مجله محاسبات نرم، جلد ۱۱، شماره ۲ (پاییز و زمستان ۱۴۰۱)، صفحه ۱۸ـ۳۷
https://doi.org/10.22052/scj.2023.246672.1085

مجله محاسبات نرم
SOFT COMPUTING JOURNAL
تارنمای مجله: scj.kashanu.ac.ir

دانشگاه کاشان
University of Kashan

ارائه راهکاری جهت بهبود استدلال مبتنی بر قالب در برآورد هزینه نرم‌افزار با استفاده از جستجوی ممنوعه✧

کوشیار اسماعیلی‌راد[۱]، کارشناسی ارشد، مهدی ملامطلبی[۲*]، استادیار

[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.

اطلاعات مقاله

تاریخچه مقاله:
دریافت ۰۴ مرداد ماه ۱۴۰۱
پذیرش ۲۵ دی ماه ۱۴۰۱

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

چکیده

ارائه برآوردی دقیق از هزینه مورد نیاز توسعه نرم‌افزار جهت مدیریت مؤثر و موفق پروژه‌های نرم‌افزاری، یک ضرورت محسوب می‌گردد. در مراحل آغازین توسعه نرم‌افزار، اطلاعات دقیقی از پروژه در دسترس نیست و همین امر، ارائه برآورد دقیق هزینه توسعه نرم‌افزار را چالش برانگیز می‌کند. این تحقیق با هدف افزایش دقت برآورد هزینه مورد نیاز توسعه نرم‌افزار، روشی بر پایه استدلال مبتنی بر قالب و جستجوی ممنوعه ارائه کرده که با حذف ویژگی‌های غیرضروری، برآورد با حداقل ویژگی‌های مورد نیاز را برای پروژه‌های نرم‌افزاری، ممکن نموده است. مکانیزم انتخاب راه حل 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 ارائه شده نسبت به روش اعتبارسنجی متقابل جامع و غیرجامع در مجموعه داده‌های آلبرخت، چاینا، دِشارنایز، کِمِرر، کیچنهام، مَکسول و میازاکی، اندازه‌گیری شده 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، میانه خطای نسبی، میانگین خطای نسبی و میانگین خطای مطلق، نشان داده است. علاوه بر این، کارایی روش پیشنهادی، از متدلوژی و محیط استفاده شده جهت توسعه نرم‌افزار، مستقل است و با تمرکز بر بخش دانش عمومی استدلال مبتنی بر قالب، پایداری دقت برآوردهای ارائه شده در مجموعه داده‌های تحقیق را فراهم آورده است.

© ۱۴۰۱ نویسندگان. مقاله با دسترسی آزاد تحت مجوز CC-BY

۱. مقدمه

علی‌رغم تلاش‌های فراوان در راستای بهبود توسعه نرم‌افزار، میزان شکست در پروژه‌های نرم‌افزاری همچنان بالا است. آمارها بیانگر آن است که هزینه تحمیل‌شده از این گونه شکست‌ها در اقتصاد ایالات متحده آمریکا در بازه 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، مرحله جمع‌آوری نیازمندی‌ها، را

✧ نوع مقاله: پژوهشی
* نویسنده مسئول
پست(های) الکترونیک: esmaeeli@buiniau.ac.ir (اسماعیلی‌راد)
motalebi@gmail.com (ملامطلبی)

نحوه ارجاع به مقاله: اسماعیلی‌راد، کوشیار، ملامطلبی، مهدی، «ارائه راهکاری جهت بهبود استدلال مبتنی بر قالب در برآورد هزینه نرم‌افزار با استفاده از جستجوی ممنوعه»، مجله محاسبات نرم، جلد ۱۱، شماره ۲، ص۱۸ـ۳۷، پاییز و زمستان ۱۴۰۱.

به خود اختصاص دهد [۲]؛ اما در مقابل، دارای فواید بسیاری است که می‌تواند موفقیت پروژه را تضمین کند. درک هزینه نرم‌افزار بسیار مهم است چرا که برآورد هزینه روی بسیاری از امور ضروری فرآیند توسعه 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 حوزه تحقیقاتی در توسعه نرم‌افزار است.

به عنوان یک تعریف، روند محاسبه طرح، سختی کار، تلاش، اندازه راهکارهای نرم‌افزاری و مجموعه هزینه‌های مرتبط با توسعه برنامه‌های نرم‌افزاری را برآورد هزینه نرم‌افزار گویند [۴]. تعریف ارائه شده از برآورد هزینه، دلالت بر مدیریت منابع دارد که از اصلی‌ترین مساله‌های روند تولید است. با شناخت کافی از منابع و توانایی‌ها، می‌توان توسعه پروژه را شتاب داده و ضمن کاهش هزینه‌ها، کیفیت نرم‌افزار را نیز بهبود بخشید [۵]. به طور کلی، مدل‌های هزینه مهندسی نرم‌افزار و تکنیک‌های برآورد برای چندین منظور شامل بودجه‌بندی، سبک/سنگین کردن و تحلیل خطر، برنامه‌ریزی و کنترل پروژه و تحلیل سرمایه‌گذاری بهبود نرم‌افزار مورد استفاده قرار می‌گیرند [۶].

از دید شرکت‌های توسعه‌دهنده نرم‌افزار، توان پردازشی و نیروی انسانی از اصلی‌ترین منابع مورد نیاز توسعه به شمار می‌روند. با توجه به اینکه بر اساس آمار موجود، توان پردازشی در هر ۱۸ ماه، به طور تقریبی رشد دو برابری دارد، نیروی انسانی یک هزینه قابل توجه پیش روی شرکت‌ها در بودجه توسعه نرم‌افزار است و برنامه‌ریزی صحیح تلاش آنها برای شرکت‌ها جنبه کلیدی دارد [۷]. به همین روی، بیشتر روش‌های برآورد هزینه نرم‌افزار، روی این جنبه تمرکز دارند و بر همین اساس، به طور معمول برآوردهای انجام شده هزینه تولید نرم‌افزار، با مبنای نفر–ماه ارائه می‌شوند [۸].

عوامل شناخته شده و شناخته نشده زیادی روی برآورد هزینه نرم‌افزار مؤثر هستند. به عنوان مثال، دقیق و هدفمند نبودن نیازمندی‌های پروژه، تازگی پروژه یا فناوری‌های مورد نیاز آن، تلاش برای تطابق برآورد انجام شده با بودجه و زمان در نظر گرفته شده، تغییرات غیرعملی و سنگین در طرح‌ریزی پروژه در زمان اجرا، زبان برنامه‌نویسی، توانایی گروه و مرحله‌ای که در حال اجرا است، از عوامل مؤثر در برآورد هزینه نرم‌افزار هستند. یکی دیگر از عواملی که در فرآیند برآورد هزینه نرم‌افزار مهم و تأثیرگذار است، اندازه پروژه است ضمن آنکه، دقت در محاسبه آن، تأثیر مستقیم بر روی برآورد انجام شده خواهد داشت [۵]، [۹]، [۱۰].

در دسترس نبودن اطلاعات کافی از پروژه در مراحل ابتدایی، تخمین اندازه 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 اندازه پروژه در نظر گرفته می‌شوند. هیچ یک از این معیارها، در ابتدای پروژه به طور دقیق در دسترس نیستند که این امر نیز به دشواری‌های ارائه برآورد دقیق می‌افزاید [۱۱]. از طرفی، نیازمندی‌های پروژه که نقشی تعیین‌کننده در اندازه نرم‌افزار دارند، به طور مکرر در فرآیند توسعه و برآورد نرم‌افزار دچار تغییر می‌شوند. همین امر، بر ضرورت انجام برآورد در سراسر چرخه حیات و توسعه نرم‌افزار دلالت دارد [۱۰]، [۱۱]. از اینرو، تغییرات مکرر در نیازمندی‌های پروژه هم به پیچیدگی‌های موجود در این حوزه‌ی تحقیقاتی می‌افزاید.

در توسعه پروژه‌های مبتنی بر وب، برآورد هزینه، حیاتی و چالش‌برانگیز است [۷]. این چالش ناشی از تفاوت‌های موجود در ویژگی‌های توسعه نرم‌افزارهای مبتنی بر وب، با توسعه سایر پروژه‌های نرم‌افزاری است. تفاوت موجود بین توسعه 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 برآوردکنندگان این حوزه را دچار چالش می‌کند، عبارت‌اند از:

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

۱) هدف اصلی توسعه نرم‌افزارهای مبتنی بر وب، ارائه سریع برنامه‌های باکیفیت، در کوتاه‌ترین زمان ممکن است. در مقابل، توسعه نرم‌افزارهای سنتی، در پی ارائه نرم‌افزار باکیفیت با کمترین هزینه هستند.

