ارائه‌ی راهکاری جهت بهبود استدلال مبتنی بر قالب در برآورد هزینه‌ی نرم‌افزار با استفاده از جستجوی ممنوعه

کوشیار اسماعیلی راد[۱]، مهدی ملامطلبی[۲،*]

[۱] گروه کامپیوتر، واحد بوئین زهرا، دانشگاه آزاد اسلامی، بوئین زهرا، ایران

[۲،*] گروه مهندسی کامپیوتر و فناوری اطلاعات، واحد قزوین، دانشگاه آزاد اسلامی، قزوین، ایران

چکیده: ارائه‌ی برآوردی دقیق از هزینه‌ی مورد نیاز توسعه‌ی نرم‌افزار جهت مدیریت مؤثر و موفق پروژه‌های نرم‌افزاری، یک ضرورت محسوب می‌گردد. در مراحل آغازین توسعه‌ی نرم‌افزار، اطلاعات دقیقی از پروژه در دسترس نیست و همین امر، ارائه‌ی برآورد دقیق هزینه‌ی توسعه‌ی نرم‌افزار را چالش برانگیز می‌کند. این تحقیق با هدف افزایش دقت برآورد هزینه‌ی مورد نیاز توسعه‌ی نرم‌افزار، روشی بر پایه‌ی استدلال مبتنی بر قالب و جستجوی ممنوعه ارائه کرده است که با حذف ویژگی‌های غیرضروری، برآورد با حداقل ویژگی‌های مورد نیاز را برای پروژه‌های نرم‌افزاری، ممکن نموده است. مکانیزم انتخاب راه‌حل اولیه، شاخص ارزیابی تابع هدف، و حذف داده‌های گمشده، پارامترهای متغیر تحقیق حاضر هستند که اثرگذاری آنها در قالب ۸ سناریوی مختلف، مورد بررسی قرار گرفته است. قدرت پیش‌بینی روش ارائه‌شده نسبت به روش اعتبارسنجی متقابل جامع و غیر جامع در مجموعه داده‌های آلبرخت، چاینا، دِشارنایز، کِمِرِر، کیچنهام، مَکسول و میازاکی، اندازه‌گیری شده است و با استفاده از دو روش اخیر، مورد ارزیابی قرار گرفته است. نتایج، برتری روش پیشنهادی را نسبت به روش‌های مشابه، بر اساس شاخص‌های ارزیابی سطح پیش‌بینی، میانه‌ی خطای نسبی، میانگین خطای نسبی و میانگین خطای مطلق، نشان داده است. به علاوه، کارایی روش پیشنهادی، از متدلوژی و محیط استفاده‌شده جهت توسعه‌ی نرم‌افزار، مستقل است و با تمرکز بر بخش دانش عمومی استدلال مبتنی بر قالب، پایداری دقت برآوردهای ارائه‌شده در مجموعه داده‌های تحقیق را فراهم آورده است.

کلیدواژه: برآورد هزینه نرم‌افزار، استدلال مبتنی بر قالب، انتخاب ویژگی، جستجوی ممنوعه، میانگین خطای نسبی، میانگین خطای مطلق

۱- مقدمه

علی‌رغم تلاش‌های فراوان در راستای بهبود توسعه‌ی نرم‌افزار، میزان شکست در پروژه‌های نرم‌افزاری همچنان بالا است. آمارها بیانگر آن است که هزینه‌ی تحمیل‌شده از این گونه شکستها در اقتصاد ایالات‌متحده‌ی آمریکا در بازه‌ی سال ۲۰۰۱ تا ۲۰۰۶ بین ۲۵ تا ۷۵ میلیارد دلار بوده است. اصلی‌ترین عامل شکست در پروژه‌های نرم‌افزاری، برآورد غیردقیق منابع مورد نیاز است [۱]. برآورد هزینه یکی از کارهای زمان‌بر در پروژه‌های بزرگ نرم‌افزاری است که می‌تواند بین ۲۰ تا ۴۰ درصد از کل زمان فاز آغازین چرخه حیات نرم‌افزار، مرحله‌ی جمع‌آوری نیازمندی‌ها، را به خود اختصاص دهد [۲]؛ اما در مقابل، دارای فواید بسیاری است که می‌تواند موفقیت پروژه را تضمین کند. درک هزینه‌ی نرم‌افزار بسیار مهم است چراکه برآورد هزینه روی بسیاری از امور ضروری فرآیند توسعه‌ی نرم‌افزار مانند انتخاب پروژه، برنامه‌ریزی پروژه و کارکنان، ارزیابی توسعه‌دهندگان، نظارت بر پیشرفت پروژه، ارزیابی میزان موفقیت پروژه و مدیریت ریسک [۳] تأثیر می‌گذارد که همگی گویای اهمیت این حوزه‌ی تحقیقاتی در توسعه‌ی نرم‌افزار است.

به‌عنوان یک تعریف، روند محاسبه‌ی طرح، سختی کار، تلاش، اندازه راه‌کارهای نرم‌افزاری و مجموعه هزینه‌های مرتبط با توسعه‌ی برنامه‌های نرم‌افزاری را برآورد هزینه‌ی نرم‌افزار گویند [۴]. تعریف ارائه‌شده از برآورد هزینه، دلالت بر مدیریت منابع دارد که از اصلی‌ترین مسائل روند تولید است. با شناخت کافی از منابع و توانایی‌ها، می‌توان توسعه‌ی پروژه را شتاب داده و ضمن کاهش هزینه‌ها، کیفیت نرم‌افزار را نیز بهبود بخشید [۵]. به‌طورکلی، مدل‌های هزینه‌ی مهندسی نرم‌افزار و تکنیک‌های برآورد برای چندین منظور شامل بودجه‌بندی، سبک‌سنگین کردن و آنالیز ریسک، برنامه‌ریزی و کنترل پروژه و آنالیز سرمایه‌گذاری بهبود نرم‌افزار مورد استفاده قرار می‌گیرند [۶].

از دید شرکت‌های توسعه‌دهنده‌ی نرم‌افزار، توان پردازشی و نیروی انسانی از اصلی‌ترین منابع موردنیاز توسعه به شمار می‌روند. با توجه به اینکه بر اساس آمار موجود، توان پردازشی در هر ۱۸ ماه، تقریباً رشد دو برابری دارد، نیروی انسانی یک هزینه‌ی قابل توجه پیش روی شرکت‌ها در بودجه‌ی توسعه نرم‌افزار است و برنامه‌ریزی صحیح تلاش آن‌ها برای شرکت‌ها جنبه‌ی کلیدی دارد [۷]. به همین روی، بیشتر روش‌های برآورد هزینه نرم‌افزار، روی این جنبه تمرکز دارند و بر همین اساس، معمولاً برآوردهای انجام‌شده هزینه تولید نرم‌افزار، با مبنای نفر-ماه ارائه می‌شوند [۸].

عوامل شناخته‌شده و شناخته‌نشده‌ی زیادی روی برآورد هزینه‌ی نرم‌افزار مؤثر هستند. به‌طور مثال، دقیق و هدفمند نبودن نیازمندی‌های پروژه، تازگی پروژه یا فنّاوری‌های موردنیاز آن، تلاش برای تطابق برآورد انجام‌شده با بودجه و زمان در نظر گرفته‌شده، تغییرات غیرعملی و سنگین در طرح‌ریزی پروژه در زمان اجرا، زبان برنامه‌نویسی، توانایی گروه و مرحله‌ای که در حال اجراست از عوامل مؤثر در برآورد هزینه‌ی نرم‌افزار هستند. یکی دیگر از عواملی

که در فرآیند برآورد هزینه‌ی نرم‌افزار مهم و تأثیرگذار است اندازه‌ی پروژه است ضمن آنکه، دقت در محاسبه آن، تأثیر مستقیم بر روی برآورد انجام‌شده خواهد داشت [۵, ۹, ۱۰].

در دسترس نبودن اطلاعات کافی از پروژه در مراحل ابتدایی، تخمین اندازه‌ی آن را دشوار می‌کند. در پروژه‌های نرم‌افزاری، معیارهایی چون تعداد خطوط کد (LOC)[1]، نقاط عملکرد (FP)[2]، نقاط مورد استفاده (UCP)[3] و یا تعداد صفحات HTML[4] در پروژه‌های مبتنی بر وب، جهت سنجش اندازه‌ی پروژه در نظر گرفته می‌شوند. هیچ‌یک از این معیارها، در ابتدای پروژه به‌طور دقیق در دسترس نیستند که این امر نیز به دشواری‌های ارائه‌ی برآورد دقیق می‌افزاید [۱۱]. از طرفی، نیازمندی‌های پروژه که نقشی تعیین‌کننده در اندازه‌ی نرم‌افزار دارند، به‌طور مکرر در فرآیند توسعه و برآورد نرم‌افزار دچار تغییر می‌شوند. همین امر، بر ضرورت انجام برآورد در سراسر چرخه‌ی حیات و توسعه‌ی نرم‌افزار دلالت دارد [۱۰, ۱۱]. ازاین‌رو، تغییرات مکرر در نیازمندی‌های پروژه هم به پیچیدگی‌های موجود در این حوزه‌ی تحقیقاتی می‌افزاید.

در توسعه‌ی پروژه‌های مبتنی بر وب، برآورد هزینه، حیاتی و چالش‌برانگیز است [۷]. این چالش ناشی از تفاوت‌های موجود در ویژگی‌های توسعه‌ی نرم‌افزارهای مبتنی بر وب، با توسعه‌ی سایر پروژه‌های نرم‌افزاری است. تفاوت موجود بین توسعه‌ی نرم‌افزارهای مبتنی بر وب و سایر نرم‌افزارها، سازگار کردن فرآیندها، معیارها و مدل‌ها را برای برآورد کنندگان دشوار کرده است [۱۲]. تفاوت‌های نرم‌افزارهای مبتنی بر وب با نرم‌افزارهای سنتی که برآورد کنندگان این حوزه را دچار چالش می‌کند، عبارت‌اند از:

۱) هدف اصلی توسعه‌ی نرم‌افزارهای مبتنی بر وب، ارائه‌ی سریع برنامه‌های باکیفیت، در کوتاه‌ترین زمان ممکن است. در مقابل، توسعه‌ی نرم‌افزارهای سنتی، در پی ارائه‌ی نرم‌افزار باکیفیت با کمترین هزینه هستند. ۲) اندازه‌ی پروژه‌های مبتنی بر وب معمولاً کوچک و اندازه‌ی گروه توسعه، بین ۳ تا ۵ نفر است. در مقابل، توسعه‌ی سنتی شامل پروژه‌هایی با اندازه متوسط و بزرگ بوده و گروه توسعه نیز معمولا متشکل از صدها نفر است. ۳) زمان‌بندی در توسعه‌ی نرم‌افزارهای مبتنی بر وب، بین ۳ تا ۶ ماه است که این زمان در توسعه‌ی نرم‌افزارهای سنتی، ۱۰ تا ۱۸ ماهه است. ۴) فرآیندهای توسعه‌ی به کار گرفته‌شده در توسعه‌ی نرم‌افزارهای مبتنی بر وب، با فرآیندهای توسعه‌ی نرم‌افزارهای سنتی متفاوت است. ۵) فنّاوری‌های مهندسی استفاده‌شده در نرم‌افزارهای مبتنی بر وب، از نرم‌افزارهای سنتی متفاوت است. ۶) محصولات توسعه‌یافته‌ی مبتنی بر وب، سامانه‌های مبتنی بر شیء هستند و قابلیت استفاده‌ی مجدد بالایی دارند. همچنین رابط خارجی آن‌ها محدود و نسبتاً ساده است. در مقابل، محصول نهایی در توسعه‌ی نرم‌افزارهای سنتی، سامانه‌های مبتنی بر کد و جدیدی است که رابط‌های خارجی زیاد دارد و غالباً نرم‌افزارهای پیچیده‌ای هستند، اما قابلیت استفاده مجدد کمتر در آن‌ها دیده می‌شود. ۷) افرادی که در توسعه‌ی نرم‌افزارهای مبتنی بر وب مشغول به کار می‌شوند، گرافیست‌ها، مهندسین نرم‌افزار با کمتر از ۲ سال سابقه و فارغ‌التحصیلان دانشگاهی هستند. در مقابل، افراد مشغول در توسعه‌ی نرم‌افزارهای سنتی، مهندسین زبده‌ی نرم‌افزار، با بیش از ۵ سال سابقه در حداقل دو حوزه‌ی برنامه‌نویسی هستند. ۸) تکنیک‌های برآورد مورد استفاده در توسعه‌ی نرم‌افزارهای مبتنی بر وب، قیاس مبتنی بر تجربه‌ی جاری است و طراحی، متناسب با منابع در دسترس انجام می‌شود. اما در توسعه‌ی نرم‌افزارهای سنتی، قیاس مبتنی بر داده‌های ثبت‌شده انجام می‌گیرد و در برآورد، از مدل‌هایی که از شاخص‌های اندازه‌گیری تعداد خطوط کد (LOC) و نقاط عملکرد (FP) بهره برده‌اند، استفاده می‌شود.

