.

پیوست :

جدول (۱ پیوست) رتبه شاخص‌های مؤثر در ارزیابی سیستم کنترل ترافیک

نام شاخص	No	VL	L	M	H	VH	نام شاخص	No	VL	L	M	H	VH
پایداری						*	دقت						*
زمان تنظیم کوتاه						*	تخطی کم						*
پایان دهی						*	سازگاری						*
مقیاس پذیری						*	حساسیت						*
دانه بندی						*	پرهیز از شکست						*
پیشگویی						*	استحکام						*
سرعت پاسخ خود مختار					*		امنیت						*
هزینه					*		دقت تصمیم گیری						*
درجه خود مختاری						*	سطح خود مختاری					*	
خود پیکربندی						*	خود بهبودی					*	
خود بهینگی						*	خود آگاهی						*
خود حفاظتی					*		آگاهی به زمینه						*
باز بودن						*	ثبات	*					
قابلیت حمل	*						رضایت						*
قابلیت نگهداری						*	قابلیت استفاده						*
قابلیت عملیاتی(عملکرد)						*	قابلیت اعتماد					*	

جدول (۲ پیوست) رتبه متریک‌های مؤثر در ارزیابی سیستم

نام متریک	No	VL	L	M	H	VH	نام متریک	No	VL	L	M	H	VH
حسگر فعال						*	وفق‌پذیری ساختار داده						*
زمان پاسخ(اجرا)						*	وفق‌پذیری محیطی سخت‌افزار						*
زمان انتظار						*	وفق‌پذیری محیط اجرایی						*
بهره‌وری از منابع					*		وفق‌پذیری محیط نرم-افزاری						*
رضایت					*		درگاه‌های کاربر پسند						*
سادگی					*		نصب آسان					*	
قابلیت گسترش						*	آسانی تنظیمات مجدد					*	
یکپارچگی					*		همزیستی قابل دسترس				*		
مناسب بودن				*			زمان تصمیم						*
قابلیت همکاری درونی					*		پیچیدگی رفتار			*			
تاخیر						*	دانه بندی ریز/ درشت				*		

توان عملیاتی				*		ثبات					*		
ظرفیت				*		زمان شکست					*		
فراگیری عملیات				*		زمان بازیابی						*	
رفع شکست و نقص				*		سودمندی					*		
تشخیص شکست و نقص				*		قابلیت یادگیری					*		
پرهیز از شکست و نقص				*		قابلیت یادسپاری					*		
دانش						پیچیدگی تعاملات							
قدرت اتصال						محرمانگی							
عدم انکار						ساختار دهی مجدد							
سرعت تصمیم‌گیری						قابلیت بازیابی و شروع دوباره							
قابلیت تحلیل						قابلیت تغییر							
قابلیت دسترسی					*	قابلیت درک					*		
قابلیت آزمایش													