۲) اندازه پروژه‌های مبتنی بر وب به طور معمول کوچک و اندازه گروه توسعه، بین ۳ تا ۵ نفر است. در مقابل، توسعه سنتی شامل پروژه‌هایی با اندازه متوسط و بزرگ بوده و گروه توسعه نیز به طور معمول متشکل از صدها نفر است.

۳) زمان‌بندی در توسعه نرم‌افزارهای مبتنی بر وب، بین ۳ تا ۶ ماه است که این زمان در توسعه نرم‌افزارهای سنتی، ۱۰ تا ۱۸ ماهه است.

۴) فرآیندهای توسعه به کار گرفته شده در توسعه نرم‌افزارهای مبتنی بر وب، با فرآیندهای توسعه نرم‌افزارهای سنتی متفاوت است.

۵) فناوری‌های مهندسی استفاده شده در نرم‌افزارهای مبتنی بر وب، از نرم‌افزارهای سنتی متفاوت است.

۶) محصولات توسعه‌یافته مبتنی بر وب، سامانه‌های مبتنی بر شیء هستند و قابلیت استفاده مجدد بالایی دارند. همچنین رابط خارجی آنها محدود و به نسبت ساده است. در مقابل، محصول نهایی در توسعه نرم‌افزارهای سنتی، سامانه‌های مبتنی بر کد و جدیدی است که رابط‌های خارجی زیاد دارد و اغلب نرم‌افزارهای پیچیده‌ای هستند، اما قابلیت استفاده مجدد کمتر در آنها دیده می‌شود.

۷) افرادی که در توسعه 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 توسعه نرم‌افزارهای سنتی، مهندسین زبده نرم‌افزار، با بیش از ۵ سال سابقه در حداقل دو حوزه‌ی برنامه‌نویسی هستند.

۸) تکنیک‌های برآورد مورد استفاده در توسعه نرم‌افزارهای مبتنی بر وب، قیاس مبتنی بر تجربه جاری است و طراحی، متناسب با منابع در دسترس انجام می‌شود. اما در توسعه نرم‌افزارهای سنتی، قیاس مبتنی بر داده‌های ثبت شده 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 عبارت‌اند از:

۱) افزایش شاخص ارزیابی سطح پیش‌بینی (PRED)[1]،

۲) کاهش شاخص ارزیابی میانگین خطای نسبی (MMRE)[2] و میانگین خطای مطلق (MAE)[3] و

۳) افزایش دقت برآورد هزینه نرم‌افزار که در نتیجه اهداف فوق حاصل خواهد شد.

همچنین محدوده تحقیق عبارت است از:

۱) در این تحقیق روش استدلال مبتنی بر قالب برای برآورد هزینه توسعه نرم‌افزار انتخاب شده 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 هزینه نرم‌افزار با مطالعه پیشینه تحقیق معرفی و مدل‌های ارائه شده بر پایه 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 به تفصیل تشریح می‌گردد. بخش چهارم نتایج به دست آمده از روش پیشنهادی را ارائه و ارزیابی می‌کند و در نهایت، 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 پروژه جدید در 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

[1] Prediction level
[2] Mean of Magnitude of Relative Error
[3] Mean Absolute Error

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 پروژه 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 عنوان مثال، با افزایش تعداد صفحات ایستایی که در توسعه 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 مساله‌های برآورد هزینه 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 داده کوکومو و شاخص‌های ارزیابی MMRE و PRED در روش‌های محاسبات نرم است. همچنین، بر اساس تحقیق کومار و بِهِرا، شبکه عصبی مصنوعی در مقایسه با منطق فازی و محاسبات تکاملی، بیشتر در حل مساله‌های برآورد هزینه نرم‌افزار، مورد استفاده قرار گرفته‌اند.

اِذَر[6] و همکارانش [۹]، تکنیک‌های استفاده شده در حوزه برآورد تلاش نرم‌افزارهای مبتنی بر وب را بررسی کرده‌اند. نتایج تحقیق آنها بیانگر آن است که تکنیک‌های استدلال قیاس محور، رگرسیون خطی[7] و رگرسیون گام‌به‌گام[8]، به ترتیب با ۳۴/۵، ۳۴/۵ 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 انجام شده روی مجموعه داده‌ها، نشان می‌دهد که پروژه‌های صنعتی، در ۶۹ درصد از مقالات، مورد استفاده واقع شده‌اند.

ایده یافتن بهترین تکنیک برآورد هزینه نرم‌افزار، توسط کادودا[9] و همکارانش [۱۴] رد شده است؛ چرا که موفقیت یک تکنیک خاص، وابسته به اندازه مجموعه 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 مقایسه مدل‌ها و تکنیک‌های برآورد هزینه نرم‌افزار، تلاش کرده‌اند مناسب‌ترین تکنیک‌ها و مدل‌ها را در مجموعه داده‌ها و محیط‌های توسعه 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 هزینه نرم‌افزارهای تلفن‌های هوشمند، ارزیابی کرده‌اند. ارزیابی با استفاده مجموعه داده 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 ذکر شده

[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)

را تنها در یک مجموعه داده ارزیابی کرده‌اند که میزان اطمینان به نتایج حاصل شده را کاهش می‌دهد و از معایب تحقیق آنها به شمار می‌رود.

دی‌یو[1] و همکارانش [۱۶]، مدلی هوشمند و ترکیبی بر پایه روش شبکه عصبی و منطق فازی معرفی کردند. این مدل با مدل الگوریتمی سی‌یِرسیم[2] [۱۷]، ترکیب شده و سیستم استنتاج سازگار فازی-عصبی (ANFIS)[3] [۱۸]، به عنوان معماری هر یک از زیرمدل‌های فازی-عصبی مورد استفاده قرار گرفته 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.

اَلگابرای[4] و همکارانش [۱۹]، ضرایب مدل کوکومو[5] ارائه شده 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[6] و جاسازی‌شده[7])، به صورت جداگانه، بررسی شده است. نتایج حاصل، بیانگر افزایش دقت برآوردهای ارائه شده 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 ارائه برآورد نیست.

در همین راستا، سینگال[8] و همکارانش [۲۱]، با استفاده از الگوریتم تکاملی تفاضلی[9] به ارائه 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 داده 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[10] [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 پایه UCP، دقیق‌تر عمل کرده است. نتایج به دست آمده، تنها در یک مجموعه داده کوچک ۲۴ پروژه‌ای ارزیابی شده است. بنابراین، جهت اطمینان از بهبودهای حاصل‌شده، عدم وابستگی مدل ارائه شده به مجموعه داده، نیازمند بررسی بیشتر است.

کورازا[11] 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 وکا [۲۶] و الگوریتم جستجوی گرید [۲۷]، ارزیابی شده است. مدل ارائه شده، 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، بیانگر برتری جستجوی ممنوعه، در تنظیم پارامترهای مدل رگرسیون بردار پشتیبان بوده‌اند.

عبدالعلی[12] و همکارانش [۲۸]، مدلی مبتنی بر الگوریتم جنگل

[1] Du et al.
[2] SEER-SEM
[3] Adaptive Neuro-Fuzzy Inference System
[4] Algabri et al.
[5] Constructive Cost Model (COCOMO)
[6] Semi-detached
[7] Embedded
[8] Singal
[9] Differential Evolution
[10] Winters
[11] Corazza et al.
[12] Abdelali

تصادفی[1] ارائه نموده‌اند. کارایی این مدل پس از بهینه‌سازی تجربی پارامترهایش با مدل پایه 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 پایه درخت رگرسیون نشان داده است. با این حال، کارایی مدل عبدالعلی و همکارانش تنها با مدل پایه درخت رگرسیون سنجیده شده است و جهت اطمینان بیشتر، مقایسه با روش‌های غیرپایه، ضروری به نظر می‌رسد.

تکنیک‌های سنتی برآورد تلاش نرم‌افزار، در برآورد پروژه‌هایی که با استفاده از متدلوژی توسعه نرم‌افزار چابک[2] توسعه داده می‌شوند، از دقت کافی برخوردار نیستند. در راستای حل این مشکل، ضیاءالدین[3] و ضیاء[4] [۲۹]، مدلی متناسب با متدلوژی‌های توسعه تکرارشونده ارائه کردند. همچنین، پریاسَمی[5] و گاود[6] [۳۰]، با تمرکز روی جزئیات داخلی هر مورد کاربرد، به توسعه مدل UCP پرداختند. الویدین[7] و هادی[8] [۳۱]، ۱۶ عامل محیطی جدید را برای برآورد تلاش نرم‌افزار، مورد استفاده قرار داده اند. ژا[9] و مالو[10] [۳۲]، به تشریح ۲۴ عامل محیطی مرتبط با محیط پیاده‌سازی و بررسی اثرات این عوامل در برآورد هزینه نرم‌افزار، با استفاده از مدل کارنر [۲۳] پرداختند.