اهداف این تحقیق عبارتند از: ۱) افزایش شاخص ارزیابی سطح پیش‌بینی (PRED)[5] ۲) کاهش شاخص ارزیابی میانگین خطای نسبی (MMRE)[6] و میانگین خطای مطلق (MAE)[7] ۳) افزایش دقت برآورد هزینه‌ی نرم‌افزار که درنتیجه‌ی اهداف فوق حاصل خواهد شد. همچنین محدوده تحقیق عبارت است از: ۱) در این تحقیق روش استدلال مبتنی بر قالب برای برآورد هزینه توسعه‌ی نرم‌افزار انتخاب‌شده است. ۲) این تحقیق، پروژه‌های نرم‌افزاری و چندرسانه‌ای مبتنی بر وب را پوشش می‌دهد. این مقاله در پنج بخش تنظیم شده است. در بخش دوم، انواع روش‌های برآورد هزینه‌ی نرم‌افزار بامطالعه‌ی پیشینه‌ی تحقیق معرفی و مدل‌های ارائه‌شده بر پایه‌ی هر یک از آن‌ها مورد نقد و بررسی قرار گرفته‌اند. روش پیشنهادی این مقاله در راستای بهبود دقت برآورد روش استدلال مبتنی بر قالب در بخش سوم به‌تفصیل تشریح می‌گردد. بخش چهارم نتایج به‌دست‌آمده از روش پیشنهادی را ارائه و ارزیابی می‌کند و درنهایت، بخش پنجم، تحقیق حاضر را نتیجه‌گیری می‌نماید.

۲-کارهای مرتبط

هدف از برآورد تلاش مورد نیاز برای تولید نرم‌افزار، پیش‌بینی میزان تلاش مورد نیاز انجام یک پروژه بر اساس ویژگی‌های پروژه‌های دیگر است که طبق بررسی، با تلاش پروژه‌ی جدید در

[1] Line Of Code
[2] Function Point
[3] Use Case Point
[4] Hypertext Markup Language

[5] prediction level
[6] Mean of Magnitude of Relative Error
[7] Mean Absolute Error

ارتباط هستند. ویژگی‌های پروژه به عنوان ورودی و متغیر مستقل هستند، و تلاش به عنوان خروجی و متغیر وابسته است. دیگر متغیرهای مستقل مانند تعداد توسعه‌دهندگان و میانگین تجربه آن‌ها، تعداد ابزارهایی که به کار گرفته می‌شوند، تعداد صفحات جدید و تعداد امکاناتی که باید تعبیه شوند نیز در میزان تلاش مؤثر هستند که به عنوان محرک‌های تلاش تعریف می‌شوند. همچنین با توجه به تکنیکی که جهت برآورد تلاش از آن استفاده می‌شود، می‌توان از داده‌های پروژه‌های پیشین جهت پیش‌بینی تلاش مورد نیاز پروژه‌ی جدید، کمک گرفت.

فرآیند برآورد تلاش شامل موارد ذیل است: ۱) شناسایی محرک‌هایی که روی میزان تلاش مورد نیاز جهت توسعه وب مؤثر هستند مانند تعداد صفحات جدید و یا تعداد تصاویر جدیدی که باید طراحی شوند. ۲) برقراری تناسب بین محرک‌ها و تلاش مورد نیاز توسعه وب. به طور مثال، با افزایش تعداد صفحات ایستایی که در توسعه‌ی پروژه نیاز است، میزان تلاش پروژه نیز افزایش پیدا می‌کند. ۳) دریافت اطلاعات از تاریخچه پروژه‌های پیشین که دارای محرک‌ها و میزان تلاش مشخص هستند. ۴) ایجاد مدل برآورد تلاش با استفاده از اطلاعات پروژه‌های پیشین جهت پیش‌بینی تلاش مورد نیاز پروژه‌های جدید. ۵) ارزیابی دقت پیش‌بینی که میزان کارآمدی آن را نشان می‌دهد.

کومار[1] و بِهِرا[2] [۱۳] حل مسائل برآورد هزینه‌ی نرم‌افزار را با استفاده از تکنیک‌های محاسبات نرم که شامل شبکه عصبی مصنوعی[3]، منطق فازی[4] و محاسبات تکاملی[5] می‌شود، بررسی کرده‌اند. نتایج تحقیق آنها بیانگر محبوبیت بیشتر مجموعه داده‌ی کوکومو و شاخص‌های ارزیابی MMRE و PRED در تکنیک‌های محاسبات نرم است. همچنین، بر اساس تحقیق کومار و بِهِرا، شبکه عصبی مصنوعی در مقایسه با منطق فازی و محاسبات تکاملی، بیشتر در حل مسائل برآورد هزینه‌ی نرم‌افزار، مورد استفاده قرار گرفته‌اند.

اِذَر[6] و همکارانش [۹]، تکنیک‌های استفاده‌شده در حوزه‌ی برآورد تلاش نرم‌افزارهای مبتنی بر وب را بررسی کرده‌اند. نتایج تحقیق آنها بیانگر آن است که تکنیک‌های استدلال قیاس‌محور، رگرسیون خطی[7] و رگرسیون گام‌به‌گام[8]، به ترتیب با ۳۴/۵، ۳۴/۵ و و ۲۳/۸ درصد، بیشترین میزان تحقیقات را به خود اختصاص داده‌اند و شاخص‌های ارزیابی MMRE و Pred(25) ، پرکاربردترین شاخص ارزیابی در تحقیقات حوزه‌ی وب هستند. همچنین، بررسی انجام‌شده روی مجموعه داده‌ها، نشان می‌دهد که پروژه‌های صنعتی، در ۶۹ درصد از مقالات، مورد استفاده واقع شده‌اند.

ایده‌ی یافتن بهترین تکنیک برآورد هزینه‌ی نرم‌افزار، توسط کادودا[9] و همکارانش [۱۴] رد شده است؛ چراکه موفقیت یک تکنیک خاص، وابسته به اندازه‌ی مجموعه‌ی آموزشی[10]، ماهیت عملکرد هزینه و ویژگی‌های مجموعه داده مانند داده‌های پِرت[11]، ناسازگاری[12]، تعداد ویژگی‌ها، و تعداد پروژه‌های پیشین است. با این حال، بسیاری از مقالات با مقایسه‌ی مدل‌ها و تکنیک‌های برآورد هزینه‌ی نرم‌افزار، تلاش کرده‌اند مناسب‌ترین مدل‌ها و تکنیک‌ها را، در مجموعه داده‌ها و محیط‌های توسعه‌ی متفاوت، بیابند.

پندی[13] و همکارانش [۱۵] توانایی تکنیکهای رگرسیون خطی چندگانه، شبکه عصبی MLP[14]، الگوریتم ژنتیک و پیش‌بینی ساده لوحانه[15] را در برآورد هزینه‌ی نرم‌افزارهای تلفن‌های هوشمند، ارزیابی کرده‌اند. ارزیابی با استفاده مجموعه داده‌ی ساموئا[16] و شاخص‌های ارزیابی MMRE و PRED انجام شده است. نتایج تحقیق آنها بر اساس شاخص‌های MMRE و PRED، برتری الگوریتم ژنتیک در مقایسه با سه تکنیک دیگر را نشان داده است. با این حال، آنها توانایی چهار تکنیک ذکرشده را تنها در یک مجموعه داده ارزیابی کرده‌اند که میزان اطمینان به نتایج حاصل‌شده را کاهش می‌دهد و از معایب تحقیق آنها به شمار می‌رود.

دی‌یو[17] و همکارانش [۱۶] مدلی هوشمند و ترکیبی بر پایه‌ی رویکرد شبکه عصبی و منطق فازی معرفی کردند. این مدل با مدل الگوریتمی سی‌پِرسیم[18] [۱۷]، ترکیب شده و سیستم استنتاج سازگار فازی-عصبی (ANFIS)[19] [۱۸]، به عنوان معماری هریک از زیرمدل‌های فازی-عصبی مورد استفاده قرارگرفته است. نتایج تحقیق آنها که روی مجموعه داده‌ای متشکل از ۹۹ پروژه انجام شده است، حاکی از بهبود ۱۸ درصدی مدل آنها بر اساس شاخص ارزیابی MMRE است. با این حال، تعداد ورودی‌های زیاد از معایب این مدل است که پیچیدگی و کاهش دقت برآوردها را، به دنبال دارد.

[1] Kumar
[2] Behera
[3] Artificial Neural Networks
[4] fuzzy logic
[5] Evolutionary Computation
[6] Azhar et al.
[7] Linear regression
[8] Stepwise regression
[9] Kadoda et al.
[10] Training set
[11] Outlier
[12] Inconsistentcy
[13] Pandey
[14] Multilayer Perceptron
[15] Naive forecasting approach
[16] SAMOA (Software Analytics for Mobile Apps)
[17] Du et al.
[18] SEER-SEM
[19] Adaptive Neuro-Fuzzy Inference System

اَلگابرای[1] و همکارانش [۱۹]، ضرایب مدل کوکومو[2] ارائه‌شده‌ی بوئم [۲۰] را به منظور افزایش دقت، با استفاده از الگوریتم ژنتیک پیکربندی کرده‌اند. جهت ارزیابی روش پیشنهادی، از مجموعه داده‌ی ناسا در آزمایش‌ها استفاده نمودند و روش پیشنهادی آنها، برای هر سه حالت مدل کوکومو (ارگانیک، نیمه جدا[3] و جاسازی-شده[4])، به صورت جداگانه، بررسی شده است. نتایج به دست آمده، نشان‌دهنده‌ی افزایش دقت برآوردهای ارائه‌شده توسط روش آنها، در هر سه حالت مدل کوکومو است. با این حال، کوکومو یک مدل الگوریتمی است و در شرایطی که محرک‌های هزینه، قابل اندازه‌گیری نباشند، قادر به ارائه‌ی برآورد نیست.

در همین راستا، سینگال[5] و همکارانش [۲۱] با استفاده از الگوریتم تکاملی تفاضلی[6] به ارائه‌ی روشی جهت بهبود مدل‌های الگوریتمی کوکومو و کوکوموی۲ پرداخته‌اند. در تحقیق آنها، مقادیر پارامترهای مدل کوکومو و کوکوموی۲ با استفاده از الگوریتم تکاملی تفاضلی محاسبه شده و روی دو مجموعه داده‌ی کوکومو و ناسا، آزمایش شده‌اند که نتایج، حاکی از برتری مدل پیشنهادی آنها در مقایسه با مدل‌های اصلی کوکومو و کوکوموی۲ بر اساس شاخص MMRE، بوده است. با این حال، کارآمدی مدل آنها تنها با روش‌های پایه سنجش شده است که از معایب این تحقیق به شمار می‌رود.

نَصیف و همکارانش [۲۲]، مدل رگرسیون دیگری جهت غلبه بر مشکلات موجود در مدل کارنِر [۲۳] ارائه نمودند که ارزیابی آن با استفاده از ۲۴ پروژه، و به روش‌های کارنِر و وینتِرز[7] [۲۴] انجام شده است. نتایج‌شان بیانگر آن است که مدل رگرسیون پیشنهادی، بر اساس شاخص‌های ارزیابی PRED(25)، PRED(35)، و MMRE به ترتیب ۱۶/۵، ۲۵، و ۶ درصد نسبت به مدل پایه‌ی UCP، دقیق-تر عمل کرده است. نتایج به دست آمده، تنها در یک مجموعه داده‌ی کوچک ۲۴ پروژه‌ای ارزیابی شده است. بنابراین، جهت اطمینان از بهبودهای حاصل‌شده، عدم وابستگی مدل ارائه‌شده به مجموعه داده، نیازمند بررسی بیشتر است.

کورازا[8] و همکارانش [۲۵]، رگرسیون بردار پشتیبان را به عنوان یک تکنیک با پتانسیل بالا، در برآورد تلاش نرم‌افزار معرفی کردند. عملکرد رگرسیون بردار پشتیبان، به پارامترهای زمان اجرا وابسته است. آنها به منظور انتخاب خودکار پارامترهای مناسب، روشی با استفاده از جستجوی ممنوعه، ارائه کردند. جهت بررسی توانایی در تنظیم پارامترهای رگرسیون بردار پشتیبان، این روش با استراتژی پیکربندی تصادفی، تنظیمات پیش‌فرض ابزار وِکا [۲۶]، و الگوریتم جستجوی گرید [۲۷]، ارزیابی شده است. مدل ارائه‌شده، با روش استدلال مبتنی بر قالب و رگرسیون چندگانه‌ی گام‌به‌گام مقایسه شده است و نتایج که با استفاده از مجموعه داده‌های پرامیس و توکوتوکو به دست آمدند، نشان‌دهنده‌ی برتری جستجوی ممنوعه، در تنظیم پارامترهای مدل رگرسیون بردار پشتیبان بوده‌اند.

عبدالعلی[9] و همکارانش [۲۸] مدلی مبتنی بر الگوریتم جنگل تصادفی[10] ارائه نموده‌اند. کارایی این مدل پس از بهینه‌سازی تجربی تجربی پارامترهایش با مدل پایه‌ی درخت رگرسیون مقایسه شده است. نتایج که با استفاده از شاخص‌های ارزیابی PRED، MMRE و MdMRE و مجموعه داده‌های توکوتوکو، کوکومو و آی‌اِس‌بی‌جی به دست آمده‌اند، برتری مدل جنگل تصادفی را نسبت به مدل پایه‌ی درخت رگرسیون نشان داده است. با این حال، کارایی مدل عبدالعلی و همکارانش تنها با مدل پایه‌ی درخت رگرسیون سنجیده شده است و جهت اطمینان بیشتر، مقایسه با روش‌های غیر پایه، ضروری به نظر می‌رسد.

تکنیک‌های سنتی برآورد تلاش نرم‌افزار، در برآورد پروژه‌هایی که با استفاده از متدلوژی توسعه‌ی نرم‌افزار چابک[11] توسعه داده می‌شوند، از دقت کافی برخوردار نیستند. در راستای حل این مشکل، ضیاءالدین[12] و ضیاء[13] [۲۹]، مدلی متناسب با متدلوژی-های توسعه‌ی تکرارشونده ارائه کردند. همچنین، پِریاسَمی[14] و گاود[15] [۳۰] با تمرکز روی جزئیات داخلی هر مورد کاربرد، به توسعه مدل UCP پرداختند. الویدِین[16] و هادی[17] [۳۱]، ۱۶ عامل محیطی جدید را برای برآورد تلاش نرم‌افزار، مورد استفاده قرار داده اند. ژا[18] و مالو[19] [۳۲] به تشریح ۲۴ عامل محیطی مرتبط با محیط پیاده‌سازی، و بررسی اثرات این عوامل در برآورد هزینه‌ی نرم‌افزار، با استفاده از مدل کارنِر [۲۳] پرداختند.