جدول (۳ پیوست) پرسشنامه رتبه متریک‌های مؤثر در ارزیابی سیستم

نام متریک	سوال	No	VL	L	M	H	VH
حسگر فعال	چه میزان معیار حسگرهای فعال (برای مشخص کردن حساسیت سیستم) برای سیستم فوق دارای اهمیت است؟						
زمان پاسخ(اجرا)	چه میزان زمان پاسخ در سیستم مورد نظر دارای اهمیت است؟						
زمان انتظار	چه میزان زمان انتظار کاربر برای دریافت جواب از سیستم دارای اهمیت است؟						
بهره‌وری در منابع	چه میزان امکان استفاده اشتراکی از منابع و مؤلفه‌های سیستم برای سیستم دارای اهمیت است؟						
سادگی	چه میزان سادگی مولفه‌های مورد استفاده در سیستم دارای اهمیت است؟						
قابلیت گسترش	چه میزان امکان بسط در طراحی سیستم دارای اهمیت است؟						
یکپارچگی	چه میزان ورودی غیر مجاز و تغییر غیر مجاز درداده‌ها دارای اهمیت است؟						
مناسب بودن	چه میزان مناسب بودن سیستم برای اجرای اهداف دارای اهمیت است؟						
قابلیت همکاری درونی	چه میزان توانایی همکاری در بین مولفه‌های درونی سیستم دارای اهمیت است؟						
تاخیر	چه میزان تاخیر در سیستم دارای اهمیت است؟						
توان عملیاتی	چه میزان توان عملیاتی و موفق شدن وظایف در یک دوره زمانی معین توسط سیستم دارای اهمیت است؟						
ظرفیت	چه میزان، ظرفیت و میزان دریافت وظیفه جهت اجرا توسط سیستم دارای اهمیت است؟						
رفع شکست و نقص	چه میزان رفع خودکار نقص و شکست در سیستم دارای اهمیت است؟						
تشخیص شکست و نقص	چه میزان تشخیص نقص و شکست در سیستم دارای اهمیت است؟						
پرهیز از شکست و نقص	چه میزان پرهیز از نقص و شکست در سیستم دارای اهمیت است؟						
قابلیت دسترسی	چه میزان دسترس بودن سیستم دارای اهمیت است؟						
زمان شکست	چه میزان مدت زمان از کار افتادن سیستم به علت شکست دارای اهمیت است؟						
زمان بازیابی	چه میزان مدت زمان بازیابی سیستم دارای اهمیت است؟						
سودمندی	چه میزان برآورده شدن اهداف کاربر توسط سیستم دارای اهمیت است؟						
قابلیت یادگیری	چه میزان سادگی یادگیری سیستم توسط کاربر دارای اهمیت است؟						
قابلیت یادسپاری	چه میزان به یادسپاری توابع و فرامین سیستم توسط کاربر دارای اهمیت است؟						
قابلیت درک	چه میزان قابل درک بودن سیستم توسط کاربر نهایی دارای اهمیت است؟						
وفق‌پذیری ساختار داده	چه میزان سازگاری داده‌های موجود با محیط جدید و سیستم جدید دارای اهمیت است؟						
وفق‌پذیری محیطی سخت‌افزار	چه میزان وفق‌پذیری نرم‌افزار با سخت‌افزار (جدید) دارای اهمیت است؟						
وفق‌پذیری محیط اجرایی	چه میزان وفق‌پذیری نرم‌افزار با محیط اجرایی مختلف دارای اهمیت است؟						
وفق‌پذیری محیط نرم‌افزاری	چه میزان وفق‌پذیری نرم‌افزار با محیط دارای اهمیت است؟						
درگاه‌های کاربر پسند	چه میزان کاربرپسند بودن درگاه‌های سیستم دارای اهمیت است؟						

سهولت آسان	چه میزان آسان بودن نصب سیستم توسط کاربر نهایی دارای اهمیت است؟						
سهولت تنظیمات مجدد	چه میزان سادگی تنظیمات مجدد سیستم توسط کاربر نهایی دارای اهمیت است؟						
همزیستی	چه میزان همزیستی سیستم با دیگر نرم‌افزارها، دارای اهمیت است؟						
زمان تصمیم	چه میزان مدت زمان لازم برای تصمیم‌گیری سیستم دارای اهمیت است؟						
پیچیدگی رفتار	چه میزان پیچیده بودن رفتار سیستم دارای اهمیت است؟						
دانه بندی ریز/ درشت	چه میزان ریز یا درشت بودن مؤلفه‌های سیستم دارای اهمیت است؟						
ثبات	چه میزان ثبات نتایج سیستم دارای اهمیت است؟						
رضایت	چه میزان رضایت کاربر نهایی از سیستم اهمیت دارد؟						
فراگیری عملیات	چه میزان امکان جایگزینی و مشابه بودن توابع در سیستم نرم افزاری دارای اهمیت است؟						
دانش	چه میزان دانش در زمان تصمیم‌گیری در سیستم اهمیت دارد؟						
قدرت اتصال	چه میزان قدرت اتصال و درنتیجه میزان تاخیر در بین مولفه‌ها در سیستم دارای اهمیت است؟						
پیچیدگی تعاملات	چه میزان پیچیدگی تعامل، و درنتیجه رفتار یک مولفه با سایر مولفه‌ها در سیستم دارای اهمیت است؟						
محرمانگی	چه میزان محرمانگی داده ها و عدم امکان دسترسی غیر مجاز به داده ها دارای اهمیت است؟						
ساختاردهی مجدد	چه میزان ساختاردهی مجدد سیستم دارای اهمیت است؟						
عدم انکار	چه میزان ردیابی فعالیت ها دارای اهمیت است؟						
سرعت تصمیم گیری	چه میزان سرعت تصمیم گیری در سیستم دارای اهمیت است؟						
قابلیت بازسازی و شروع دوباره	چه میزان قابلیت بازسازی و شروع دوباره در سیستم دارای اهمیت است؟						
قابلیت تحلیل	چه میزان تشخیص شکست یا شناسایی آیتم‌هایی که باید اصلاح شوند، در سیستم دارای اهمیت است؟						
قابلیت تغییر	چه میزان انجام و پیاده‌سازی تغییرات در سیستم دارای اهمیت است؟						
قابلیت آزمایش	چه میزان آزمایش در سیستم بعد از انجام تغییرات دارای اهمیت است؟						