ژیا[11] و همکارانش [۳۳]، یک روش پیکربندی تحت نام ROME[12] ارائه دادند که از بهینه‌سازی مدل ترتیبی (SMO) جهت یافتن بهترین پیکربندی برآورد تلاش نرم‌افزار با توجه به مجموعه داده انتخاب شده، استفاده نموده است و نتیجه تحقیق آنها 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[13] و همکارانش [۳۴]، مدلی مبتنی بر رگرسیون تقویت گرادیان با استفاده از یادگیری گروهی[14] با هدف بهبود دقت برآورد تلاش نرم‌افزار و با بکارگیری 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[15] و همکارانش [۳۵]، یک روش جهت بهبود ارزیابی تلاش نرم‌افزار مبتنی بر بلاکچین[16] و وب ارائه نمودند. همچنین آنها روش پیشنهادی را بر اساس MMRE[17]، MAE[18] و PRED مورد سنجش قرار داده 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 داده و نتیجه گرفتند که روش مبتنی بر بلاکچین می‌تواند در غلبه بر فقدان موارد فوق، به آنها یاری رساند.

رانکوویچ[19] 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. به علاوه، آنها از مجموعه داده‌های کوکومو ۸۱، کوکومو ۲۰۰۰ و کمرر جهت مقایسه‌ها و انجام آزمایشات، بهره بردند.

نتیجه بررسی انواع مدل‌های برآورد تلاش، نشان‌دهنده برتری مدل‌های مبتنی بر یادگیری ماشین، نسبت به مدل‌های الگوریتمی و مدل‌های مبتنی بر نظریه کارشناس است. نیازمندی مدل‌های

[1] Random forest
[2] Agile software development
[3] Ziauddin
[4] Zia
[5] Periyasamy
[6] Ghode
[7] Alwidian
[8] Hadi
[9] Jha
[10] Malu
[11] Xia
[12] Rapid Optimizing Methods for Estimation
[13] Kumar
[14] Ensemble learning
[15] Mansoor Ahmed
[16] Blockchain
[17] Mean Magnitude of Relative Error
[18] Mean Absolute Error
[19] Rankovic

الگوریتمی به داشتن اطلاعات کامل از پروژه و مشخص نبودن فرآیند برآورد در مدل‌های مبتنی بر نظریه کارشناس، نشان‌دهنده عدم کارایی مدل‌های الگوریتمی و عدم تکرارپذیری مدل‌های مبتنی بر نظریه کارشناس است. در مقابل، مدل‌های مبتنی بر یادگیری ماشین، تکرارپذیر هستند و برای ارائه برآورد، به اطلاعات کامل پروژه هدف نیاز ندارند. به علاوه، مدل‌های مبتنی بر یادگیری ماشین دارای رویکرد افزایشی هستند و با استفاده از تجربه کارشناس و اطلاعات پروژه‌های پیشین، مزیت دو مدل الگوریتمی و نظریه کارشناس را به همراه دارند. در جدول (۱)، جمع‌بندی و مقایسه این سه دسته مدل برآورد تلاش نرم‌افزار ارائه شده است.

۳. روش پیشنهادی

در آزمایشات مربوط به ارزیابی روش پیشنهادی، برد اصلی رایانه میزبان، مدل Z270P-D3-CF برند گیگابایت با پردازنده چهار هسته‌ای Corei5 سری ۷۴۰۰ شرکت اینتل با فرکانس ۳/۰ گیگاهرتز و حافظه اصلی ۸ گیگابایتی است. سیستم عامل نصب شده روی سخت‌افزار رایانه میزبان، نسخه 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.

جدول (۱): جمع‌بندی و مقایسه مدل‌های ارزیابی تلاش نرم‌افزار

مدل	ویژگیها	مزایا	معایب	نمونه
مبتنی بر نظریه کارشناس	بهره‌گیری از دانش برآوردکننده (کارشناس) و تجربه به دست آمده وی در پروژه‌های مشابه پیشین، اساس برآورد هزینه نرم‌افزار در مدل‌های مبتنی بر نظریه کارشناس است	انجام برآورد توسط نیروی انسانی و به کمک هوش طبیعی باعث می‌شود در مواقعی که داده‌ها غیرقابل اندازه‌گیری هستند هم قادر به ارائه برآورد باشد.	قاعده خاصی را دنبال نمی‌کند و بنابراین تکرار پذیر نیست.	دِلفی[1] [۳۷]، دِلفی پهن‌باند[2] [۳۸] و ساختار شکست کار[3] [۳۹]
الگوریتمی	اساس کار استفاده از معادلات ریاضی، جهت برآورد هزینه نرم‌افزار بوده و اصلی‌ترین محرک هزینه در آنها، اندازه نرم‌افزار بر اساس تعداد خطوط برنامه، تعداد صفحات و تعداد پیوندها (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 پیش‌بینی نشده. در مواردی که عوامل تاثیرگذار در برآورد هزینه، قابل اندازه‌گیری نباشند، قادر به ارائه برآورد نیست. نیازمند داشتن اطلاعات کامل از پروژه است.	کوکومو [۲۰]، اِسلیم[4] [۴۰]، سی‌بِرسیم [۱۷]، اِس‌دی‌سی[5] [۴۱]، تی‌آردِبلیو وُلورتون[6] [۴۲]، داتی[7] [۴۳]، پرایس اِس[8] [۴۴]، آی‌بی‌اِم-اِف‌دی‌اِس[9] [۴۵]، بوئینگ[10] [۴۶]، جی‌آرسی[11] [۴۷] و مِتا[12]
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 ارائه برآورد، به اطلاعات کامل پروژه هدف نیاز ندارد.	به طور معمول مستقل استفاده نمی‌شود و به عنوان مکمل به همراه مدل‌های دیگر به 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

این تحقیق، به بهینه‌سازی استدلال مبتنی بر قالب با استفاده از الگوریتم جستجوی ممنوعه پرداخته است. الگوریتم جستجوی ممنوعه در پنج مرحله زیر خلاصه می‌شود. مرحله اول (انتخاب راه‌حل اولیه) راه‌حل اولیه، به صورت تصادفی یا طبق مکانیزم تعیین‌شده، از مجموعه راه‌حل‌های مساله انتخاب می‌شود. مرحله دوم (تشکیل مجموعه کاندید) مجموعه 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. مرحله سوم (یافتن راه‌حل برتر) راه‌حل‌های موجود در مجموعه کاندید، با استفاده از تابع هدف امتیازدهی می‌شوند. سپس، راه‌حلی که دارای بیشترین امتیاز است، به عنوان راه‌حل برتر، در نظر گرفته می‌شود. مرحله 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 این صورت، راه‌حل برتر از مجموعه کاندید حذف شده و فرآیند جستجو از مرحله سوم ادامه می‌یابد. مرحله پنجم (حرکت به راه‌حل برتر) حرکت به راه‌حل برتر، شامل به‌روزرسانی شاخص آرمانی، به‌روزرسانی لیست ممنوعه و انتخاب راه‌حل برتر به عنوان راه‌حل جاری است. پس از انجام این مرحله، جستجو از مرحله دوم تا رسیدن به شرط خاتمه ادامه می‌یابد.

پس از برقراری شرط خاتمه و توقف فرآیند جستجو، آخرین راه‌حل برتر پذیرفته شده در مرحله چهارم، به عنوان راه‌حل نهایی مساله ارائه می‌شود. کیفیت هر یک از اعضای مجموعه کاندید، در مرحله سوم جستجوی ممنوعه، مورد بررسی قرار می‌گیرد و متناسب با تابع هدف، امتیازی به اعضا اختصاص داده می‌شود. از بین اعضا مجموعه کاندید، عضوی که دارای بیشترین امتیاز است، به عنوان راه‌حل برتر انتخاب می‌شود. این تحقیق، در صدد بهبود دقت هزینه برآورد شده نرم‌افزار است و لذا، بیشترین امتیاز به عضوی اختصاص می‌یابد که کمترین میزان خطا را بر اساس شاخص‌های ارزیابی کسب کند. رابطه (۱)، نشان‌دهنده تابع هدف و چگونگی محاسبه امتیاز اعضای مجموعه کاندید است.