ژیا[20] و همکارانش [۳۳]، یک روش پیکربندی تحت نام ROME[21] ارائه دادند که از بهینه‌سازی مدل ترتیبی (SMO) جهت یافتن بهترین پیکربندی برآورد تلاش نرم‌افزار با توجه به مجموعه

[1] Algabri et al.
[2] Constructive Cost Model (COCOMO)
[3] Semi-detached
[4] Embedded
[5] Singal
[6] Differential Evolution
[7] Winters
[8] Corazza et al.

[9] abdelali
[10] Random forest
[11] Agile software development
[12] Ziauddin
[13] Zia
[14] Periyasamy
[15] Ghode
[16] Alwidian
[17] Hadi
[18] Jha
[19] Malu
[20] Xia
[21] Rapid Optimizing Methods for Estimation

داده انتخاب شده، استفاده نموده است و نتیجه تحقیق‌شان بیانگر آن است که نمی‌توان یک روش را جهت برآورد تلاش، توصیه نمود بلکه باید با توجه به مجموعه داده انتخاب شده یا محلی، روشها را اولویت‌بندی نمود.

کومار[1] و همکارانش [۳۴]، مدلی مبتنی بر رگرسیون تقویت گرادیان با استفاده از یادگیری گروهی[2] با هدف بهبود دقت برآورد تلاش نرم‌افزار و با به کارگیری مجموعه داده‌های کوکومو شامل ۶۳ پروژه و چاینا شامل ۴۹۹ پروژه ارائه دادند. مدل پیشنهادی توسط معیارهای MAE، MSE، RMSE، و R مورد ارزیابی قرار گرفت و دقت بالای ۹۰٪ را به دست آورد.

منصور احمد[3] و همکارانش [۳۵] یک روش جهت بهبود ارزیابی تلاش نرم‌افزار مبتنی بر بلاک چین[4] و وب ارائه نمودند. همچنین روش پیشنهادی‌شان را بر اساس MMRE[5]، MAE[6]، و PRED مورد سنجش قرار دادند و نتایج قابل قبولی از بهبود به دست آوردند. آنها آزمایشات خود را با استفاده از ۵۲ سازمان با در نظر گرفتن فقدان داده‌های تاریخچه‌ای، فقدان افراد خبره، و عدم جانبداری، انجام دادند و نتیجه گرفتند که روش مبتنی بر بلاک چین می‌تواند در غلبه بر فقدان موارد فوق، به آنها یاری رساند.

رانکوویچ[7] و همکارانش [۳۶] دو معماری شبکه عصبی مصنوعی جهت پیش‌بینی و ارزیابی تلاش نرم‌افزار ارائه نمودند. با توجه به سرعت یادگیری و دقت مناسب شبکه‌های عصبی، این ساختار در معماری آنها انتخاب شد و تلاش نمودند که MRE را در برآورد تلاش نرم‌افزار، کمینه نمایند. همچنین معماری پیشنهادی آنها، تاثیر مدلهای غیرپارامتری را در مقایسه با مدلهای پارامتری نظیر کوکومو۲۰۰۰ بررسی نموده است. بعلاوه، آنها از مجموعه داده‌های کوکومو۸۱، کوکومو۲۰۰۰، و کمرر جهت مقایسه‌ها و انجام آزمایشات، بهره بردند.

نتیجه‌ی بررسی انواع مدل‌های برآورد تلاش، نشان‌دهنده‌ی برتری مدل‌های مبتنی بر یادگیری ماشین، نسبت به مدل‌های الگوریتمی و مدل‌های مبتنی بر نظریه کارشناس است. نیازمندی مدل‌های الگوریتمی به داشتن اطلاعات کامل از پروژه، و مشخص نبودن فرآیند برآورد در مدل‌های مبتنی بر نظریه‌ی کارشناس، نشان‌دهنده‌ی عدم کارایی مدل‌های الگوریتمی و عدم تکرارپذیری مدل‌های مبتنی بر نظریه‌ی کارشناس است.

در مقابل، مدل‌های مبتنی بر یادگیری ماشین، تکرارپذیر هستند و برای ارائه‌ی برآورد، به اطلاعات کامل پروژه‌ی هدف نیاز ندارند. به علاوه، مدل‌های مبتنی بر یادگیری ماشین دارای رویکرد افزایشی هستند و با استفاده از تجربه‌ی کارشناس، و اطلاعات پروژه‌های پیشین، مزیت دو مدل الگوریتمی و نظریه‌ی کارشناس را به همراه دارند. در جدول ۱، جمع‌بندی و مقایسه این سه دسته مدل برآورد تلاش نرم‌افزار ارائه شده است.

[1] Kumar
[2] Ensemble learning
[3] Mansoor Ahmed
[4] Blockchain
[5] Mean Magnitude of Relative Error
[6] Mean Absolute Error
[7] Rankovic

پذیرفته شده در مجله محاسبات نرم

جدول ۱. جمع‌بندی و مقایسه مدلهای ارزیابی تلاش نرم‌افزار

مدل	ویژگیها	مزایا	معایب	نمونه
مدل مبتنی بر نظریه کارشناس	بهره‌گیری از دانش برآورد کننده (کارشناس)، و تجربه به‌دست‌آمده‌اش در پروژه‌های مشابه پیشین، اساس برآورد هزینه‌ی نرم‌افزار در مدل‌های مبتنی بر نظریه‌ی کارشناس است	انجام برآورد توسط نیروی انسانی و به کمک هوش طبیعی باعث می‌شود در مواقعی که داده‌ها غیرقابل‌اندازه‌گیری هستند هم قادر به ارائه‌ی برآورد باشد.	قاعده خاصی را دنبال نمی کند و بنابراین، تکرار پذیر نیست.	دِلفی[1] [۳۷]، دِلفی پهن‌باند[2] [۳۸]، و ساختار شکست کار[3] [۳۹]
مدل الگوریتمی	اساس کار استفاده از معادلات ریاضی، جهت برآورد هزینه‌ی نرم‌افزار بوده و اصلی‌ترین محرک هزینه در آنها، اندازه‌ی نرم‌افزار بر اساس تعداد خطوط برنامه، تعداد صفحات، و تعداد لینک‌ها (در پروژه‌های مبتنی بر وب) است.	تولید برآوردهای قابل تکرار، اصلاح آسان داده‌های ورودی، و اصلاح و شخصی‌سازی آسان معادلات	عدم قابلیت رویارویی با شرایط پیش‌بینی‌نشده. در مواردی که عوامل تأثیرگذار در برآورد هزینه، قابل اندازه‌گیری نباشند، قادر به ارائه برآورد نیست. نیازمند داشتن اطلاعات کامل از پروژه است.	کوکومو [۲۰]، اِسلیم[4] [۴۰]، سی‌پِرسیم [۱۷]، اِس‌دی‌سی[5] [۴۱]، تی‌آرِدَبلیو وُلورتون[6] [۴۲]، داتی[7] [۴۳]، پرایس اِس[8] [۴۴]، آی‌بی‌اِم-اِف‌دی‌اِس[9] [۴۵]، بوئینگ[10] [۴۶]، جی‌آرسی[11] [۴۷]، و مِتا[12]
مدل مبتنی بر یادگیری ماشین	با استفاده از ورودی‌های دریافتی، یک مدل را برای پیش‌بینی، و یا تصمیم‌گیری ایجاد می‌کند.	از داده‌های موجود، دانش جدید می‌آموزد و تکرارپذیر است. برای ارائه‌ی برآورد، به اطلاعات کامل پروژه‌ی هدف نیاز ندارد.	معمولا مستقلا استفاده نمی‌شود و به عنوان مکمل به همراه مدلهای دیگربه کار می‌رود.	مدل‌های مبتنی بر منطق فازی [۴۸]، درخت طبقه‌بندی و رگرسیون [۴۹]، شبکه‌های بیزین [۵۰]، شبکه‌های عصبی [۵۱]، رگرسیون بردار پشتیبان [۵۲]، و استدلال مبتنی بر قالب [۵۳]

[1] Delphi
[2] Wideband Delphi
[3] Work Breakdown Structure
[4] SLIM
[5] SDC
[6] TRW Wolverton
[7] Doty
[8] PRICE S
[9] IBM-FSD
[10] Boeing
[11] GRC
[12] Meta

پذیرفته شده در مجله محاسبات نرم

۳- روش پیشنهادی

در آزمایشات مربوط به ارزیابی روش پیشنهادی، برد اصلی رایانه میزبان، مدل Z270P-D3-CF برند گیگابایت است که با پردازنده‌ی چهار هسته‌ای Corei5 سری ۷۴۰۰ شرکت اینتل با فرکانس ۳/۰ گیگاهرتز، و حافظه‌ی اصلی ۸ گیگابایتی، تجهیز شده است. همچنین، سیستم عامل نصب‌شده روی سخت‌افزار رایانه‌ی میزبان، نسخه‌ی ۱۹۰۹ ویندوز ۱۰ است که بستر اجرای نرم‌افزار محاسبات آماری را فراهم کرده است. در این تحقیق، محاسبات آماری آزمایش‌ها در نرم‌افزار R انجام شده است که یک محیط یکپارچه و توسعه‌پذیر نرم‌افزاری برای پیاده‌سازی تکنیک‌ها و محاسبات آماری است.

این تحقیق، به بهینه‌سازی استدلال مبتنی بر قالب با استفاده از الگوریتم جستجوی ممنوعه پرداخته است. الگوریتم جستجوی ممنوعه در پنج مرحله‌ی زیر خلاصه می‌شود: مرحله اول (انتخاب راه‌حل اولیه): راه‌حل اولیه، به صورت تصادفی یا طبق مکانیزم تعیین‌شده، از مجموعه راه‌حل‌های مسئله انتخاب می‌شود. مرحله دوم (تشکیل مجموعه‌ی کاندید): مجموعه‌ی کاندید از راه‌حل‌های موجود در همسایگی راه‌حل جاری، تشکیل می‌شود. در اولین دور تکرار از جستجوی ممنوعه، راه‌حل اولیه به عنوان راه‌حل جاری در نظر گرفته می‌شود. مرحله‌ی سوم (یافتن راه‌حل برتر): راه‌حل‌های موجود در مجموعه‌ی کاندید، با استفاده از تابع هدف امتیازدهی می‌شوند. سپس، راه‌حلی که دارای بیشترین امتیاز است، به عنوان راه‌حل برتر، در نظر گرفته می‌شود. مرحله‌ی چهارم (بررسی راه‌حل برتر): پذیرش حرکت به راه‌حل برتر، در این مرحله بررسی می‌شود. اگر راه‌حل برتر در لیست ممنوعه وجود نداشته باشد، یا دارای امتیازی بیشتر از شاخص آرمانی باشد، حرکت به آن مجاز است؛ در غیر اینصورت، راه‌حل برتر از مجموعه‌ی کاندید حذف شده و فرآیند جستجو از مرحله‌ی سوم ادامه می‌یابد. مرحله‌ی پنجم (حرکت به راه‌حل برتر): حرکت به راه‌حل برتر، شامل به روزرسانی شاخص آرمانی، به روزرسانی لیست ممنوعه و انتخاب راه‌حل برتر به عنوان راه‌حل جاری است. پس از انجام این مرحله، جستجو از مرحله‌ی دوم تا رسیدن به شرط خاتمه ادامه می‌یابد.

پس از برقراری شرط خاتمه و توقف فرآیند جستجو، آخرین راه‌حل برتر پذیرفته‌شده در مرحله‌ی چهارم، به عنوان راه‌حل نهایی مسأله ارائه می‌شود. کیفیت هر یک از اعضای مجموعه‌ی کاندید، در مرحله‌ی سوم جستجوی ممنوعه، مورد بررسی قرار می‌گیرد و متناسب با تابع هدف، امتیازی به اعضا اختصاص داده می‌شود. از بین اعضاء مجموعه‌ی کاندید، عضوی که دارای بیشترین امتیاز است، به عنوان راه‌حل برتر انتخاب می‌شود. این تحقیق، در صدد بهبود دقت هزینه‌ی برآوردشده‌ی نرم‌افزار است و لذا، بیشترین امتیاز به عضوی اختصاص می‌یابد که کمترین میزان خطا را بر اساس شاخص‌های ارزیابی کسب کند. رابطه‌ی (۱)، نشان‌دهنده‌ی تابع هدف، و چگونگی محاسبه‌ی امتیاز اعضای مجموعه‌ی کاندید است.

$$P(x) = \frac{\sum_{i=1}^{n} 1 - e_i}{n} \times 100 \qquad (۱)$$

در رابطه‌ی (۱)، امتیاز عضو x از مجموعه‌ی کاندید محاسبه می‌شود که در آن، n اندازه‌ی مجموعه داده‌ی مورد آزمایش، و e_i خطای پیش‌بینی مورد i اُم از مجموعه داده است. مکانیزم انتخاب زیرمجموعه‌ای از ویژگی‌ها، در استدلال مبتنی بر قالب، با استفاده از الگوریتم جستجوی ممنوعه انجام می‌شود.