جدول (۴ پیوست) پرسشنامه رتبه شاخص‌های مؤثر در ارزیابی سیستم

نام شاخص	No	VL	L	M	H	VH	نام شاخص	No	VL	L	M	H	VH
پایداری							دقت						
زمان تنظیم کوتاه							تخطی کم						
پایان دهی							سازگاری						
مقیاس پذیری							حساسیت						
دانه بندی							پرهیز از شکست						
پیشگویی							استحکام						
سرعت پاسخ خود مختار							امنیت						
هزینه							دقت تصمیم گیری						
درجه خود مختاری							سطح خود مختاری						
خود پیکربندی							خود بهبودی						
خود بهینگی							خود آگاهی						
خود حفاظتی							آگاهی به زمینه						
باز بودن							ثبات						
قابلیت حمل							رضایت						
قابلیت نگهداری							قابلیت استفاده						
قابلیت عملیاتی(عملکرد)							قابلیت اعتماد						

جدول (۵ پیوست) پرسشنامه مقادیر فازی متریک‌های مؤثر در ارزیابی سیستم

نام متریک	سوال	No	VL	L	M	H	VH
حسگر فعال	چه میزان معیار حسگرهای فعال (برای مشخص کردن حساسیت سیستم) برای سیستم مورد نظر برآورده شده است؟						
زمان پاسخ(اجرا)	چه میزان زمان پاسخ در سیستم مورد نظر برآورده شده است؟						
زمان انتظار	چه میزان زمان انتظار کاربر برای دریافت جواب از سیستم مطلوب است ؟						
بهره‌وری از منابع	چه میزان امکان استفاده اشتراکی از منابع و مؤلفه‌های سیستم برای سیستم برآورده شده است ؟						
سادگی	چه میزان سادگی مولفه‌های مورد استفاده در سیستم برآورده شده است؟						
قابلیت گسترش	چه میزان امکان بسط در طراحی سیستم وجود دارد؟						
یکپارچگی	چه میزان عدم امکان ورود غیر مجاز و تغییر غیر مجاز در داده‌ها برآورده شده است؟						
مناسب بودن	چه میزان سیستم برای اجرای اهداف خود مناسب است؟						
قابلیت همکاری درونی	چه میزان توانایی همکاری در بین مولفه‌های درونی سیستم برآورده شده است؟						
تاخیر	چه میزان تاخیر در سیستم قابل تحمل است؟						
توان عملیاتی	چه میزان توان عملیاتی و موفق شدن وظایف در یک دوره زمانی معین توسط سیستم مطلوب است؟						
ظرفیت	چه میزان، ظرفیت و میزان دریافت وظیفه جهت اجرا توسط سیستم مطلوب است؟						
رفع شکست و نقص	چه میزان رفع خودکار نقص و شکست در سیستم مطلوب است؟						
تشخیص شکست و نقص	چه میزان تشخیص نقص و شکست در سیستم برآورده شده است؟						
پرهیز از شکست و نقص	چه میزان پرهیز از نقص و شکست در سیستم برآورده شده است؟						
قابلیت دسترسی	چه میزان سیستم دسترس است؟						
زمان شکست	چه میزان مدت زمان از کار افتادن سیستم به علت شکست مطلوب است؟						
زمان بازیابی	چه میزان مدت زمان بازیابی سیستم مطلوب است؟						
سودمندی	چه میزان اهداف کاربر توسط سیستم برآورده شده است؟						
قابلیت یادگیری	چه میزان یادگیری سیستم توسط کاربر ساده است؟						
قابلیت یادسپاری	چه میزان به یادسپاری توابع و فرامین سیستم توسط کاربر ساده است؟						
قابلیت درک	چه میزان سیستم توسط کاربر نهایی قابل درک بودن است؟						
وفق‌پذیری ساختار داده	چه میزان داده‌های موجود با محیط جدید و سیستم جدید سازگار است؟						
وفق‌پذیری محیطی سخت‌افزار	چه میزان نرم‌افزار با سخت‌افزار (جدید) وفق‌پذیر است؟						
وفق‌پذیری محیط اجرایی	چه میزان نرم‌افزار با محیط اجرایی مختلف وفق‌پذیر است؟						
وفق‌پذیری محیط نرم‌افزاری	چه میزان نرم‌افزار با محیط وفق‌پذیر است؟						
درگاه‌های کاربر پسند	چه میزان درگاه‌های سیستم کاربرپسند است؟						
نصب آسان	چه میزان نصب سیستم توسط کاربر نهایی آسان است؟						
سهولت تنظیمات مجدد	چه میزان تنظیمات مجدد سیستم توسط کاربر نهایی ساده است؟						
همزیستی	چه میزان سیستم با دیگر نرم‌افزارها، همزیستی دارد؟						
زمان تصمیم	چه میزان مدت زمان لازم برای تصمیم‌گیری سیستم مطلوب است؟						
پیچیدگی رفتار	چه میزان رفتار سیستم پیچیده است؟						
دانه بندی ریز/ درشت	چه میزان مؤلفه‌های سیستم ریز است؟						
ثبات	چه میزان نتایج سیستم دارای ثبات است؟						
رضایت	چه میزان کاربر نهایی از سیستم رضایت دارد؟						
فراگیری عملیات	چه میزان امکان جایگزینی و مشابه بودن توابع در سیستم نرم افزاری وجود دارد؟						
دانش	چه میزان دانش مناسب در زمان تصمیم‌گیری در سیستم وجود دارد؟						
قدرت اتصال	چه میزان قدرت اتصال و درنتیجه میزان تاخیر در بین مولفه‌ها در سیستم مطلوب است؟						
پیچیدگی تعاملات	چه میزان تعاملات و درنتیجه رفتار یک مولفه با سایر مولفه‌ها در سیستم پیچیده است؟						
محرمانگی	چه میزان محرمانگی داده ها و عدم امکان دسترسی غیر مجاز به داده ها برآورده شده است؟						
ساختاردهی مجدد	چه میزان سیستم دارای قابلیت ساختاردهی مجدد است؟						
عدم انکار	چه میزان امکان ردیابی فعالیت ها وجود دارد؟						
سرعت تصمیم گیری	چه میزان سرعت تصمیم گیری در سیستم مطلوب است؟						
قابلیت بازسازی و شروع دوباره	چه میزان سیستم قابلیت بازسازی و شروع دوباره دارد؟						
قابلیت تحلیل	چه میزان تشخیص شکست یا شناسایی آیتم‌هایی که باید اصلاح شوند، در سیستم برآورده شده است؟						
قابلیت تغییر	چه میزان انجام و پیاده‌سازی تغییرات در سیستم برآورده شده است؟						
قابلیت آزمایش	چه میزان امکان آزمایش در سیستم بعد از انجام تغییرات برآورده شده است؟						