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

در این رابطه امتیاز عضو x از مجموعه کاندید محاسبه می‌شود که در آن n اندازه مجموعه داده مورد آزمایش و e_i خطای پیش‌بینی مورد i ام 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 خاتمه جستجو و اندازه 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، فاصله اقلیدسی بدون وزن، شاخص اندازه‌گیری شباهت است که بر اساس آن، ۳ مورد مشابه برای پروژه هدف انتخاب می‌گردد. پارامتر تطبیق تشابه استدلال مبتنی بر قالب در تحقیق حاضر، میانگین رتبه‌بندی شده معکوس بوده و قوانین تطبیق نیز، پارامتر خنثی شده استدلال مبتنی بر قالب است.

مقدار پارامترهای ثابت جستجوی ممنوعه، بر اساس تعداد ویژگی‌های مجموعه داده مورد آزمایش، تعیین می‌شوند. با

فرض n به عنوان تعداد ویژگی‌ها، پارامتر اندازه لیست ممنوعه برابر $2 \times \sqrt{2^n}$ و پارامتر شرط خاتمه 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، زیرمجموعه تهی راه‌حل‌های مساله، 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، در مجموع ۸ سناریوی متفاوت در آزمایش‌های این تحقیق وجود دارد که مقدار پارامترهای متغیر آنها، در جدول (۲) معرفی شده است.

جدول ۲. پارامترهای متغیر سناریوی آزمایش‌های تحقیق

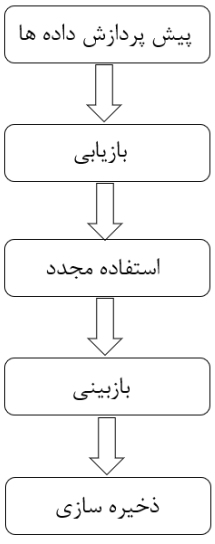
شناسه سناریو	روش انتخاب راه‌حل اولیه	شاخص ارزیابی	حذف داده‌های گمشده
۱	مجموعه ویژگی‌ها	MAE	بله
۲	مجموعه ویژگی‌ها	MMRE	بله
۳	مجموعه ویژگی‌ها	MAE	خیر
۴	مجموعه ویژگی‌ها	MMRE	خیر
۵	مجموعه تهی	MAE	بله
۶	مجموعه تهی	MMRE	بله
۷	مجموعه تهی	MAE	خیر
۸	مجموعه تهی	MMRE	خیر

در جدول (۲)، تنها مقدار پارامترهای متغیر ۸ سناریوی آزمایش‌ها قید شده است. در ضمن 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 داده گروه پیش‌بینی تلاش پرامیس، اعمال شده است. آلبرخت [۵۴]، چاینا [۵۵]، دِشارنایز [۵۶]، کِمِرر [۵۷]، کیچنهام [۵۸]، مَکسول [۵۹] و میازاکی [۶۰]، هفت مجموعه داده این تحقیق هستند که خلاصه‌ای از ویژگی‌های آنها در جدول (۳) شرح داده شده است.

جدول ۳. خلاصه‌ای از ویژگی‌های مجموعه داده 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
کِمِرر	۱۵	۷	۲	۰	نفر-ماه	۲۳/۲	۱۱۰۷/۳۱	۲۱۹/۲۴
کیچنهام	۱۴۵	۱۰	۳	۱۳	نفر-ساعت	۲۱۹	۱۱۳۹۳۰	۳۱۱۳/۱۱
مَکسول	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، استفاده مجدد و ذخیره‌سازی استدلال مبتنی بر قالب به‌همراه گام پیش‌پردازش داده‌ها، مراحل تشکیل‌دهنده روش پیشنهادی این تحقیق هستند. نوآوری تحقیق جاری، ارائه پیکربندی جدید استدلال مبتنی بر قالب و استفاده از جستجوی ممنوعه در مرحله بازیابی آن است که در این بخش تشریح شده است.



شکل (۱): مراحل روش پیشنهادی

[1] NA (Not Available)

۳،۱. 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 مرحله پیش‌پردازش داده‌ها حذف می‌شوند. پس از حذف ویژگی‌های خنثی و داده‌های گمشده، مقیاس‌گذاری روی ویژگی‌های عددی مجموعه داده اعمال می‌شود. قاعده مقیاس‌گذاری این تحقیق، استانداردسازی مقادیر ویژگی‌های عددی در بازه 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 این رابطه $S(x_i)$ مقدار استانداردسازی شده ویژگی x، در i امین پروژه مجموعه داده است. همچنین x_i نشان‌دهنده مقدار ویژگی x، در i امین پروژه، پیش از استانداردسازی است و $min(x_1, \dots, x_n)$ و $max(x_1, \dots, x_n)$ به ترتیب مقدار کمینه و بیشینه ویژگی x در مجموعه داده n عضوی را نشان می‌دهند.

۳،۲. 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 اندازه لیست ممنوعه و شرط خاتمه جستجو، به همراه پارامتر متغیر مکانیزم انتخاب راه‌حل اولیه، سه پارامتر استفاده شده در گام دوم جهت پیکربندی جستجوی ممنوعه هستند. افزون بر شش پارامتر ذکر شده، پارامتر متغیر شاخص ارزیابی، دیگر پارامتر تعیین‌کننده گام دوم تحقیق است که در تابع هدف مورد استفاده قرار می‌گیرد.

گام دوم روش برآورد هزینه داده گذر، به شرح چگونگی بکارگیری جستجوی ممنوعه در استدلال مبتنی بر قالب اختصاص دارد. وظیفه 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 مجموعه آرمانی ارائه شده 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 مجموعه تهی و مجموعه کامل ویژگی‌ها است، آغاز می‌شود. اگر مجموعه داده مورد آزمایش دارای ۳ ویژگی باشد، در حالت مجموعه تهی و مجموعه کامل ویژگی‌ها، به ترتیب جستجو از ۰۰۰ و ۱۱۱ شروع می‌شود. جستجوی ممنوعه پس از انتخاب یکی از حالت‌های راه‌حل اولیه، مجموعه کاندید را تولید می‌کند. همان‌طور که در شبه کد ارائه شده در الگوریتم (۱) نشان داده شده است، مجموعه کاندید از همسایه‌های راه‌حل اولیه تشکیل می‌شود. متغیرهای S، TL، AL، I و WOI در خطوط ۱ تا ۳، مقداردهی شده‌اند که به ترتیب نشان‌دهنده راه‌حل اولیه، لیست ممنوعه، شاخص آرمانی، تعداد تکرارهای جستجوی ممنوعه و تعداد تکرارهای بدون بهبود شاخص آرمانی هستند.

حلقه تعریف شده در خط چهارم تا رسیدن به شرط خاتمه ادامه می‌یابد و در هر دور تکرار، خط هشتم، مجاز بودن حرکت به راه‌حل برتر ذخیره شده در متغیر B را ارزیابی می‌کند. راه‌حل برتر از مجموعه 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 خاتمه حلقه، اجرای n تکرار بدون بهبود شاخص آرمانی است.

الگوریتم (۱): گام دوم روش پیشنهادی

```
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
```

در نهایت، زیرمجموعه آرمانی ذخیره شده در متغیر topS توسط جستجوی ممنوعه ارائه شده و به عنوان اولین ورودی، در تابع similarProjects مورد استفاده قرار می‌گیرد. در واقع، تابع similarProjects، انتخاب ویژگی را از مجموعه داده مورد آزمایش، مطابق متغیر topS انجام داده و متناسب با مشخصات پروژه 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 پروژه هدف هستند. بردار ویژگی موارد مشابه، در متغیر SP ذخیره می‌شود که خروجی گام دوم روش برآورد هزینه داده گذر است.

۳،۳. استفاده مجدد

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

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

این رابطه 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 پروژه هدف هستند که بر اساس میزان تشابه به صورت نزولی مرتب‌سازی شده‌اند. ضرایب اختصاص داده شده به هر یک از پروژه‌های مشابه، نشان‌دهنده میزان اثرگذاری آنها در تلاش برآورد شده پروژه هدف است. میزان تاثیرگذاری اولین، دومین و سومین پروژه مشابه، روی تلاش پروژه هدف، به ترتیب ۵۰٪، ۳۳٪ و ۱۷٪ است. حاصل رابطه (۳)، تلاش برآورد شده برای پروژه هدف است که با E_{target} نشان داده شده است.