در این تحقیق، تفاوت سناریوی آزمایش‌ها، در پارامترهای متغیر پیکربندی استدلال مبتنی بر قالب و جستجوی ممنوعه است. مقدار پارامترهای ثابت پیکربندی در تمامی سناریوها یکسان است. در مقابل، مقدار پارامترهای متغیر، در هر سناریو تغییر می‌کند. استدلال مبتنی بر قالب، دارای شش پارامتر پیکربندی ثابت است. همچنین، دو پارامتر شرط خاتمه‌ی جستجو و اندازه‌ی لیست ممنوعه از پارامترهای ثابت جستجوی ممنوعه هستند. پارامتر مکانیزم انتخاب راه‌حل اولیه، تنها پارامتر متغیر جستجوی ممنوعه دارای دو حالت مختلف است. همچنین، پارامتر شاخص ارزیابی تابع هدف که در جستجوی ممنوعه و استدلال مبتنی بر قالب استفاده می‌شود، با چهار حالت مختلف، از پارامترهای متغیر سناریوی آزمایش‌ها است.

مکانیزم انتخاب زیرمجموعه‌ای از ویژگی‌ها، شاخص اندازه‌گیری شباهت، مقیاس‌گذاری، تعداد موارد مشابه، تطبیق تشابه، و قوانین تطبیق، پارامترهای ثابت استدلال مبتنی بر قالب هستند. مکانیزم انتخاب زیرمجموعه‌ای از ویژگی‌ها، در استدلال مبتنی بر قالب، با استفاده از الگوریتم جستجوی ممنوعه انجام می‌شود. در این تحقیق، فاصله‌ی اقلیدسی بدون وزن، شاخص اندازه‌گیری شباهت است که بر اساس آن، ۳ مورد مشابه برای پروژه‌ی هدف انتخاب می‌گردد. پارامتر تطبیق تشابه استدلال مبتنی بر قالب در تحقیق حاضر، میانگین رتبه‌بندی شده معکوس بوده و قوانین تطبیق نیز، پارامتر خنثی‌شده‌ی استدلال مبتنی بر قالب است.

مقدار پارامترهای ثابت جستجوی ممنوعه، بر اساس تعداد ویژگی‌های مجموعه داده‌ی مورد آزمایش، تعیین می‌شوند. با فرض n به عنوان تعداد ویژگی‌ها، پارامتر اندازه‌ی لیست ممنوعه برابر $2 * \sqrt{2^n}$، و پارامتر شرط خاتمه‌ی جستجو، اجرای $\sqrt{2^n}$ تکرار بدون بهبود شاخص آرمانی است. پارامتر مکانیزم انتخاب راه‌حل اولیه در جستجوی ممنوعه، دارای دو حالت متفاوت است. در حالت اول، تمامی ویژگی‌ها در راه‌حل اولیه انتخاب می‌شوند و جستجو از انتهای لیست آغاز می‌گردد. در مقابل، زیرمجموعه‌ی تهی راه‌حل‌های مسئله، در حالت دوم به عنوان راه‌حل اولیه در نظر گرفته می‌شود و جستجو از ابتدای لیست آغاز می‌گردد. همچنین، شاخص ارزیابی تابع هدف، دارای دو حالت MMRE و MAE است.

افزون بر پارامترهای متغیر مکانیزم انتخاب راه‌حل اولیه، و شاخص ارزیابی تابع هدف، پیش‌پردازش داده‌های این تحقیق دارای دو حالت متفاوت برآمده از حذف یا عدم حذف داده‌های گمشده[1] است. بنابراین، مجموعاً ۸ سناریوی متفاوت در آزمایش‌های این تحقیق وجود دارد که مقدار پارامترهای متغیرشان، در جدول۲ معرفی شده است.

[1] NA (Not Available)

جدول ۲. پارامترهای متغیر سناریوی آزمایش‌های تحقیق

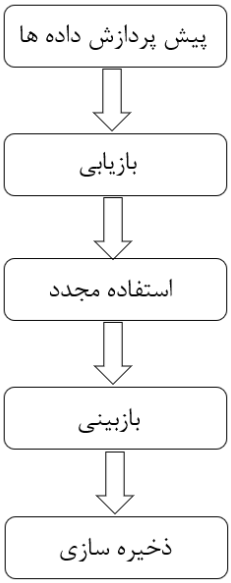
شناسه‌ی سناریو	مکانیزم انتخاب راه‌حل اولیه	شاخص ارزیابی	حذف داده‌های گمشده
۱	مجموعه‌ی ویژگی‌ها	MAE	بله
۲	مجموعه‌ی ویژگی‌ها	MMRE	بله
۳	مجموعه‌ی ویژگی‌ها	MAE	خیر
۴	مجموعه‌ی ویژگی‌ها	MMRE	خیر
۵	مجموعه‌ی تهی	MAE	بله
۶	مجموعه‌ی تهی	MMRE	بله
۷	مجموعه‌ی تهی	MAE	خیر
۸	مجموعه‌ی تهی	MMRE	خیر

در جدول۲، تنها مقدار پارامترهای متغیر ۸ سناریوی آزمایش‌ها، قید شده است. فرآیند جستجوی ممنوعه، در سناریوهای ۱ تا ۴ از ابتدای لیست، و در ۴ سناریوی دیگر، از انتهای لیست شروع می‌شود. در سناریوهای ۱، ۲، ۵ و ۶، پروژه‌های دارای مقادیر گمشده از مجموعه داده حذف شده و در دیگر سناریوها، حفظ شده‌اند. آزمایش‌های این تحقیق، دارای ۸ سناریوی متفاوت هستند و روی ۷ مجموعه داده از ۱۲ مجموعه داده‌ی گروه پیش‌بینی تلاش پرامیس، اعمال شده است. آلبرخت [۵۴]، چاینا [۵۵]، دِشارنایز [۵۶]، کِمِرِر [۵۷]، کیچنهام [۵۸]، مَکسوِل [۵۹]، و میازاکی [۶۰]، هفت مجموعه داده‌ی این تحقیق هستند که خلاصه‌ای از ویژگی‌هایشان، در جدول ۳ شرح داده شده است.

جدول ۳. خلاصه‌ای از ویژگی‌های مجموعه داده‌ی آزمایش‌ها

مجموعه داده	تعداد پروژه‌ها	تعداد ویژگی	ویژگی رسته‌ای	مقادیر گمشده	تلاش مورد نیاز توسعه			
					واحد	کمینه	بیشینه	میانگین
آلبرخت	۲۴	۸	۱	۰	هزار نفر-ساعت	۰/۵	۱۰۵/۲	۲۱/۸۷
چاینا	۴۹۹	۱۹	۲	۰	نفر-ساعت	۲۶	۵۴۶۲۰	۳۹۲۱/۰۴
دِشارنایز	۸۱	۱۲	۱	۴	نفر-ساعت	۵۴۶	۲۳۹۴۰	۵۰۴۶/۳۰
کِمِرِر	۱۵	۷	۲	۰	نفر-ماه	۲۳/۲	۱۱۰۷/۳۱	۲۱۹/۲۴
کیچنهام	۱۴۵	۱۰	۳	۱۳	نفر-ساعت	۲۱۹	۱۱۳۹۳۰	۳۱۱۳/۱۱
مَکسوِل	۶۲	۲۷	۵	۰	نفر-ساعت	۵۸۳	۶۳۶۹۴	۸۲۲۳/۲۰
میازاکی	۴۸	۹	۱	۰	نفر-ماه	۵/۶	۱۵۸۶	۸۷/۴۷

روش پیشنهادی تحقیق حاضر (روش برآورد هزینه داده‌گذر) توسعه‌ای از استدلال مبتنی بر قالب است. همانگونه که در شکل ۱ نشان داده شده است، چهار گام بازیابی، بازبینی، استفاده‌ی مجدد، و ذخیره‌سازی استدلال مبتنی بر قالب، به همراه گام پیش‌پردازش داده‌ها، مراحل تشکیل دهنده‌ی روش پیشنهادی این تحقیق هستند. نوآوری این تحقیق، ارائه‌ی پیکربندی جدید استدلال مبتنی بر قالب، و به کارگیری جستجوی ممنوعه در مرحله‌ی بازیابی‌اش است که در این بخش، تشریح شده است.



شکل ۱. مراحل روش پیشنهادی

گام اول: پیش‌پردازش داده‌ها

عملیات پیش‌پردازش داده‌ها شامل حذف ویژگی‌های خنثی، حذف پروژه‌های دارای مقادیر گمشده، و مقیاس‌گذاری روی ویژگی‌های عددی مجموعه داده است. منظور از ویژگی‌های خنثی، ویژگی‌هایی است که در تمامی پروژه‌ها دارای مقدار یکسان هستند. یکی از پارامترهای متغیر سناریوی آزمایش‌های این تحقیق، حذف داده‌های گمشده است. حذف یا عدم حذف داده‌های گمشده، دو حالت مختلف این پارامتر هستند. اگر در سناریوی آزمایش، پارامتر حذف داده‌های گمشده در حالت "بله" قرار گیرد، داده‌های گمشده در مرحله‌ی پیش‌پردازش داده‌ها حذف می‌شوند. پس از حذف ویژگی‌های خنثی و داده‌های گمشده، مقیاس‌گذاری روی ویژگی‌های عددی مجموعه داده اعمال می‌شود. قاعده‌ی مقیاس‌گذاری این تحقیق، استانداردسازی مقادیر ویژگی‌های عددی در بازه‌ی ۰ و ۱ است که در آن ۰ به کمترین مقدار و ۱ به بیشترین مقدار اختصاص می‌یابد. رابطه (۲)، چگونگی استانداردسازی مقادیر را نشان می‌دهد.

$$S(x_i) = \frac{x_i - min(x_1, \dots, x_n)}{max(x_1, \dots, x_n) - min(x_1, \dots, x_n)} \qquad (۲)$$

در رابطه (۲)، $S(x_i)$ مقدار استانداردسازی شده‌ی ویژگی x در i اُمین پروژه‌ی مجموعه داده است. همچنین x_i نشان‌دهنده‌ی مقدار ویژگی x در i اُمین پروژه، پیش از استانداردسازی است، و $min(x_1, \dots, x_n)$، و $max(x_1, \dots, x_n)$ به ترتیب مقدار کمینه و بیشینه‌ی ویژگی x در مجموعه داده‌ی n عضوی را نشان می‌دهند.

گام دوم: بازیابی

پارامترهای انتخاب زیرمجموعه‌ای از ویژگی‌ها، شاخص اندازه‌گیری شباهت، و تعداد موارد مشابه از پارامترهای ثابت تحقیق در پیکربندی استدلال مبتنی بر قالب هستند که در گام دوم مورد استفاده قرار گرفته است. همچنین، پارامترهای ثابت اندازه‌ی لیست ممنوعه، و شرط خاتمه‌ی

جستجو، به همراه پارامتر متغیر مکانیزم انتخاب راه‌حل اولیه، سه پارامتر استفاده‌شده در گام دوم جهت پیکربندی جستجوی ممنوعه هستند. افزون بر شش پارامتر ذکرشده، پارامتر متغیر شاخص ارزیابی، دیگر پارامتر تعیین‌کننده‌ی گام دوم تحقیق است که در تابع هدف مورد استفاده قرار می‌گیرد.

گام دوم روش برآورد هزینه داده گذر ، به شرح چگونگی به کارگیری جستجوی ممنوعه، در استدلال مبتنی بر قالب اختصاص دارد. وظیفه‌ی جستجوی ممنوعه در روش برآورد هزینه داده گذر، یافتن زیرمجموعه‌ای آرمانی از ویژگی‌ها، با هدف افزایش دقت برآورد به دست آمده از استدلال مبتنی بر قالب است. زیر مجموعه‌ی آرمانی ارائه‌شده توسط جستجوی ممنوعه، باید منطقی و قابل استفاده در استدلال مبتنی بر قالب باشد. هر زیرمجموعه از ویژگی‌ها، با رشته بیتی به طول n نشان داده می‌شود که در آن n برابر تعداد ویژگی‌های مجموعه داده است.

جستجوی ممنوعه با انتخاب راه‌حل اولیه که در این تحقیق دارای دو حالت مجموعه‌ی تهی و مجموعه‌ی کامل ویژگی‌ها است، آغاز می‌شود. اگر مجموعه داده‌ی مورد آزمایش دارای ۳ ویژگی باشد، در حالت مجموعه تهی، و مجموعه‌ی کامل ویژگی‌ها، به ترتیب جستجو از ۰۰۰ و ۱۱۱ شروع می‌شود. جستجوی ممنوعه پس از انتخاب یکی از حالت‌های راه‌حل اولیه، مجموعه‌ی کاندید را تولید می‌کند. همانطور که در شبه کد ارائه شده در شکل ۲ نشان داده شده است، مجموعه‌ی کاندید از همسایه‌های راه‌حل اولیه تشکیل می‌شود. متغیرهای S، TL، AL، I، و WOI در خطوط ۱ تا ۳، مقداردهی شده‌اند که به ترتیب نشان‌دهنده‌ی راه‌حل اولیه، لیست ممنوعه، شاخص آرمانی، تعداد تکرارهای جستجوی ممنوعه، و تعداد تکرارهای بدون بهبود شاخص آرمانی هستند.

```
Start
1.    S = initial solution
2.    TL = ∅
3.    AL = I = WOI = 0
4.    While (termination rule is reached):
5.        C = Generate candidate set from S neighbours
6.        B = Find best solution of candidate set (C)
7.        bScore = objectiveFunction(B)
8.        if (B ∉ TL  OR  bScore > AL) Move to B:
9.            S = B
10.           Add B to TL
11.           if (bScore > AL):
12.               WOI = 0
13.               AL = bScore
14.               topS = B
15.           else WOI++
16.       else:
17.           WOI++
18.           Remove B from C and return to line7
19.       I++
20.   EndWhile
21.   SP = similarProjects(topS , P)
22.   Return SP
End
```

شکل ۲. گام دوم روش پیشنهادی

حلقه‌ی تعریف‌شده در خط چهارم، تا رسیدن به شرط خاتمه ادامه می‌یابد و در هر دور تکرار، خط هشتم، مجاز بودن حرکت به راه‌حل برتر ذخیره‌شده در متغیر B را ارزیابی می‌کند. راه‌حل برتر از مجموعه‌ی کاندید C که همسایه‌های S هستند بر اساس امتیاز دریافتی از تابع objectiveFunction انتخاب می‌شود. در صورت مجاز بودن حرکت، راه‌حل برتر جایگزین راه‌حل اولیه شده و B به لیست ممنوعه اضافه می‌شود. شرط خاتمه‌ی حلقه، اجرای n تکرار بدون بهبود شاخص آرمانی است.