Designing A New Genetic-Fuzzy Type 2 Approach to Evaluate Self-Adaptive Systems by Software Quality Indicators

Majid Abdolrazzagh-Nezhad[1], Eshrat Zargari[2], Mahdi Kherad[3]

1- Associate Professor, Department of Computer Science, Faculty of Computer & Industrial Engineering, Birjand University of Technology, Birjand, Iran

2- Department of Computer Engineering, Birjand Branch, Islamic Azad University, Birjand, Iran

3- Ph.D. Student of Information Technology, Faculty of Engineering, Department of computer Engineering, University of Qom, Iran

Emails: {abdolrazzagh@birjandut.ac.ir, Zargari.Eshrat@gmail.com, m.kherad@stu.qom.ac.ir }

***Abstract*:**
The possibility of developing intelligent systems compatible with the environment has increased by expanding computer engineering fields. Self-adaptive systems are one of the types of software systems that change their behavior according to system conditions and environmental conditions and adapt themselves to it. Although the evaluation of the performance of these systems has been covered in most researches, the evaluation of their quality has remained closed. Therefore, designing an approach is an essential issue to evaluate the quality of self-adaptive systems. The first challenge, the quality indicators of these systems are not fixed, specific, and definite parameters. For example, one of the qualitative indicators for a self-adaptive system is the ability to run software on different operating systems. This indicator may have different degrees of importance for different experienced individuals. The next challenge, some qualitative indicators are not mathematical variables, but rather a linguistic variable between users and experts, indicating that these qualitative quantities are fuzzy variables and can be completely formulated by fuzzy logic. In this article, a new genetic-fuzzy type 2 approach has been proposed to evaluate the quality of the systems based on their quality indicators. The fuzzy logic type 2 is used to describe qualitative indicators, and a genetic algorithm is utilized to determine the optimal fuzzy weights of the qualitative indicators. In the proposed method, it has been tried to compare the self-adaptive systems from two dimensions, including the software dimension and their self-adaptive dimension. This is despite the fact that most of the existing research deals with only one dimension. In order to evaluate the proposed method, a traffic control system called InSync, which is an adaptive traffic control system (ATCS) and contains multiple qualitative factors, has been used. The obtained results confirm the effectiveness of the proposed method due to the lack of scenario generation, the ability to extend the method to all software quality parameters, and its simplicity. Also, the method is more comprehensive than other existing evaluation models. In addition, it has been tried to compare the self-adaptive systems from two dimensions, including the software dimension and their self-adaptive dimension. This is despite the fact that most of the existing research deals with only one dimension.

***Keywords*:** Self-Adaptive System, Quality Indicators of The Software, Genetic Algorithm, fuzzy Type 2, Adaptive traffic control system (ATCS).