۳،۴. بازبینی

راه‌حل ارائه شده در گام سوم، یک راه‌حل ارزیابی نشده است که صحت آن در گام چهارم بررسی می‌شود. گام چهارم شامل دو بخش است. در اولین بخش، راه‌حل ارائه شده گام سوم ارزیابی شده و در صورت درستی، به گام پنجم ارائه می‌شود. ارزیابی توسط خود مدل، به کمک کارشناس و یا با بکارگیری در محیط واقعی انجام می‌گیرد که بسته به نوع مساله، تعیین می‌شود. علاوه بر نوع ارزیابی، مدت زمان مورد نیاز جهت ارزیابی هم به نوع مساله وابسته است.

در صورت عدم درستی راه‌حل ارائه شده 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 ارائه شده با استفاده از توضیحات خطا ویرایش می‌شود. با توجه به مشخص شدن

تلاش واقعی پروژه پس از تکمیل آن، راه‌حل ویرایش‌شده قطعی است و نیاز به بازبینی مجدد ندارد.

۳،۵. ذخیره‌سازی

گام پنجم، به ذخیره‌سازی اطلاعات مفید راه‌حل بازبینی شده، در موارد پیشین می‌پردازد. این گام شامل انتخاب اطلاعات مفید پروژه جدید و چگونگی ذخیره‌سازی آن در ساختار حافظه است. اطلاعات پروژه به عنوان مساله و هزینه مورد نیاز توسعه آن به عنوان راه‌حل، اطلاعات مفید پروژه جدید هستند. با این حال، اطلاعات راه‌حل‌های اشتباه، جهت تفکیک راه‌حل‌های درست و غلط و محاسبه مجموع خطاها مفید هستند. به علاوه، سیستم هنگام مواجهه با یک خطا می‌تواند از خطاهای مشابه پیشین، جهت اصلاح راه‌حل جدید استفاده کند. مساله‌ها و راه‌حل‌های ارائه شده پیشین، بخشی از دانش عمومی سیستم استدلال مبتنی بر قالب هستند. اگر مجموعه داده مورد آزمایش دارای n پروژه باشد، اطلاعات بازبینی شده پروژه جدید، به عنوان $n+1$ امین پروژه، در بخش دانش عمومی ذخیره می‌شود. دانش به دست آمده از مساله جدید، جهت بهبود راه‌حل‌های ارائه شده سیستم در مساله‌های آینده 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 مقاله مبنای تحقیق، پرداخته می‌شود. سیستم استدلال مبتنی بر قالب که روش برآورد هزینه داده گذر، مبتنی بر آن است، با بهره‌گیری از دانش عمومی خود به حل مساله‌های جدید بر اساس مساله‌های مشابه پیشین می‌پردازد. تجربیات موفق و ناموفق این سیستم در ارائه راه‌حل مساله‌های جدید، در بخش دانش عمومی ذخیره شده و در مساله‌های 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 مساله 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 آزمایش می‌شود.

جهت ارزیابی روش برآورد هزینه داده‌گذر، بایستی قدرت پیش‌بینی آن اندازه‌گیری شود. قدرت پیش‌بینی، بیانگر دقت این روش در برآورد هزینه پروژه هدف است که با فاصله هزینه پیش‌بینی شده و هزینه واقعی، رابطه معکوس دارد. بنابراین، هرچه فاصله هزینه پیش‌بینی شده و هزینه واقعی بیشتر شود، از قدرت پیش‌بینی کاسته خواهد شد. در این تحقیق، شاخص‌های اندازه‌گیری قدرت پیش‌بینی، میانگین خطای نسبی، میانگین خطای مطلق، میانه خطای نسبی و سطح پیش‌بینی هستند.

جهت اندازه‌گیری دقت پیش‌بینی، زیرمجموعه‌ای به نام مجموعه آموزشی از مجموعه داده مورد آزمایش، انتخاب می‌شود. سپس، روش برآورد هزینه داده گذر با استفاده از مجموعه آموزشی متناسب‌سازی شده و زیرمجموعه دیگری به نام مجموعه ارزیابی از مجموعه داده، انتخاب می‌شود. از آنجا که این روش با مجموعه 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 ارائه شده 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 زیرمجموعه ارزیابی یک پروژه‌ای، از مجموعه داده n پروژه‌ای را بررسی می‌کند. در هر یک از حالت‌ها، یک پروژه به عنوان مجموعه ارزیابی و $n-1$ پروژه باقیمانده به عنوان مجموعه 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 داده

[1] Cross-validation
[2] Leave-One-Out cross validation

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[1] انجام شده است. این روش غیرجامع است و مجموعه داده را به ۱۰ زیرمجموعه ارزیابی با تعداد پروژه‌های یکسان، تقسیم می‌کند. زیرمجموعه‌های ارزیابی به صورت تصادفی انتخاب می‌شوند. هر زیرمجموعه ارزیابی در روش 10-Fold، دارای یک زیرمجموعه آموزشی متناظر است. اگر مجموعه داده مورد آزمایش، مجموعه مرجع باشد، متمم هر یک از زیرمجموعه‌های ارزیابی، زیرمجموعه آموزشی متناظر آن است. دقت پیش‌بینی، به ازای هر زیرمجموعه ارزیابی محاسبه شده و میانگین آن به عنوان قدرت پیش‌بینی روش برآورد هزینه داده گذر در مجموعه داده 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 بسته آماده در نرم‌افزار R، پیاده‌سازی شده است. بسته‌های فیلنترُپی[2]، دی‌پلیر[3]، اُپن‌ایکس‌ال‌اِس‌ایکس[4] و اِف‌ای‌آر‌اِف‌اِف[5]، چهار بسته آماده‌ای هستند که جهت پیاده‌سازی روش برآورد هزینه داده گذر، مورد استفاده قرار گرفته‌اند. بسته فیلنترُپی، دربرگیرنده تابع euclidean است که در گام دوم این روش جهت محاسبه فاصله بردار پروژه هدف با بردار پروژه‌های پیشین، استفاده شده است. مجموعه داده مورد آزمایش، پس از بازگشایی توسط بسته اِف‌ای‌آر‌اِف‌اِف و ورود به نرم‌افزار R، در قالب یک چهارچوب داده[6]، نگهداری می‌شود. بسته دی‌پلیر نرم‌افزار R، متمرکز بر کار با چهارچوب‌های داده است. دی‌پلیر در گام دوم روش برآورد هزینه داده گذر وظیفه انتخاب ویژگی از مجموعه داده مورد آزمایش را بر اساس راه‌حل ارائه شده توسط جستجوی ممنوعه، عهده‌دار است. همچنین، بسته 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 تعبیه شده در نرم‌افزار R، استفاده شده است.

مجموعه داده 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 زیرمجموعه کاندید است. جدول (۴)، قدرت پیش‌بینی روش پیشنهادی تحقیق در مجموعه داده آلبرخت را به ازای هر یک از سناریوها نشان می‌دهد.

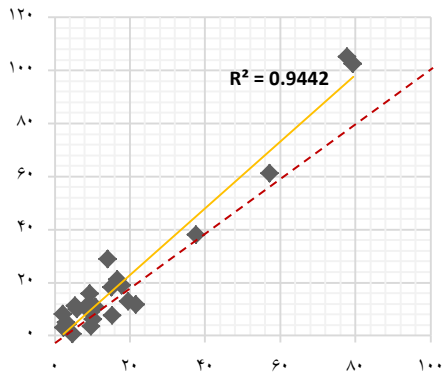
جدول (۴): نتایج مجموعه داده آلبرخت

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

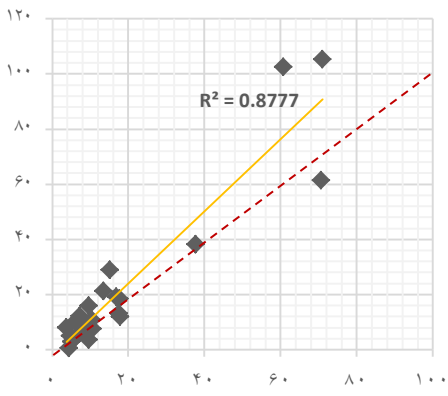
هر یک از سطرهای جدول (۴) نشان‌دهنده قدرت پیش‌بینی سناریوهای تحقیق در مجموعه داده آلبرخت بر اساس شاخص‌های ارزیابی MMRE، MdMRE، MAE و PRED(۰/۲۵) هستند. دو قدرت پیش‌بینی متفاوت از اجرای سناریوها روی مجموعه داده آلبرخت به دست آمده است. قدرت پیش‌بینی اول مربوط به سناریوهای ۱، ۳، ۵ و ۷ (گروه آ) و قدرت پیش‌بینی 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 داده 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

[1] 10-Fold cross validation
[2] Philentropy
[3] Dplyr
[4] Openxlsx
[5] Farff
[6] Data frame