در نهایت، زیرمجموعه‌ی آرمانی ذخیره‌شده در متغیر topS، توسط جستجوی ممنوعه ارائه‌شده و به عنوان اولین ورودی، در تابع similarProjects مورد استفاده قرار می‌گیرد. تابع similarProjects، انتخاب ویژگی را از مجموعه داده‌ی مورد آزمایش، مطابق متغیر topS انجام می‌دهد و متناسب با مشخصات پروژه‌ی هدف P، و شاخص اندازه‌گیری شباهت به یافتن موارد مشابه می‌پردازد. خروجی تابع similarProjects، بردار ویژگی‌های سه پروژه است که دارای کمترین فاصله با بردار ویژگی‌های پروژه‌ی هدف هستند. بردار ویژگی موارد مشابه، در متغیر SP ذخیره می‌شود که خروجی گام دوم روش برآورد هزینه داده گذر است.

گام سوم: استفاده‌ی مجدد

خروجی گام دوم، راه‌حل مسائل مشابه مسئله‌ی جدید است. در گام سوم، بر اساس فرض استدلال مبتنی بر قالب، از راه‌حل مسائل مشابه، برای ارائه‌ی راه‌حل مسئله‌ی جدید استفاده شده است. در این تحقیق، مسئله، مشخصات پروژه‌ی نرم‌افزاری، و راه‌حل، تلاش مورد نیاز توسعه‌ی پروژه است. بنابراین، گام سوم به ارائه‌ی برآورد تلاش پروژه‌ی هدف با استفاده از تلاش موارد مشابه، پرداخته است. بر اساس پارامتر ثابت تعداد موارد مشابه، گام دوم، سه مسئله را برای استفاده‌ی مجدد به گام سوم ارجاع می‌دهد. دیگر پارامتر ثابت گام سوم، تطبیق تشابه است که با استفاده از رابطه (۳)، به ارائه‌ی برآورد تلاش پروژه‌ی هدف بر اساس تلاش موارد مشابه، می‌پردازد.

$$E_{target} = \frac{3E_{1stSim} + 2E_{2thSim} + E_{3thSim}}{6} \qquad (۳)$$

رابطه (۳)، روش میانگین رتبه‌بندی شده معکوس است که جهت استفاده در پارامتر تطبیق تشابه این تحقیق، شخصی‌سازی شده است. در رابطه (۳)، E_{1stSim}، E_{2thSim}، و E_{3thSim}، مقدار تلاش اولین، دومین، و سومین مورد مشابه پروژه‌ی هدف هستند که بر اساس میزان تشابه به صورت نزولی مرتب‌سازی شده‌اند. ضرایب اختصاص داده شده به هر یک از پروژه‌های مشابه، نشان‌دهنده‌ی میزان اثرگذاری‌شان در تلاش برآوردشده‌ی پروژه‌ی هدف است. میزان تأثیرگذاری اولین، دومین، و سومین پروژه‌ی مشابه، روی تلاش پروژه‌ی هدف، به ترتیب ۵۰٪، ۳۳٪، و ۱۷٪ است. حاصل رابطه (۳)، تلاش برآورد شده برای پروژه‌ی هدف است که با E_{target} نشان داده شده است.

گام چهارم: بازبینی

راه‌حل ارائه‌شده در گام سوم، یک راه‌حل ارزیابی‌نشده است که صحت آن در گام چهارم بررسی می‌شود. گام چهارم شامل دو بخش است. در اولین بخش، راه‌حل ارائه‌شده‌ی گام سوم ارزیابی شده و در صورت درستی،

به گام پنجم ارائه می‌شود. ارزیابی توسط خود مدل، به کمک کارشناس، و یا با به کارگیری در محیط واقعی انجام می‌گیرد که بسته به نوع مسئله، تعیین می‌شود. علاوه بر نوع ارزیابی، مدت زمان مورد نیاز جهت ارزیابی هم به نوع مسئله وابسته است.

در صورت عدم درستی راه‌حل ارائه‌شده‌ی گام سوم، بخش دوم گام چهارم به اصلاح راه‌حل غلط می‌پردازد. به همین منظور، گام چهارم ابتدا خطای راه‌حل را شناسایی و سپس توضیحاتی در مورد چرایی صحیح نبودن راه‌حل، ارائه می‌کند. پس از شناسایی و تشریح خطا، راه‌حل ارائه‌شده با استفاده از توضیحات خطا ویرایش می‌شود. با توجه به مشخص‌شدن تلاش واقعی پروژه پس از تکمیل آن، راه‌حل ویرایش‌شده قطعی است و نیاز به بازبینی مجدد ندارد.

گام پنجم: ذخیره‌سازی

گام پنجم، به ذخیره‌سازی اطلاعات مفید راه‌حل بازبینی‌شده، در موارد پیشین می‌پردازد. این گام شامل انتخاب اطلاعات مفید پروژه‌ی جدید و چگونگی ذخیره‌سازی آن در ساختار حافظه است. اطلاعات پروژه به عنوان مسئله، و هزینه‌ی مورد نیاز توسعه‌اش به عنوان راه‌حل، اطلاعات مفید پروژه‌ی جدید هستند. با این حال، اطلاعات راه‌حل‌های اشتباه، جهت تفکیک راه‌حل‌های درست و غلط و محاسبه‌ی مجموع خطاها مفید هستند. به علاوه، سیستم هنگام مواجهه با یک خطا می‌تواند از خطاهای مشابه پیشین، جهت اصلاح راه‌حل جدید استفاده کند. مسائل و راه‌حل‌های ارائه‌شده‌ی پیشین، بخشی از دانش عمومی سیستم استدلال مبتنی بر قالب هستند. اگر مجموعه داده‌ی مورد آزمایش دارای n پروژه باشد، اطلاعات بازبینی‌شده‌ی پروژه‌ی جدید، به عنوان $n+1$ اُمین پروژه، در بخش دانش عمومی ذخیره می‌شود. دانش به دست آمده از مسئله‌ی جدید، جهت بهبود راه‌حل‌های ارائه‌شده‌ی سیستم در مسائل آتی مورد استفاده قرار می‌گیرد.

۴- نتایج و ارزیابی

در این بخش، به پیاده‌سازی روش پیشنهادی و ارزیابی نتایج آن با استفاده از مقایسه با دو مقاله‌ی مبنای تحقیق، پرداخته میشود. سیستم استدلال مبتنی بر قالب که روش برآورد هزینه داده گذر، مبتنی بر آن است، با بهره‌گیری از دانش عمومی‌اش به حل مسائل جدید بر اساس مسائل مشابه پیشین می‌پردازد. تجربیات موفق و ناموفق این سیستم در ارائه‌ی راه‌حل مسائل جدید، در بخش دانش عمومی ذخیره شده و در مسائل آتی جهت بهبود خطا مورد استفاده قرار می‌گیرد. با این حال، سیستم‌ها و ابزار استدلال مبتنی بر قالب، در مهمترین گام که بازیابی موارد مشابه مسئله‌ی جدید است، مانند یک جعبه سیاه عمل می‌کنند. این تحقیق، با بهره‌برداری از گام بازیابی به کمک جستجوی ممنوعه، پیکربندی جدیدی برای استدلال مبتنی بر قالب ارائه کرده است که در این بخش، مورد آزمایش قرار می‌گیرد.

جهت ارزیابی روش برآورد هزینه دادگذر، بایستی قدرت پیش‌بینی آن اندازه‌گیری شود. قدرت پیش‌بینی، بیانگر دقت این روش در برآورد هزینه‌ی پروژه‌ی هدف است که با فاصله‌ی هزینه‌ی پیش‌بینی‌شده و هزینه‌ی واقعی، رابطه‌ی معکوس دارد. بنابراین، هرچه فاصله‌ی هزینه‌ی پیش‌بینی‌شده و هزینه‌ی واقعی بیشتر شود، از قدرت پیش‌بینی کاسته خواهد شد. در این تحقیق، شاخص‌های اندازه‌گیری قدرت پیش‌بینی، میانگین خطای نسبی، میانگین خطای مطلق، میانه‌ی خطای نسبی و سطح پیش‌بینی هستند.

جهت اندازه‌گیری دقت پیش‌بینی، زیرمجموعه‌ای به نام مجموعه‌ی آموزشی از مجموعه داده‌ی مورد آزمایش، انتخاب می‌شود. سپس، روش برآورد هزینه داده گذر با استفاده از مجموعه‌ی آموزشی متناسب‌سازی شده و زیرمجموعه‌ی دیگری به نام مجموعه‌ی ارزیابی از مجموعه داده، انتخاب می‌شود. از آنجا که این روش با مجموعه‌ی آموزشی متناسب‌سازی می‌شود، تناسب آن با زیرمجموعه‌های آموزشی و ارزیابی، متفاوت خواهد بود. این تفاوت در مجموعه داده‌های کوچکتر، چشمگیرتر می‌شود که عدم اطمینان به ارزیابی روش ارائه‌شده را به دنبال خواهد داشت. جهت مقابله با این مشکل، ارزیابی روش برآورد هزینه داده گذر با استفاده از روش اعتبارسنجی متقابل[1] انجام شده است.

این تحقیق، جهت ارزیابی مجموعه داده‌های دارای کمتر از ۴۰ پروژه، از روش اعتبارسنجی متقابل LOO[2] استفاده کرده است. این روش جامع است و تمامی n حالت انتخاب زیرمجموعه‌ی ارزیابی یک پروژه‌ای، از مجموعه داده‌ی n پروژه‌ای را بررسی می‌کند. در هر یک از حالت‌ها، یک پروژه به عنوان مجموعه ارزیابی و $n-1$ پروژه‌ی باقیمانده به عنوان مجموعه‌ی آموزشی در نظر گرفته می‌شوند. سپس، دقت پیش‌بینی برای تمامی حالت‌ها بر اساس شاخص ارزیابی محاسبه شده و میانگین آنها به عنوان قدرت پیش‌بینی روش برآورد هزینه داده گذر در مجموعه داده‌ی مورد آزمایش، ارائه می‌شود.

همچنین، ارزیابی روش برآورد هزینه داده گذر در مجموعه داده‌های بزرگتر از ۴۰ پروژه با استفاده از روش اعتبارسنجی متقابل 10-Fold[3] انجام شده است. روش 10-Fold، غیر جامع است و مجموعه داده را به ۱۰ زیرمجموعه‌ی ارزیابی با تعداد پروژه‌های یکسان، تقسیم می‌کند. زیرمجموعه‌های ارزیابی به صورت تصادفی انتخاب می‌شوند. هر زیرمجموعه‌ی ارزیابی در روش 10-Fold، دارای یک زیرمجموعه‌ی آموزشی متناظر است. اگر مجموعه داده‌ی مورد آزمایش، مجموعه‌ی مرجع باشد، متمم هر یک از زیرمجموعه‌های ارزیابی، زیرمجموعه‌ی آموزشی متناظرشان است. دقت پیش‌بینی، به ازای هر یک از زیرمجموعه‌های ارزیابی محاسبه شده و میانگین آن به عنوان قدرت پیش‌بینی روش برآورد هزینه داده گذر در مجموعه داده‌ی مورد آزمایش، در نظر گرفته می‌شود.

نرم‌افزار R، دارای طیف گسترده‌ای از توابع پیش‌فرض و بسته‌های آماده جهت انجام محاسبات آماری است. همین امر، پیاده‌سازی سناریو آزمایش‌های تحقیق را تسهیل نموده است. سناریوی آزمایش‌های این تحقیق، در قالب ۱۷ تابع و ۸۳۱ خط کد به کمک چهار بسته‌ی آماده در

[1] cross-validation
[2] Leave-One-Out cross validation
[3] 10-Fold cross validation

نرم‌افزار R، پیاده‌سازی شده است. بسته‌های فیلِنترُپی[1]، دی‌پلِیر[2]، اُپن‌ایکس‌ال‌اِس‌ایکس[3] و اِف‌ای‌آر‌اِف‌اِف[4]، چهار بسته‌ی آماده‌ای هستند که جهت پیاده‌سازی روش برآورد هزینه داده گذر، مورد استفاده قرار گرفته‌اند. بسته‌ی فیلِنترُپی، در بر گیرنده‌ی تابع euclidean است که در گام دوم این روش جهت محاسبه‌ی فاصله‌ی بردار پروژه‌ی هدف با بردار پروژه‌های پیشین، استفاده شده است.

مجموعه داده‌ی مورد آزمایش، پس از بازگشایی توسط بسته‌ی اِف‌ای‌آر‌اِف‌اِف و ورود به نرم‌افزار R، در قالب یک چهارچوب داده[5]، نگهداری می‌شود. بسته‌ی دی‌پلِیر نرم‌افزار R، متمرکز بر کار با چهارچوب‌های داده است. دی‌پلِیر در گام دوم روش برآورد هزینه داده گذر، وظیفه‌ی انتخاب ویژگی از مجموعه داده‌ی مورد آزمایش را بر اساس راه‌حل ارائه‌شده توسط جستجوی ممنوعه، عهده‌دار است. همچنین، بسته‌ی اُپن‌ایکس‌ال‌اِس‌ایکس، جهت ذخیره‌سازی نتایج آزمایش‌ها در قالب فایل اِکسِل مورد استفاده قرار گرفته است. جهت پیاده‌سازی سایر بخش‌های روش برآورد هزینه داده گذر، از توابع پیش‌فرض، شرط‌ها، حلقه‌ها و عملگرهای آرایه و ماتریس تعبیه‌شده در نرم‌افزار R، استفاده شده است.