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 میانه خطای نسبی گروه (آ) نسب به گروه (ب)، سناریوهای ۱، ۳، ۵ و ۷ نتایج دقیق‌تری ارائه کرده‌اند. رابطه تلاش واقعی پروژه‌های مجموعه داده آلبرخت با تلاش برآورد شده توسط سناریوهای گروه (آ) و گروه (ب) به ترتیب در شکل‌های (۳) و (۴) نشان داده شده است. همچنین، در شکل‌های (۵) و (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



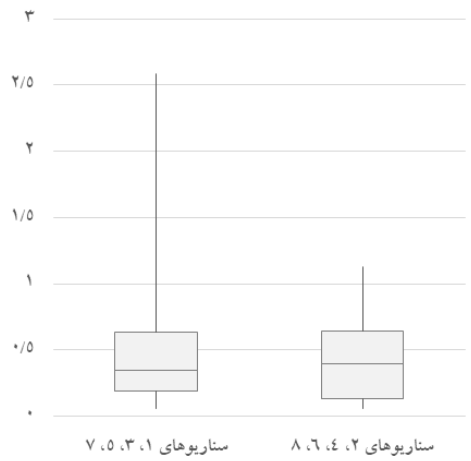
شکل (۴): نمودار نقطه‌ای سناریوهای ۲، ۴، ۶ و ۸ مجموعه داده آلبرخت

در شکل‌های (۳) و (۴)، محور x به تلاش‌های برآورد شده 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 رگرسیون و حداقل بودن زاویه تشکیل شده 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 پیش‌بینی شده را نشان می‌دهد. زاویه تشکیل شده بین خط روند و خط رگرسیون شکل (۳)، نسبت به شکل (۴) کوچکتر است. در ضمن نقاط شکل (۳) در مقایسه با شکل (۴) به خط رگرسیون نزدیکتر هستند و مقدار شاخص R^2 در شکل (۳)، 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)، 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 نشان‌دهنده خطای نسبی است، نمودار سمت چپ به سناریوهای گروه (آ) و نمودار سمت راست به سناریوهای گروه (ب) اختصاص دارد. مشاهدات شکل (۵)، 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 بیشینه 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.

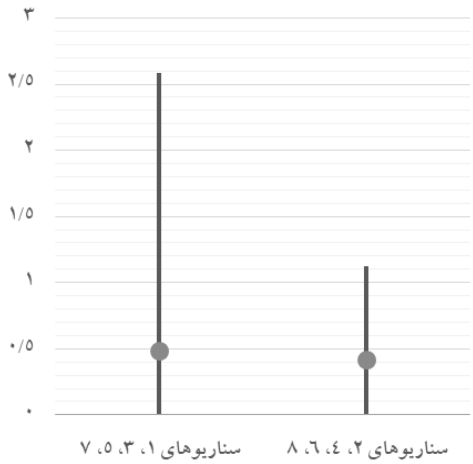
در شکل (۶)، تنوع نتایج به دست آمده از دو گروه (آ) و (ب) با استفاده از نمودار توزیع فاصله نشان داده شده است که در آن، نمودار سمت چپ به گروه (آ) و نمودار سمت راست به گروه (ب)، تعلق دارد. محور y این نمودار نشان‌دهنده خطای نسبی است و بلندی طول آن تنوع بیشتر نتایج را نشان می‌دهد. همچنین، نقاط درج شده روی نمودار، نشان‌دهنده میانگین

[1] Coefficient of determination

خطای نسبی گروه مربوطه است. مشاهدات این شکل 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.



شکل (۵): نمودار جعبه‌ای خطای نسبی مجموعه داده آلبرخت



شکل (۶): نمودار توزیع فاصله خطای نسبی مجموعه داده آلبرخت

با استناد مشاهدات جدول (۴) و بررسی نقطه به نقطه نتایج دو گروه در شکل‌های (۵) و (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 و MAE، قابل اطمینان نیستند. بنابراین در مجموع، سناریوهای گروه (آ) که متشکل از سناریوهای شماره ۱، ۳، ۵ و ۷ هستند، دقیق‌ترین نتایج را با استفاده از روش برآورد هزینه داده گذر در مجموعه داده 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، نتیجه 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 پرداخته 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 فاصله اقلیدسی، فاصله منهتن و درجه رابطه خاکستری استفاده می‌کنند. همچنین، هر یک از سه مدل، دارای دو حالت وزن‌دهی شده و بدون وزن هستند. روش برآورد هزینه داده گذر با مدل‌های وزن‌دهی شده وو که دارای برآوردهای دقیق‌تری هستند، ارزیابی شده است. جداول (۵)، (۶) و (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 داده	مدل WEuc-CBR وو			روش پیشنهادی تحقیق		
	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 جداول (۵) تا (۷)، نتایج مدل‌های وو و نتیجه برترین سناریوی روش برآورد هزینه داده گذر در مجموعه داده‌های دِشارنایز و مَکسول، نشان داده شده است. نتایج این جداول بر اساس شاخص‌های MMRE، MdMRE و PRED(۰/۲۵) می‌باشند. نتایج روش برآورد هزینه داده گذر این جداول در مجموعه داده دِشارنایز، متعلق به سناریوهای ۱، ۲، ۵ و ۶ و در مجموعه داده مَکسول، متعلق به سناریوهای ۵ و ۷ است.

دقیق‌ترین مدل وو در مجموعه داده 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 داده دِشارنایز هستند. نتایج حاکی از برتری قاطع روش برآورد هزینه داده گذر بر اساس تمامی شاخص‌ها، نسبت به ۳ مدل وو در مجموعه داده دِشارنایز است. برترین نتایج مدل‌های وو در مجموعه داده دِشارنایز با زیر خط، نشانه‌گذاری شده‌اند.

در مجموعه داده 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، براساس 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 داده 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 داده مَکسول با زیر خط در جداول مذکور نشانه‌گذاری شده‌اند. جدول (۸) 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) را نمایش می‌دهد. سطر اول این جدول به مجموعه داده دِشارنایز و سطر دوم آن به مجموعه داده مَکسول اختصاص دارد. میزان اختلاف و درصد بهبود در تمامی شاخص‌ها و مجموعه داده‌ها، مثبت است. مثبت بودن میزان اختلاف و درصد بهبودی، نشان‌دهنده برتری روش برآورد هزینه داده گذر، نسبت به مدل‌های ارائه شده توسط وو و همکارانش است.

جدول (۸): مقایسه 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

لیو و همکارانش [۶۲]، با هدف بهبود استدلال مبتنی بر قالب، به طراحی یک روش حریصانه انتخاب ویژگی بر اساس واحد اندازه‌گیری اطلاعات متقابل همسایگی بومی شده، پرداخته‌اند. مدل لیو، روی مجموعه داده‌های آلبرخت، چاینا، دِشارنایز، کِمِرر، کیچنهام و میازاکی آزمایش شده است. جدول (۹)، نتایج مدل لیو و نتیجه برترین سناریوی روش برآورد هزینه داده گذر در مجموعه داده‌های آلبرخت، چاینا، دِشارنایز، کِمِرر، کیچنهام و میازاکی نشان داده شده است. نتایج این جدول بر اساس شاخص‌های ارزیابی MMRE و MAE می‌باشد. نتایج روش برآورد هزینه داده گذر در مجموعه داده آلبرخت متعلق به سناریوهای ۱، ۳، ۵ و ۷ و در مجموعه داده چاینا متعلق به سناریوهای ۲، ۴، ۶ و ۸ است. همچنین، نتایج این روش در مجموعه داده دِشارنایز متعلق به سناریوهای ۱، ۲، ۵ و ۶ و در مجموعه داده کیچنهام متعلق به سناریوهای ۳ و ۷ می‌باشد. نتایج سناریوهای ۱ تا ۸ نیز در مجموعه داده‌های کِمِرر و میازاکی یکسان است که از آنها به عنوان نتایج روش پیشنهادی جدول (۹) استفاده شده 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
کِمِرر	۰/۵۳	۱۰۹	۰/۴۳۴۰	۱۱۲/۴۵۷۶
کیچنهام	۰/۲۸	۱۳۰۵	۰/۲۶۴۴	۶۲۹/۶۹۲۸
میازاکی	۰/۳۴	۴۹/۶	۰/۲۷۳۴	۱۵/۴۳۶۲

شاخص‌های MMRE و MAE جدول (۹) 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 داده کِمِرر، شاخص MAE برتری مدل لیو و شاخص MMRE برتری روش برآورد هزینه داده گذر را نشان می‌دهند. جدول (۱۰) به مقایسه نتایج روش برآورد هزینه داده گذر و نتایج مدل لیو در مجموعه داده‌های آلبرخت، چاینا، دِشارنایز، کِمِرر، 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. همان‌طور که در این جدول نشان داده شده است، میزان اختلاف و درصد بهبود تمامی شاخص‌ها و مجموعه داده‌ها، به جز شاخص MAE مجموعه داده کِمِرر، 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 کِمِرر، قابل چشم‌پوشی است. همچنین، این روش نسبت به مدل لیو در مجموعه داده کِمِرر، بر اساس شاخص تعیین‌کننده 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.