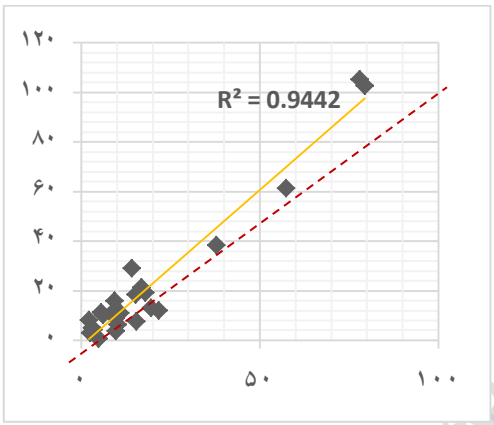
مجموعه داده‌ی آلبرخت از ۲۴ پروژه با میانگین تلاش ۲۱/۸۷ هزار نفر–ساعت، تشکیل شده است. ارزیابی روش پیشنهادی تحقیق توسط این مجموعه داده با استفاده از روش ارزیابی متقابل LOO انجام شده است. آلبرخت با ۸ ویژگی، دارای ۱۲۷ زیرمجموعه‌ی کاندید است. جدول ۴، قدرت پیش‌بینی روش پیشنهادی تحقیق در مجموعه داده‌ی آلبرخت را به ازای هر یک از سناریوها نشان می‌دهد.

جدول ۴. نتایج مجموعه داده‌ی آلبرخت

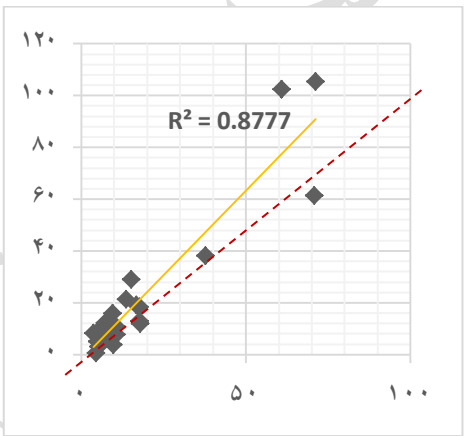
سناریو	MMRE	MdMRE	MAE	PRED(۰/۲۵)
۱	۰/۴۸۹۱	۰/۳۴۵۴	۶/۰۳۸۸	۰/۳۳۳۳
۲	۰/۴۱۴۸	۰/۳۹۱۵	۶/۷۷۶۳	۰/۳۳۳۳
۳	۰/۴۸۹۱	۰/۳۴۵۴	۶/۰۳۸۸	۰/۳۳۳۳
۴	۰/۴۱۴۸	۰/۳۹۱۵	۶/۷۷۶۳	۰/۳۳۳۳
۵	۰/۴۸۹۱	۰/۳۴۵۴	۶/۰۳۸۸	۰/۳۳۳۳
۶	۰/۴۱۴۸	۰/۳۹۱۵	۶/۷۷۶۳	۰/۳۳۳۳
۷	۰/۴۸۹۱	۰/۳۴۵۴	۶/۰۳۸۸	۰/۳۳۳۳
۸	۰/۴۱۴۸	۰/۳۹۱۵	۶/۷۷۶۳	۰/۳۳۳۳

هر یک از سطرهای جدول ۴ نشان‌دهنده‌ی قدرت پیش‌بینی سناریوهای تحقیق در مجموعه داده‌ی آلبرخت بر اساس شاخص‌های ارزیابی MMRE، MdMRE، MAE و PRED(۰/۲۵) هستند. دو قدرت پیش‌بینی متفاوت از اجرای سناریوها روی مجموعه داده‌ی آلبرخت به دست آمده است. قدرت پیش‌بینی اول مربوط به سناریوهای ۱، ۳، ۵ و ۷ (گروه آ) و قدرت پیش‌بینی دوم مربوط به سناریوهای ۲، ۴، ۶ و ۸ (گروه ب) است. یکسان بودن نتایج سناریوهای ۱ و ۳، ۲ و ۴، ۵ و ۷، ۶ و ۸ ناشی از عدم وجود مقادیر گمشده در مجموعه داده‌ی آلبرخت است.

بر اساس شاخص MMRE، سناریوهای گروه (ب)، عملکرد بهتری نسبت به سناریوهای گروه (آ) داشته‌اند. با این حال، از نظر شاخص‌های MdMRE و MAE، برتری ازآن سناریوهای گروه (آ) است. همچنین، هر دو گروه، نتایج یکسانی را بر اساس شاخص PRED(۰/۲۵) به دست آورده‌اند. بنابراین، با توجه به برابری دو گروه در شاخص ارزیابی سطح پیش‌بینی و پایین‌تر بودن میانگین خطای مطلق و میانه‌ی خطای نسبی گروه (آ) نسب به گروه (ب)، سناریوهای ۱، ۳، ۵ و ۷ نتایج دقیق‌تری ارائه کرده‌اند. رابطه‌ی تلاش واقعی پروژه‌های مجموعه داده‌ی آلبرخت با تلاش برآوردشده توسط سناریوهای گروه (آ) و گروه (ب) به ترتیب در شکل ۳ و شکل ۴ نشان داده شده است. همچنین، در شکل ۵ و شکل ۶ به ترتیب نمودار جعبه‌ای خطای نسبی و نمودار توزیع فاصله سناریوهای گروه (آ) و گروه (ب)، نشان داده شده است.



شکل ۳. نمودار نقطه‌ای سناریوهای ۱، ۳، ۵ و ۷ مجموعه داده آلبرخت



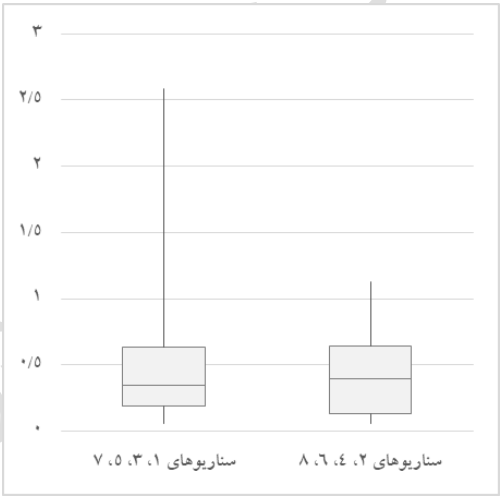
شکل۴. نمودار نقطه‌ای سناریوهای ۲، ۴، ۶ و ۸ مجموعه داده آلبرخت

در شکل ۳ و شکل ۴، محور x به تلاش‌های برآوردشده و محور y به تلاش‌های واقعی اختصاص داده شده است. همچنین، خط‌چین‌های قرمز، خطوط رگرسیون نمودارها هستند که مکان آرمانی قرارگیری نقاط را نشان می‌دهند. خطوط زرد رنگ نیز خطوط روند نمودارها هستند و بیانگر گرایش نقاط نمودار هستند. در نمودار نقطه‌ای، نزدیکی نقاط به خط

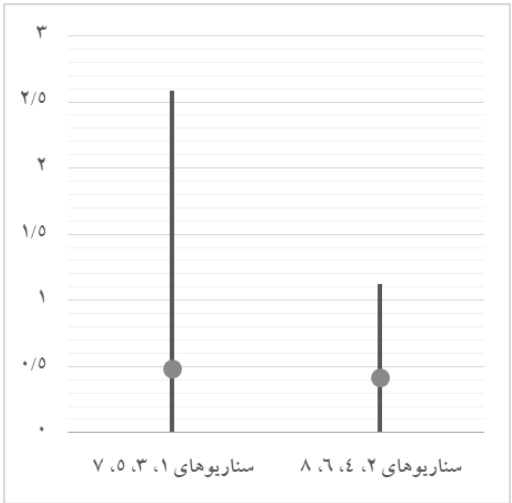
[1] philentropy
[2] dplyr
[3] openxlsx
[4] farff
[5] data frame

رگرسیون و حداقل بودن زاویه‌ی تشکیل‌شده بین خط رگرسیون و خط روند، مورد انتظار است. همچنین، شاخص ضریب تعیین[1] (R^2)، دیگر معیار ارزیابی در نمودار نقطه‌ای است که میزان همبستگی تلاش واقعی و تلاش پیش‌بینی‌شده را نشان می‌دهد. زاویه‌ی تشکیل‌شده بین خط روند و خط رگرسیون شکل ۳، نسبت به شکل ۴ کوچکتر است.

همچنین، نقاط شکل ۳ در مقایسه با شکل ۴ به خط رگرسیون نزدیکتر هستند و مقدار شاخص R^2 در شکل ، نسبت به شکل ۴ دارای مقدار بیشتری است. بنابراین، بر اساس معیارهای ارزیابی نمودار نقطه‌ای، می‌توان نتیجه گرفت که سناریوهای ۱، ۳، ۵ و ۷ در مقایسه با سناریوهای ۲، ۴، ۶ و ۸ عملکرد بهتری ارائه نموده‌اند.



شکل ۵. نمودار جعبه‌ای خطای نسبی مجموعه داده‌ی آلبرخت



شکل ۶. نمودار توزیع فاصله‌ی خطای نسبی مجموعه داده‌ی آلبرخت

شکل ۵، چگونگی توزیع خطای نسبی برآوردهای به دست آمده از سناریوهای گروه (آ) و (ب) را بین چارک‌های اول تا چهارم در قالب نمودار جعبه‌ای به تصویر کشیده است. در این تصویر که محور y آن نشان‌دهنده‌ی خطای نسبی است، نمودار سمت چپ به سناریوهای گروه (آ) و نمودار سمت راست به سناریوهای گروه (ب) اختصاص دارد. مشاهدات شکل، عملکرد مثبت هر دو گروه و گرایش خطای نسبی آنها به سمت صفر را نشان می‌دهد. همچنین، علی‌رغم کشیدگی بیشتر نمودار سناریوهای گروه (آ)، همپوشانی جعبه‌های دو نمودار، تشابه نتایج آنها را نشان می‌دهد. با این حال، خط میانه و بیشینه‌ی مؤثر نمودار سناریوهای گروه (آ) نسبت به نمودار سناریوهای گروه (ب) به صفر نزدیکتر است که قدرت پیش‌بینی بیشتر سناریوهای گروه (آ) را نشان می‌دهد.

در شکل ۶، تنوع نتایج به دست آمده از دو گروه (آ) و (ب) با استفاده از نمودار توزیع فاصله نشان داده شده است که در آن، نمودار سمت چپ به گروه (آ) و نمودار سمت راست به گروه (ب)، تعلق دارد. محور y این نمودار نشان‌دهنده‌ی خطای نسبی است و بلندی طول آن تنوع بیشتر نتایج را نشان می‌دهد. همچنین، نقاط درج‌شده روی نمودار، نشان‌دهنده‌ی میانگین خطای نسبی گروه مربوطه است. مشاهدات شکل ۶، تنوع نتایج سناریوهای گروه (آ) و ثبات نتایج سناریوهای گروه (ب) را نشان می‌دهد. به علاوه، میانگین خطای نسبی سناریوهای گروه (ب)، پایین‌تر از میانگین خطای نسبی سناریوهای گروه (آ) قرار گرفته است که بر دقت بیشتر نتایج سناریوهای گروه (ب) دلالت دارد.

با استناد مشاهدات جدول ۴ و بررسی نقطه به نقطه‌ی نتایج دو گروه در شکل ۵ و شکل ۶، سناریوهای گروه (آ) برآورد دقیق‌تری نسبت به سناریوهای گروه (ب) ارائه کرده‌اند. برتری نسبی سناریوهای گروه (ب) در شاخص ارزیابی MMRE، به دلیل ضعف گروه (ب) در شکل ۵ بوده و شاخص‌های ارزیابی MdMRE وMAE، قابل اطمینان نیستند. بنابراین در مجموع، سناریوهای گروه (آ) که متشکل از سناریوهای شماره‌ی ۱، ۳، ۵ و ۷ هستند، دقیق‌ترین نتایج را با استفاده از روش برآورد هزینه داده گذر در مجموعه داده‌ی آلبرخت، ارائه کرده‌اند. در ادامه، نتایج سناریوهای روش برآورد هزینه داده گذر در مجموعه داده‌های تحقیق با نتایج روشهای دیگر، مقایسه و ارزیابی می‌گردد. از این رو، نتیجه‌ی دقیق‌ترین سناریو به ازای هر مجموعه داده، جهت مقایسه با مقالات مبنا انتخاب شده است.

وو و همکارانش [۶۱] با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات به وزن‌دهی ویژگی‌ها پرداخته‌اند و ۳ مدل بر اساس استدلال مبتنی بر قالب ارائه کرده‌اند. تفاوت مدل‌های وو و همکارانش در شاخص اندازه‌گیری شباهت است. این سه مدل، از شاخص‌های اندازه‌گیری شباهت فاصله‌ی اقلیدسی، فاصله منهتن و درجه رابطه خاکستری استفاده می‌کنند. همچنین، هر یک از سه مدل، دارای دو حالت وزن‌دهی‌شده و بدون وزن هستند. روش برآورد هزینه داده گذر با مدل‌های وزن‌دهی‌شده‌ی وو که دارای برآوردهای دقیق‌تری هستند، ارزیابی شده است. جدول ۵، جدول ۶ و جدول ۷ به ترتیب نتایج مدل‌های WEuc-CBR، WMan-CBR و WGre-CBR وو را به همراه نتایج روش برآورد هزینه داده گذر تحقیق نشان می‌دهند.

جدول ۵. نتایج مدل WEuc-CBR وو و روش برآورد هزینه داده گذر

مجموعه	مدل WEuc-CBR وو	روش پیشنهادی تحقیق

[1] coefficient of determination

داده	MMRE	MdMRE	PRED(۰/۲۵)	MMRE	MdMRE	PRED(۰/۲۵)
دِشارنایز	۰/۷۲۴	۰/۴۲۱	۰/۳۰۳	۰/۳۳۳۴	۰/۲۵۵۵	۰/۴۸۵۷
مَکسول	۰/۷۳۲	۰/۳۹۸	۰/۳۱۷	۰/۴۰۴۲	۰/۳۲۴۳	۰/۳۸۳۳

جدول ۶. نتایج مدل WMan-CBR وو و روش برآورد هزینه داده گذر

مجموعه داده	مدل WMan-CBR وو			روش پیشنهادی تحقیق		
	MMRE	MdMRE	PRED(۰/۲۵)	MMRE	MdMRE	PRED(۰/۲۵)
دِشارنایز	۰/۶۸۶	۰/۳۵۸	۰/۳۷۲	۰/۳۳۳۴	۰/۲۵۵۵	۰/۴۸۵۷
مَکسول	۰/۵۴۴	۰/۳۸۹	۰/۳۰۲	۰/۴۰۴۲	۰/۳۲۴۳	۰/۳۸۳۳

جدول ۷. نتایج مدل WGre-CBR وو و روش برآورد هزینه داده گذر

مجموعه داده	مدل WGre-CBR وو			روش پیشنهادی تحقیق		
	MMRE	MdMRE	PRED(۰/۲۵)	MMRE	MdMRE	PRED(۰/۲۵)
دِشارنایز	۰/۵۷۳	۰/۴۰۳	۰/۳۵۶	۰/۳۳۳۴	۰/۲۵۵۵	۰/۴۸۵۷
مَکسول	۰/۵۳۰	۰/۴۳۸	۰/۳۱۷	۰/۴۰۴۲	۰/۳۲۴۳	۰/۳۸۳۳

در جدول ۵ تا ۷، نتایج مدل‌های وو و نتیجه‌ی برترین سناریوی روش برآورد هزینه داده گذر در مجموعه داده‌های دِشارنایز و مَکسول، نشان داده شده است. نتایج این جداول بر اساس شاخص‌های ارزیابی MMRE، MdMRE و PRED(۰/۲۵) می‌باشند. نتایج روش برآورد هزینه داده گذر این جداول در مجموعه داده‌ی دِشارنایز، متعلق به سناریوهای ۱، ۲، ۵ و ۶ و در مجموعه داده‌ی مَکسول، متعلق به سناریوهای ۵ و ۷ است.

دقیق‌ترین مدل وو در مجموعه داده‌ی دِشارنایز بر اساس شاخص‌های MdMRE و PRED(۰/۲۵)، مدل WMan-CBR است. بر اساس شاخص MMRE نیز، عملکرد مدل WGre-CBR دقیق‌تر است. بنابراین، مدل‌های WMan-CBR و WGre-CBR، دو مدل برتر وو در مجموعه داده‌ی دِشارنایز هستند. نتایج حاکی از برتری قاطع روش برآورد هزینه داده گذر بر اساس تمامی شاخص‌ها، نسبت به سه مدل وو در مجموعه داده‌ی دِشارنایز است. برترین نتایج مدل‌های وو در مجموعه داده‌ی دِشارنایز با زیر خط، نشانه‌گذاری شده‌اند.

در مجموعه داده‌ی مَکسول، مدل‌های WGre-CBR و WEuc-CBR، بر اساس شاخص PRED(۰/۲۵) دارای عملکرد مشابه هستند و بر مدل WMan-CBR برتری دارند. همچنین، بر اساس شاخص‌های MdMRE و MMRE، برتری به ترتیب به مدل‌های WMan-CBR و WGre-CBR اختصاص دارد. با این حال، جداول ۴ تا ۶ در مجموعه داده‌ی مَکسول نیز، برتری قاطع روش برآورد هزینه داده گذر را نسبت به مدل وو نشان می‌دهند. برترین نتایج به دست آمده از مدل‌های وو در مجموعه داده‌ی مَکسول با زیر خط در جداول مذکور نشانه‌گذاری شده است. جدول ۸ به مقایسه‌ی نتایج روش برآورد هزینه داده گذر با برترین نتایج مدل‌های وو در مجموعه داده‌ی مَکسول پرداخته است.

جدول ۸ مقایسه‌ی نتایج برتر مدل‌های وو با روش پیشنهادی

مجموعه داده	میزان اختلاف			درصد بهبود		
	MMRE	MdMRE	PRED(۰/۲۵)	MMRE	MdMRE	PRED(۰/۲۵)
دِشارنایز	۰/۲۳۹۶	۰/۱۰۲۵	۰/۱۱۳۷	۴۱/۸۱	۲۸/۶۳	۳۰/۵۶
مَکسول	۰/۱۲۵۸	۰/۰۶۴۷	۰/۰۶۶۳	۲۳/۷۳	۱۶/۶۳	۲۰/۹۱

جدول ۸ به نمایش میزان اختلاف و درصد بهبود نتایج روش برآورد هزینه داده گذر نسبت به مدل‌های وو بر اساس شاخص‌های ارزیابی MMRE، MdMRE و PRED(۰/۲۵)، پرداخته است. سطر اول این جدول به مجموعه داده‌ی دِشارنایز و سطر دوم آن به مجموعه داده‌ی مَکسول اختصاص دارد. میزان اختلاف و درصد بهبود در تمامی شاخص‌ها و مجموعه داده‌ها، مثبت است. مثبت‌بودن میزان اختلاف و درصد بهبودی، نشان دهنده برتری روش برآورد هزینه داده گذر، نسبت به مدل‌های ارائه‌شده توسط وو و همکارانش است.

لیو و همکارانش [۶۲] با هدف بهبود استدلال مبتنی بر قالب، به طراحی یک روش حریصانه‌ی انتخاب ویژگی بر اساس واحد اندازه‌گیری اطلاعات متقابل همسایگی بومی‌شده، پرداخته‌اند. مدل لیو، روی مجموعه داده‌های آلبرخت، چاینا، دِشارنایز، کِمِرِر، کیچنهام، و میازاکی آزمایش شده است. جدول ۹، نتایج به دست آمده از مدل لیو و روش برآورد هزینه داده گذر را نشان داده است.

جدول ۹. نتایج مدل لیو و روش برآورد هزینه داده گذر

مجموعه داده	مدل لیو		روش پیشنهادی تحقیق	
	MMRE	MAE	MMRE	MAE
آلبرخت	۰/۸۷	۱۰/۰۴	۰/۴۸۹۱	۶/۰۳۸۸
چاینا	۰/۱۱	۴۱۴	۰/۰۹۱۴	۳۸۱/۴۰۰۳
دِشارنایز	۰/۳۷	۱۷۶۶	۰/۳۳۳۴	۱۶۴۶/۷۰۷۱
کِمِرِر	۰/۵۳	۱۰۹	۰/۴۳۴۰	۱۱۲/۴۵۷۶
کیچنهام	۰/۲۸	۱۳۰۵	۰/۲۶۴۴	۶۲۹/۶۹۲۸
میازاکی	۰/۳۴	۴۹/۶	۰/۲۷۳۴	۱۵/۴۳۶۲

در جدول ۹، نتایج مدل لیو و نتیجه‌ی برترین سناریوی روش برآورد هزینه داده گذر در مجموعه داده‌های آلبرخت، چاینا، دِشارنایز، کِمِرِر، کیچنهام، و میازاکی، نشان داده شده است. نتایج جدول ۹ بر اساس شاخص‌های ارزیابی MMRE و MAE می‌باشد. نتایج روش برآورد هزینه داده گذر در مجموعه داده‌ی آلبرخت متعلق به سناریوهای ۱، ۳، ۵ و ۷ و در مجموعه داده‌ی چاینا متعلق به سناریوهای ۲، ۴، ۶ و ۸ است. همچنین، نتایج این روش در مجموعه داده‌ی دِشارنایز متعلق به سناریوهای ۱، ۲، ۵ و ۶ و در مجموعه داده‌ی کیچنهام متعلق به سناریوهای ۳ و ۷ می‌باشد. نتایج سناریوهای ۱ تا ۸ نیز در مجموعه داده‌های کِمِرِر و میازاکی یکسان است که از آنها به عنوان نتایج روش پیشنهادی جدول استفاده شده است.

شاخص‌های MMRE و MAE جدول برتری قابل توجه روش برآورد هزینه داده گذر را نسبت به مدل لیو در مجموعه داده‌های آلبرخت و میازاکی نشان می‌دهد. همچنین، روش برآورد هزینه داده گذر در مقایسه با مدل لیو در مجموعه داده‌های چاینا، دِشارنایز و کیچنهام بر اساس هر دو شاخص MMRE و MAE دقیق‌تر عمل کرده است. با این حال در مجموعه داده‌ی کِمِرِر، شاخص MAE برتری مدل لیو و شاخص MMRE برتری روش برآورد هزینه داده گذر را نشان می‌دهند. جدول ۱۰ به مقایسه‌ی نتایج روش برآورد هزینه داده گذر و نتایج مدل لیو در مجموعه داده‌های آلبرخت، چاینا، دِشارنایز، کِمِرِر، کیچنهام، و میازاکی پرداخته است. در این نتایج، میزان اختلاف و درصد بهبود نتایج روش برآورد هزینه داده گذر نسبت به مدل لیو بر اساس شاخص‌های ارزیابی MMRE و MAE نشان داده شده است.

جدول ۱۰. مقایسه‌ی نتایج برتر مدل‌های لیو با روش برآورد هزینه داده گذر

مجموعه داده	میزان اختلاف		درصد بهبود	
	MMRE	MAE	MMRE	MAE
آلبرخت	۰/۳۸۰۹	۴/۰۰۱۲	۴۳/۷۸	۳۹/۸۵
چاینا	۰/۰۱۸۶	۳۲/۵۹۹۷	۱۶/۹۰	۷/۸۷
دِشارنایز	۰/۰۳۶۶	۱۱۹/۲۹۲۹	۹/۸۹	۶/۷۵
کِمِرِر	۰/۰۹۶	-۳/۴۵۷۶	۱۸/۱۱	-۳/۱۷
کیچنهام	۰/۰۱۵۶	۶۷۵/۳۰۷۲	۵/۵۷	۵۱/۷۴
میازاکی	۰/۰۶۶۶	۳۴/۱۶۳۸	۱۹/۵۸	۶۸/۸۷

سطرهای اول تا ششم جدول ۱۰ به ترتیب به مجموعه داده‌های آلبرخت، چاینا، دِشارنایز، کیچنهام، کِمِرِر، و میازاکی اختصاص دارند. همانطور که در جدول نشان داده شده است، میزان اختلاف و درصد بهبود تمامی شاخص‌ها و مجموعه داده‌ها، به جز شاخص MAE مجموعه داده‌ی کِمِرِر، مثبت است. با این حال، برتری ۳/۴۵۷۶ واحدی مدل لیو نسبت به روش برآورد هزینه داده گذر، با توجه به میانگین تلاش ۲۱۹/۲۴ نفر-ماه پروژه‌های کِمِرِر، قابل چشم‌پوشی است. همچنین، این روش نسبت به مدل لیو در مجموعه داده‌ی کِمِرِر، بر اساس شاخص تعیین‌کننده‌ی MMRE، ۱۸/۱۱٪ بهتر عمل کرده است. بنابراین، نتایج ارائه شده در جدول ۱۰، بر برتری روش برآورد هزینه داده گذر نسبت به مدل لیو دلالت دارند.

با توجه به ارزیابی‌های صورت گرفته، روش برآورد هزینه داده گذر در مقایسه با مدل وو در تمامی مجموعه داده‌ها و شاخص‌های ارزیابی، دارای برتری قاطع است. در مقایسه با مدل لیو، گرچه بر اساس شاخص ارزیابی MAE در مجموعه داده‌ی کِمِرِر، لیو از روش برآورد هزینه داده گذر به میزان ۳/۱۷٪ بهتر عمل کرده است، ولی شاخص MMRE در مجموعه داده‌ی کِمِرِر، برتری ۱۸/۱۱٪ روش برآورد هزینه داده گذر را نسبت به مدل لیو نشان می‌دهد. همچنین، برتری ۳/۴۵۷۶ واحدی مدل لیو نسبت به روش برآورد هزینه داده گذر با توجه به میانگین ۲۱۹/۲۴ نفر-ماه تلاش مجموعه داده‌ی کِمِرِر، قابل چشم‌پوشی است. به علاوه، در سایر شاخص‌های ارزیابی و مجموعه داده‌ها، روش برآورد هزینه داده گذر نسبت به مدل لیو نیز دارای برتری است.

۵- نتیجه‌گیری

این تحقیق با هدف افزایش دقت برآورد هزینه‌ی نرم‌افزار و بهبود شاخص‌های ارزیابی، روشی بر پایه‌ی استدلال مبتنی بر قالب ارائه کرده است. اساس کار این تحقیق، پیکربندی پارامترهای شش‌گانه‌ی استدلال مبتنی بر قالب است. پیکربندی شش پارامتر روش استدلال مبتنی بر قالب بر دقت برآورد هزینه‌ی نرم‌افزار این روش اثرگذار بوده است. با توجه به نتایج به دست آمده از روش برآورد هزینه داده گذر و مدل‌های وو و لیو که هر یک پیکربندی متفاوتی از روش استدلال مبتنی بر قالب هستند، اثرگذاری پیکربندی در نتایج استدلال مبتنی بر قالب، بدیهی است. میزان اثرگذاری مشاهده‌شده در این تحقیق با استفاده از شاخص‌های میانگین و میانه‌ی خطای نسبی، میانگین خطای مطلق و سطح پیش‌بینی، اندازه‌گیری شده است. در این تحقیق، بر اساس شاخص‌های MMRE، MdMRE، MAE، و PRED(۰/۲۵)، به ترتیب ۲۲/۴۲٪، ۲۲/۶۳٪، ۲۵/۷۳٪، و ۲۸/۶۵٪، اثرگذاری مشاهده شده است.