جدول (۱۰): مقایسه 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
کِمِرر	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

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 داده کِمِرر، لیو از روش برآورد هزینه داده گذر به میزان ۳/۱۷٪ بهتر عمل کرده است، ولی شاخص MMRE در مجموعه داده کِمِرر، 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 داده کِمِرر، قابل چشم‌پوشی است. به علاوه، در سایر شاخص‌های ارزیابی و مجموعه داده‌ها، روش برآورد هزینه داده گذر نسبت به مدل لیو نیز برتری دارد.

۵. نتیجه‌گیری

این تحقیق با هدف افزایش دقت برآورد هزینه نرم‌افزار و بهبود شاخص‌های ارزیابی، روشی بر پایه استدلال مبتنی بر قالب ارائه کرده است. اساس کار این تحقیق، پیکربندی پارامترهای شش‌گانه 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 مشاهده شده در این تحقیق با استفاده از شاخص‌های میانگین و میانه خطای نسبی، میانگین خطای مطلق و سطح پیش‌بینی، اندازه‌گیری شده است. در این تحقیق، بر اساس شاخص‌های ارزیابی MMRE، MdMRE، MAE و PRED(۰/۲۵)، به‌ترتیب ۲۲/۴۲٪، ۲۲/۶۳٪، ۲۵/۷۳٪ و ۲۸/۶۵٪ اثرگذاری مشاهده شده است. همچنین بهره‌گیری از جستجوی ممنوعه بر افزایش دقت برآورد هزینه نرم‌افزار ارائه شده 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)، به‌ترتیب 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 به‌ترتیب

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، کِمِرر و میازاکی، بی‌تاثیر بودن این پارامتر را نشان داده است. بنابراین، پارامتر شاخص ارزیابی در استدلال مبتنی بر قالب، وابسته به مجموعه داده مورد آزمایش می‌باشد و دارای اثرگذاری نسبی است. در عین حال، بکارگیری مکانیزم راه‌حل اولیه متفاوت در جستجوی ممنوعه، در بهبود دقت برآورد هزینه نرم‌افزار ارائه شده توسط روش استدلال مبتنی بر قالب مؤثر است. پارامتر متغیر مکانیزم انتخاب راه‌حل اولیه در سناریوهای تحقیق، تعبیه شده است. مشاهدات، تاثیرگذاری این پارامتر را در نتایج مجموعه داده مَکسول نشان می‌دهد. وجه تمایز مجموعه داده مَکسول در مقایسه با دیگر مجموعه داده‌ها، تعداد بالای زیرمجموعه‌های کاندید آن است. از آنجا که این پارامتر بر سایر مجموعه داده‌های تحقیق بی‌تاثیر بوده است، بکارگیری مکانیزم انتخاب راه‌حل اولیه 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 پروژه 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 بهینه موارد مشابه متناسب با مجموعه داده‌های مورد آزمایش در استدلال مبتنی بر قالب، ارائه راهکاری بهینه جهت تطبیق تشابه راه‌حل‌های بازیابی‌شده با مساله هدف در استدلال مبتنی بر قالب، جهت تحقیقات آینده پیشنهاد می‌گردند.

تعارض منافع: نویسندگان اعلام می‌کنند که هیچ تعارض منافعی ندارند.

مراجع

[1] A.L.I. Oliveira, "Estimation of software project effort with support vector regression," Neurocomputing, vol. 69, no. 13-15, pp. 1749-1753, 2006, doi: 10.1016/j.neucom.2005.12.119.

[2] M. Saroha and S. Sahu, "Tools & methods for software effort estimation using use case points model - A review," in International Conference on Computing, Communication & Automation, Greater Noida, India, 2015, pp. 874-879, doi: 10.1109/CCAA.2015.7148498.

[3] D. Basten and W. Mellis, "A Current Assessment of Software Development Effort Estimation," in 2011 International Symposium on Empirical Software Engineering and Measurement, Banff, AB, Canada, 2011, pp. 235-244, doi: 10.1109/ESEM.2011.32.

[4] W. Swati and K. Kishor, "A Novel Way of Cost Estimation in Software Project Development Based on Clustering Techniques," Int. J. Innov. Res. Comput. Commun. Eng., vol. 2, no. 4, pp. 3892-3899, 2014.

[5] S.A. Abbas, X. Liao, A.U. Rehman, A. Azam, and M.I. Abdullah, "Cost Estimation: A Survey of Well-known Historic Cost Estimation Techniques," J. Emerging Trends Comput. Inf. Sci., vol. 3, no. 4, pp. 612-636, 2012.

[6] B. Boehm, C. Abts, and S. Chulani, "Software development cost estimation approaches - A survey," Ann. Softw. Eng., vol. 10, no. 1, pp. 177-205, 2000, doi: 10.1023/A:1018991717352.

[7] K. Dejaeger, W. Verbeke, D. Martens, and B. Baesens, "Data Mining Techniques for Software Effort Estimation: A Comparative Study," IEEE Trans. Software Eng., vol. 38, no. 2, pp. 375-397, 2012, doi: 10.1109/TSE.2011.55.

[8] J. Borade and V.R. Khalkar, "Software Project Effort and Cost Estimation Techniques," Int. J.Adv. Res. Comput. Sci. Softw. Eng., vol. 3, no. 8, pp. 730-739, 2013.

[9] D. Azhar, 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, ACM: Lund, Sweden, 2012, pp. 49-58, 10.1145/2365324.2365332.

[10] V. Bardsiri and D. Jawawi, "Software Cost Estimation Methods: A Review," J. Emerg. Trends Comput. Inf. Sci., vol. 2, no. 1, pp. 21–29, 2010.

[11] E. Mendes, 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, Oxford, England, 2002, pp. 393-398, doi: 10.1109/CMPSAC.2002.1045034.

[12] D.J. Reifer, "Web development: estimating quick-to-market software," IEEE Softw., vol. 17, no. 6, pp. 57-64, 2000, doi: 10.1109/52.895169.

[13] P.S. Kumar and H. Behera, "Role of soft computing techniques in software effort estimation: an analytical study," in Computational Intelligence in Pattern Recognition. Advances in Intelligent Systems and Computing, vol 999, Springer, Singapore, doi: 10.1007/978-981-13-9042-5_70.

[14] G. Kadoda, M. Cartwright, and M. Shepperd, "Issues on the effective use of CBR technology for software project prediction," in Proceedings of 4th International Conference on Case-Based Reasoning, ICCBR, Vancouver, BC, Canada, 2001, pp. 276-290, doi: 10.1007/3-540-44593-5_20.

[15] M. Pandey, R. Litoriya, and P. Pandey, "Validation of Existing Software Effort Estimation Techniques in Context with Mobile Software Applications," Wirel. Pers. Commun., vol. 110, no. 4, pp. 1659-1677, 2020, doi: 10.1007/s11277-019-06805-0.

[16] W.L. Du, L.F. Capretz, A.B. Nassif, and D. Ho, "A hybrid intelligent model for software cost estimation," J. Comput. Sci., vol. 9, no. 11, pp. 1506-1513, 2013, doi: 10.3844/jcssp.2013.1506.1513.

[17] D.D. Galorath and M.W. Evans, Software sizing, estimation, and risk management: when performance is measured performance improves, 2006: Auerbach Publications.

[18] J.S. Jang, "ANFIS: adaptive-network-based fuzzy inference system," IEEE Trans. Syst. Man Cybern., vol. 23, no. 3, pp. 665-685, 1993, doi: 10.1109/21.256541.

[19] M. Algabri, F. Saeed, H. Mathkour, and N. Tagoug, "Optimization of soft cost estimation using genetic algorithm for NASA software projects," 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW), Riyadh, Saudi Arabia, 2015, pp. 1-4, doi: 10.1109/NSITNSW.2015.7176416.

[20] B.W. Boehm, B.K. Clark, E. Horowitz, J.C. Westland, R.J. Madachy, and R.W. Selby, "Cost models for future software life cycle processes: COCOMO 2.0," Ann. Softw. Eng., vol. 1, pp. 57-94, 1995, doi: 10.1007/BF02249046.

[21] P. Singal, A.C. Kumari, and P. Sharma, "Estimation of Software Development Effort: A Differential Evolution Approach," Proc. Comput. Sci., vol. 167, pp. 2643-2652, 2020, doi: 10.1016/j.procs.2020.03.343.

[22] A.B. Nassif, D. Ho, and L.F. Capretz, "Regression model for software effort estimation based on the use case point method," in International Conference on Computer and Software Modeling, IPCSIT, Singapore, 2011, pp.117-121.

[23] G. Karner, "Resource estimation for objectory projects," Obj. Syst. SF AB, vol. 17, pp. 1-9, 1993.

[24] G. Schneider and J.P. Winters, Applying Use Cases: A Practical Guide, 2nd edition, 2001: Addison-Wesley Professional.

[25] A. Corazza, S.D. Martino, F. Ferrucci, C. Gravino, F. Sarro, and E. Mendes, "Using tabu search to configure support vector regression for effort estimation," Empir. Softw. Eng., vol. 18, no. 3, pp. 506-546, 2013, doi: 10.1007/s10664-011-9187-3.

[26] M.A. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten, "The WEKA data mining software: an update," ACM SIGKDD Explor., vol. 11, no. 1, pp. 10-18, 2009, doi: 10.1145/1656274.1656278.

[27] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," ACM Trans. Intell. Syst. Technol., vol. 2, no. 3, pp. 1-27, 2011, doi: 10.1145/1961189.1961199.

[28] Z. Abdelali, H. Mustapha, and N. Abdelwahed, "Investigating the use of random forest in software effort estimation," Proc. Comput. Sci., vol. 148, pp. 343-352, 2019, doi: 10.1016/j.procs.2019.01.042.

[29] Z. Zia, T. Kamal and Z. Ziauddin, "An effort estimation model for agile software development," Adv. Comput. Sci. Appl., vol. 2, no. 1, pp. 314-324, 2012.

[30] K. Periyasamy and A. Ghode, "Cost estimation using extended use case point (e-UCP) model," in International Conference on Computational Intelligence and Software Engineering, Wuhan, China, 2009, pp. 1-5, doi: 10.1109/CISE.2009.5364515.

[31] J. Alwidian and W. Hadi, "Enhancing the results of ucp in cost estimation using new external environmental factors," in International Conference on Information Technology and e-Services, Sousse, Tunisia, 2012, pp. 1-11, doi: 10.1109/ICITeS.2012.6216623.

[32] P. Jha and R.K. Malu, "Cost estimation using parameterized for use case point model (P-UCP)," Int. J. Eng. Res. Appl., vol. 3, no. 5, pp. 1280-1283, 2013.

[33] T. Xia, R. Shu, X. Shen, and T. Menzies, "Sequential model optimization for software effort estimation," IEEE Trans. Software Eng., vol. 48, no. 6, pp. 1994-2009, 2022, doi: 10.1109/TSE.2020.3047072.

[34] P.S. Kumar, H.S. Behera, J. Nayak, and B. Naik, "A pragmatic ensemble learning approach for effective software effort estimation," Innov. Syst. Softw. Eng., vol. 18, no. 2, pp. 283-299, 2022, doi:

10.1007/s11334-020-00379-y.

[35] M. Ahmed, N. Iqbal, F. Hussain, M.-A. Khan, M. Helfert, Imran, and J. Kim, "Blockchain-Based Software Effort Estimation: An Empirical Study," IEEE Access, vol. 10, pp. 120412-120425, 2022, doi: 10.1109/ACCESS.2022.3216840.

[36] N. Rankovic, D. Rankovic, M. Ivanovic, and L. Lazic, "A new approach to software effort estimation using different artificial neural network architectures and Taguchi orthogonal arrays," IEEE Access, vol. 9, pp. 26926-26936, 2021, doi: 10.1109/ACCESS.2021.3057807.

[37] H.A. Linstone and M. Turoff, "The Delphi method: techniques and applications," J. Mark. Res., vol. 18, no. 3, 1975, doi: 10.2307/3150755.

[38] B.W. Barry, Software engineering economics, New York, 1981.

[39] E.S. Norman, S.A. Brotherton, and R.T. Fried, Work Breakdown Structures: The Foundation for Project Management Excellence, Wiley, 2010.

[40] L.H. Putnam, "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," IEEE Trans. Software Eng., vol. 4, no. 4, pp. 345-361, 1978, doi: 10.1109/TSE.1978.231521.

[41] E.A. Nelson, Management handbook for the estimation of computer programming costs, 1967.

[42] R.W. Wolverton, "The Cost of Developing Large-Scale Software," IEEE Trans. Computers, vol. 23, no. 6, pp. 615-636, 1974, doi: 10.1109/T-C.1974.224002.

[43] J.D. Aron, Estimating resources for large programming systems, FSC-69-5013, Federal Systems Center, IBM, Gaithersburg, Maryland, 1969, pp. 1-21.

[44] R. Park, "The central equations of the price software cost model," in 4th COCOMO Users Group Meeting, November 1988.

[45] C.E. Walston and C.P. Felix, "A method of programming measurement and estimation," IBM Syst. J., vol. 16, no. 1, pp. 54-73, 1977, doi: 10.1147/sj.161.0054.

[46] R.K.D. Black, R.P. Curnow, R. Katz, and M.D. Gray, "BCS Software Production Data," RADC-TR-77-116, Boeing Computer Services, 1977.

[47] W.M. Carriere and R. Thibodeau, "Development of a logistics software cost estimating technique for foreign military sales," Report CR-3-839, General Reserch Corporation, 1979.

[48] L.A. Zadeh, "Fuzzy sets," Inf. Control, vol. 8, no. 3, pp. 338-353, 1965.

[49] L. Breiman, Classification and regression trees, Routledge, 2017.

[50] J. Pearl, Probabilistic reasoning in intelligent systems: Networks of plausible inference, Morgan Kaufmann, 1988.

[51] G.G. Daniel, "Artificial Neural Network," in Encyclopedia of Sciences and Religions, A.L.C. Runehov, L. Oviedo, Eds., Springer: Dordrecht, The Netherlands, 2013, p. 143, doi: 10.1007/978-1-4020-8265-8_200980.

[52] M. Awad and R. Khanna, "Support Vector Regression," in Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers, Apress: Berkeley, CA pp. 67-80, 2015, doi: 10.1007/978-1-4302-5990-9_4.

[53] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," AI Commun., vol. 7, no. 1, pp. 39-59, 1994, doi: 10.3233/AIC-1994-7104.

[54] A.J. Albrecht and J.E. Gaffney, "Software function, source lines of code, and development effort prediction: a software science validation," IEEE Trans. Software Eng., vol. 9, no. 6, pp. 639-648, 1983, doi: 10.1109/TSE.1983.235271.

[55] T. Menzies, A. Butcher, D.R. Cok, A. Marcus, L. Layman, F. Shull, B. Turhan, and T. Zimmermann, "Local versus Global Lessons for Defect Prediction and Effort Estimation," IEEE Trans. Software Eng., vol. 39, no. 6, pp. 822-834, 2013, doi: 10.1109/TSE.2012.83.

[56] J.-M. Desharnais, "Statistical Analysis on the Productivity of Data Processing with Development Projects using the Function Point Technique," Master's Thesis, University of Quebec in Montreal, 1988.

[57] C.F. Kemerer, "An empirical validation of software cost estimation models," Commun. ACM, vol. 30, no. 5, pp. 416-429, 1987, doi: 10.1145/22899.22906.

[58] B.A. Kitchenham, S.L. Pfleeger, B. McColl, and S. Eagan, "An empirical study of maintenance and development estimation accuracy," J. Syst. Softw., vol. 64, no. 1, pp. 57-77, 2002, doi: 10.1016/S0164-1212(02)00021-3.

[59] K.D. Maxwell, Applied Statistics for Software Managers, Prentice Hall PTR, 2002.

[60] Y. Miyazaki, M. Terakado, K. Ozaki, and H. Nozaki, "Robust regression for developing software estimation models," J. Syst. Softw., vol. 27, no. 1, pp. 3-16, 1994, doi: 10.1016/0164-1212(94)90110-4.

[61] D. Wu, J. Li, and C. Bao, "Case-based reasoning with optimized weight derived by particle swarm optimization for software effort estimation," Soft Comput., vol. 22, no. 16, pp. 5299-5310, 2018, doi: 10.1007/s00500-017-2985-9.

[62] Q. Liu, J. Xiao, and H. Zhu, "Feature selection for software effort estimation with localized neighborhood mutual information," Clust. Comput., vol. 22, no. 3, pp. 6953-6961, 2019, doi: 10.1007/s10586-018-1884-x.