همچنین بهره‌گیری از جستجوی ممنوعه بر افزایش دقت برآورد هزینه‌ی نرم‌افزار ارائه‌شده‌ی روش استدلال مبتنی بر قالب مؤثر است. مدل‌های وو و لیو جهت مقایسه و ارزیابی کارایی در نظر گرفته شده‌اند. نتایج حاکی از آن است که بهره‌گیری از جستجوی ممنوعه نسبت به مدل وو، بر اساس شاخص‌های MMRE، MdMRE، و PRED(۰/۲۵)، به ترتیب ۳۲/۷۷٪، ۲۲/۶۳٪، و ۲۸/۶۵٪ دقت برآوردها را افزایش داده است. همچنین، بهره‌گیری از جستجوی ممنوعه، دقت برآوردها را بر اساس شاخص‌های MMRE و MAE، به ترتیب ۱۸/۹۸٪ و ۲۵/۷۳٪ نسبت به مدل لیو، افزایش داده است.

یکی از پارامترهای متغیر سناریوهای تحقیق، پارامتر شاخص ارزیابی است که در تابع هدف مورد استفاده قرار گرفته است. مشاهدات، اثرگذاری این پارامتر را در مجموعه داده‌های چاینا، دِشارنایز، کیچنهام، و مَکسول نشان می‌دهند. در مقابل، نتایج مجموعه داده‌های آلبرخت، کِمِرِر، و میازاکی، بی‌تأثیر بودن این پارامتر را نشان داده است. بنابراین، پارامتر شاخص ارزیابی در استدلال مبتنی بر قالب، وابسته به مجموعه داده‌ی مورد آزمایش می‌باشد و دارای اثرگذاری نسبی است.

در عین حال، به کارگیری مکانیزم راه‌حل اولیه‌ی متفاوت در جستجوی ممنوعه، در بهبود دقت برآورد هزینه‌ی نرم‌افزار ارائه‌شده توسط روش استدلال مبتنی بر قالب مؤثر است. پارامتر متغیر مکانیزم انتخاب راه‌حل اولیه در سناریوهای تحقیق، تعبیه شده است. مشاهدات، تأثیرگذاری این پارامتر را در نتایج مجموعه داده‌ی مَکسول نشان می‌دهد. وجه تمایز مجموعه داده‌ی مَکسول در مقایسه با دیگر مجموعه داده‌ها، تعداد بالای زیرمجموعه‌های کاندید آن است. از آنجا که این پارامتر بر سایر مجموعه داده‌های تحقیق بی‌تأثیر بوده است، به‌کارگیری مکانیزم انتخاب راه‌حل اولیه‌ی متفاوت در جستجوی ممنوعه، تنها در مجموعه داده‌هایی که دارای بیش از ۱۹ ویژگی هستند، مؤثر است.

بهبود هرچه بیشتر کیفیت برآوردها در راستای افزایش دقت و کاهش مدت زمان مورد نیاز پیش‌بینی، و وابستگی به اطلاعات پروژه‌ی هدف، مورد انتظار است. به همین منظور، مواردی نظیر بررسی میزان اثرگذاری دیگر الگوریتم‌های اکتشافی و تکنیک‌های انتخاب ویژگی در استدلال مبتنی بر قالب، بررسی میزان اثرگذاری دیگر شاخص‌های اندازه‌گیری شباهت در استدلال مبتنی بر قالب، کشف مکانیزمی جهت انتخاب تعداد بهینه‌ی موارد مشابه متناسب با مجموعه داده‌های مورد آزمایش در استدلال مبتنی بر قالب، ارائه‌ی راهکاری بهینه جهت تطبیق تشابه راه‌حل‌های بازیابی‌شده با مسئله‌ی هدف در استدلال مبتنی بر قالب، جهت تحقیقات آینده پیشنهاد می‌گردند.

مراجع

1. Oliveira, A.L.I., *Estimation of software project effort with support vector regression*. Neurocomputing, 2006. **69**(13): p. 1749-1753.
2. Saroha, M. and S. Sahu. *Tools & methods for software effort estimation using use case points model — A review*. in *International Conference on Computing, Communication & Automation*. 2015.
3. Basten, D. and W. Mellis. *A Current Assessment of Software Development Effort Estimation*. in *2011 International Symposium on Empirical Software Engineering and Measurement*. 2011.
4. Swati, W. and K. Kishor, *A Novel Way of Cost Estimation in Software Project Development Based on Clustering Techniques.* A Novel Way of Cost Estimation in Software Project Development Based on Clustering Techniques, 2014. **2**(4): p. 7.
5. Abbas, S., et al., *Cost Estimation: A Survey of Well-known Historic Cost Estimation Techniques*. Vol. 3. 2012.
6. Boehm, B., C. Abts, and S. Chulani, *Software development cost estimation approaches — A survey.* Annals of Software Engineering, 2000. **10**(1): p. 177-205.
7. Dejaeger, K., et al., *Data Mining Techniques for Software Effort Estimation: A Comparative Study.* IEEE Transactions on Software Engineering, 2012. **38**(2): p. 375-397.
8. Borade, J. and V.R. Khalkar, *Software Project Effort and Cost Estimation Techniques*. Vol. 3. 2013. 730-739.
9. Azhar, D., E. Mendes, and P. Riddle, *A systematic review of web resource estimation*, in *Proceedings of the 8th International Conference on Predictive Models in Software Engineering*. 2012, ACM: Lund, Sweden. p. 49-58.
10. Bardsiri, V. and D. Jawawi, *Software Cost Estimation Methods: A Review*. Vol. 2. 2011. 21-29.
11. Mendes, E., N. Mosley, and S. Counsell. *The application of case-based reasoning to early Web project cost estimation*. in *Proceedings 26th Annual International Computer Software and Applications*. 2002.
12. Reifer, D.J., *Web development: estimating quick-to-market software.* IEEE Software, 2000. **17**(6): p. 57-64.
13. Kumar, P.S. and H. Behera, *Role of soft computing techniques in software effort estimation: an analytical study*, in *Computational Intelligence in Pattern Recognition*. 2020, Springer. p. 807-831.
14. Kadoda, G., M. Cartwright, and M. Shepperd. *Issues on the effective use of CBR technology for software project prediction*. Springer.
15. Pandey, M., R. Litoriya, and P. Pandey, *Validation of Existing Software Effort Estimation Techniques in Context with Mobile Software Applications.* Wireless Personal Communications, 2020. **110**(4): p. 1659-1677.
16. Du, W.L., et al., *A hybrid intelligent model for software cost estimation.* arXiv preprint arXiv:1512.00306, 2015.
17. Galorath, D.D. and M.W. Evans, *Software sizing, estimation, and risk management: when performance is measured performance improves*. 2006: Auerbach Publications.
18. Jang, J.S., *ANFIS: adaptive-network-based fuzzy inference system.* IEEE transactions on systems, man, and cybernetics, 1993. **23**(3): p. 665-685.
19. Algabri, M., et al. *Optimization of soft cost estimation using genetic algorithm for NASA software projects*. IEEE.
20. Boehm, B., et al., *Cost models for future software life cycle processes: COCOMO 2.0.* Annals of software engineering, 1995. **1**(1): p. 57-94.
21. Singal, P., A.C. Kumari, and P. Sharma, *Estimation of Software Development Effort: A Differential Evolution Approach.* Procedia Computer Science, 2020. **167**: p. 2643-2652.
22. Nassif, A.B., D. Ho, and L.F. Capretz. *Regression model for software effort estimation based on the use case point method*.
23. Karner, G., *Resource estimation for objectory projects.* Objective Systems SF AB, 1993. **17**: p. 1-9.
24. Winters, G.S.J.P., *Applying Use Cases: A Practical Guide*. 2 edition ed. 2001: Addison-Wesley Professional.
25. Corazza, A., et al., *Using tabu search to configure support vector regression for effort estimation.* Empirical Software Engineering, 2013. **18**(3): p. 506-546.
26. Hall, M., et al., *The WEKA data mining software: an update.* ACM SIGKDD explorations newsletter, 2009. **11**(1): p. 10-18.
27. Chang, C.-C. and C.-J. Lin, *LIBSVM: A library for support vector machines.* 2011. **2**(3 %J ACM Trans. Intell. Syst. Technol.): p. Article 27.
28. abdelali, Z., H. Mustapha, and N. Abdelwahed, *Investigating the use of random forest in software effort estimation.* Procedia Computer Science, 2019. **148**: p. 343-352.
29. Ziauddin, S.K.T. and S. Zia, *An effort estimation model for agile software development.* Advances in computer science and its applications (ACSA), 2012. **2**(1): p. 314-324.
30. Periyasamy, K. and A. Ghode. *Cost estimation using extended use case point (e-UCP) model*. IEEE.
31. Alwidian, J. and W. Hadi. *Enhancing the results of ucp in cost estimation using new external environmental factors*. IEEE.
32. Jha, P. and R.K. Malu, *Cost estimation using parameterized for use case point model (P-UCP)*, in *Int. Journal of Engineering Research and Applications*. 2013, Citeseer. p. 1280-1283.
33. Xia, T., et al., *Sequential model optimization for software effort estimation.* IEEE Transactions on Software Engineering, 2020.

and system approaches. AI communications, 1994. **7**(1): p. 39-59.
54. Albrecht, A.J. and J.E. Gaffney, *Software function, source lines of code, and development effort prediction: a software science validation.* IEEE transactions on software engineering, 1983(6): p. 639-648.
55. Menzies, T., et al., *Local versus Global Lessons for Defect Prediction and Effort Estimation.* IEEE Transactions on Software Engineering, 2013. **39**(6): p. 822-834.
56. Desharnais, J.-M., *Analyse Statistique de la Productivite des Projets de Developpement en Informatique a Partir de la Technique des Points de Fonction.* 1988.
57. Kemerer, C.F., *An empirical validation of software cost estimation models.* Communications of the ACM, 1987. **30**(5): p. 416-429.
58. Kitchenham, B., et al., *An empirical study of maintenance and development estimation accuracy.* Journal of Systems and Software, 2002. **64**(1): p. 57-77.
59. Maxwell, K., *Applied Statistics for Software Managers*. 2002: Prentice Hall PTR.
60. Miyazaki, Y., et al., *Robust regression for developing software estimation models.* Journal of Systems and Software, 1994. **27**(1): p. 3-16.
61. Wu, D., J. Li, and C. Bao, *Case-based reasoning with optimized weight derived by particle swarm optimization for software effort estimation.* Soft Computing, 2018. **22**(16): p. 5299-5310.
62. Liu, Q., J. Xiao, and H. Zhu, *Feature selection for software effort estimation with localized neighborhood mutual information.* Cluster Computing, 2019. **22**(3): p. 6953-6961.
34. Suresh Kumar, P., et al., *A pragmatic ensemble learning approach for effective software effort estimation.* Innovations in Systems and Software Engineering, 2022. **18**(2): p. 283-299.
35. Ahmed, M., et al., *Blockchain-Based Software Effort Estimation: An Empirical Study.* IEEE Access, 2022. **10**: p. 120412-120425.
36. Rankovic, N., et al., *A new approach to software effort estimation using different artificial neural network architectures and Taguchi orthogonal arrays.* IEEE Access, 2021. **9**: p. 26926-26936.
37. Linstone, H.A. and M. Turoff, *The Delphi method: techniques and applications*. 1975: Addison-Wesley Pub. Co., Advanced Book Program.
38. Barry, B., *Software engineering economics.* New York, 1981. **197**.
39. Norman, E.S., S.A. Brotherton, and R.T. Fried, *Work Breakdown Structures: The Foundation for Project Management Excellence*. 2010: Wiley.
40. Putnam, L.H., *A General Empirical Solution to the Macro Software Sizing and Estimating Problem.* IEEE Transactions on Software Engineering, 1978. **SE-4**(4): p. 345-361.
41. Nelson, E.A. *MANAGEMENT HANDBOOK FOR THE ESTIMATION OF COMPUTER PROGRAMMING COSTS*. 1967.
42. Wolverton, R.W., *The Cost of Developing Large-Scale Software.* IEEE Transactions on Computers, 1974. **C-23**(6): p. 615-636.
43. Aron, J.D. *Estimating resources for large programming systems*. 1969.
44. Park, R. *The Central Equations of the PRICE Software Cost Model*. 1988.
45. Walston, C.E. and C.P. Felix, *A method of programming measurement and estimation.* IBM Systems Journal, 1977. **16**(1): p. 54-73.
46. Black, R.K.E., et al. *BCS Software Production Data*. 1977.
47. Carriere, W. and R.J.G.R.C. Thibodeau, Rep. CR-3-839, *Development of a logistics software cost estimating technique for foreign military sales.* 1979.
48. Zadeh, L.A., *Fuzzy sets.* Information and Control, 1965. **8**(3): p. 338-353.
49. Breiman, L., *Classification and regression trees*. 2017: Routledge.
50. Pearl, J., et al., *Probabilistic reasoning in intelligent systems: Networks of plausible inference.* 1995.
51. Daniel, G.G., *Artificial Neural Network*, in *Encyclopedia of Sciences and Religions*, A.L.C. Runehov and L. Oviedo, Editors. 2013, Springer Netherlands: Dordrecht. p. 143-143.
52. Awad, M. and R. Khanna, *Support Vector Regression*, in *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, M. Awad and R. Khanna, Editors. 2015, Apress: Berkeley, CA. p. 67-80.
53. Aamodt, A. and E. Plaza, *Case-based reasoning: Foundational issues, methodological variations,*