دوفصلنامه محاسبات و سامانه های توزیع شده
سال پنجم، شماره اول، شماره پیاپی ۹، صفحه ۵۶ تا ۶۳، سال ۱۴۰۱

ارائه راهکاری نوین جهت بهبود تخمین هزینه توسعه نرم افزار مبتنی بر تشابه

تقی جاودانی گندمانی*[۱]، مائده دشتی[۲]
[۱]گروه علوم کامپیوتر، دانشگاه شهرکرد، شهرکرد، ایران.
[۲]هسته پژوهشی علوم داده، گروه علوم کامپیوتر، دانشگاه شهرکرد، شهرکرد، ایران.

چکیده

یکی از مهمترین و بحرانی ترین عوامل در توسعه پروژه های نرم افزاری تخمین مناسب هزینه ها است. اهمیت تخمین چنان است که یکی از مهمترین عوامل شکست یا توفیق پروژه ها می باشد. دیدگاه ها و روش های متعددی در خصوص نحوه انجام تخمین ارائه شده است که یکی از مهمترین آنها دیدگاه مبتنی بر تشابه می باشد. بهره گیری از ویژگی های مناسب و وزن دهی ویژگی ها نمونه ای از تلاش برای بهبود دقت است. این مطالعه در جهت بهبود تخمین هزینه توسعه نرم افزار، میزان تاثیر الگوریتم LEM بر بهینه سازی وزن ویژگی ها را مورد بررسی قرار داده است و اقدام به ارائه راهکاری نوین در این خصوص نموده است. در این تحقیق میزان اثربخشی الگوریتم بر روی دو مجموعه داده Desharnais و Maxwell بررسی شده است و از معیارهای MMRE، (۰.۲۵) PRED و MdMRE برای ارزیابی و مقایسه روش پیشنهادی با دیگر الگوریتم های تکاملی از جمله ژنتیک، تکامل تفاضلی و ازدحام ذرات استفاده شده است. نتایج نشان می‌دهد این الگوریتم توانسته است بهبود قابل توجهی را به دست آورد.

کلمات کلیدی: تخمین هزینه نرم افزار، مدل تکاملی یادگیر، تخمین مبتنی بر تشابه، بهینه سازی وزن ویژگی ها.

Improving Analogy-based Software Development Cost Estimation Using Learnable Evolution Model

Taghi Javdani Gandomani*[1], Maedeh Dashti[2]
[1]Department of Computer Science, Shahrekord University, Shahrekord, Iran.
[2]Data Science Research Group, Shahrekord University, Shahrekord.

Abstract

one of the most important and critical factors in software projects is the proper cost estimation. due to the high significance and impact of the proper cost estimation, several approaches and methods have been proposed regarding how to perform cost estimation, in which the analogy-based approach is one of the foremost popular ones. To improve software development cost estimation, the current study has investigated the effect of the LEM algorithm on optimization of features weighting and proposed a new method as well. In this research, the effectiveness of this algorithm was inspected on two datasets, Desharnais and Maxwell. Then, MMRE, PRED (0.25), and MdMRE criteria have been used to evaluate and compare the proposed method against other evolutionary algorithms. Employing the proposed method showed considerable improvement in estimating software cost estimation.

Keywords: software cost estimation, Learnable evolution model (LEM) , Analogy-based estimation, Features weighting optimization.

تاریخچه مقاله:

تاریخ ارسال: ۱۴۰۰/۱۲/۲۹

تاریخ اصلاحات: ۱۴۰۱/۰۱/۱۰

تاریخ پذیرش: ۱۴۰۱/۰۴/۱۷

تاریخ انتشار: ۱۴۰۱/۰۵/۱۱

Keywords:

Software Cost Estimation, Learnable evolution model (LEM), Analogy-based estimation, Features weighting optimization

*ایمیل نویسنده مسئول:

javdani@sku.ac.ir

۱- مقدمه

فرآیند توسعه نرم افزار به مجموعه ای از فعالیتهای مهندسی نرم افزار اطلاق می شود که با هدف مدیریت چرخه حیات یک محصول نرم افزاری طراحی و برنامه ریزی می گردد. در فرآیند توسعه نرم افزار می بایست برای هر کدام از بخش ها و جزئیات آنها روش قواعد و اصول معینی را ارائه نمود که در این بین تخمین هزینه ها در ابتدای پروژه بخش بزرگی از برنامه‌ریزی را شامل می‌شود. اهمیت تخمین چنان است که به عنوان یکی از مهمترین عوامل شکست یا توفیق پروژه ها محسوب می شود. در دو دهه گذشته تلاش‌های قابل توجهی در جهت پیش بینی دقیق تر هزینه ها برای زمان بندی و برنامه ریزی موثر و مدیریت کیفیت محصول انجام شده است. اما به دلیل رشد روز افزون اندازه نرم افزارها و همچنین متغیر بودن هزینه ها در توسعه پروژه های نرم افزاری، دستیابی به تخمین دقیق هنوز هم دغدغه بسیاری از متخصصین باشد[۱].

در گذشته محققان بر مبنای اصول و فاکتورهای مختلف دسته بندی های متفاوتی را از روشهای موجود ارائه نموده اند. یورگنسن و شپرد[1] در طی یک بررسی سیستماتیک ۱۱ روش تحقیق را شناسایی نموده و آنها را به دو دسته مدل های پارامتری و غیر پارامتری تقسیم بندی نموده اند[۲]. در مدل های پارامتری تخمین بر اساس تجزیه و تحلیل آماری و یا عددی از مجموعه داده های تاریخی است. در مدلهای غیر پارامتری تخمین بر اساس اصول بهینه سازی و روش های هوش مصنوعی می باشد.

روش تخمین مبتنی بر تشابه یک مدل مبتنی بر مورد است که توسط استنبرگ[2] معرفی شد. و از برای بهبود تخمین هزینه ها در توسعه نرم افزار استفاده می شود. در سال های اخیر از روشهای یادگیری ماشین و محاسبات نرم برای بالا بردن دقت تخمین استفاده می شود [۳,۴]. این روش ها می توانند از طریق فرایند انتخاب پروژه یا وزن دهی به ویژگی ها مورد استفاده قرار گیرند[۵].

مدل تکاملی یاد گیر یکی از الگوریتم های نوظهور در هوش مصنوعی است. این الگوریتم یکی از الگوریتم های محاسبات تکاملی غیرداروینی است که از یادگیری ماشین برای هدایت فرآیند تکاملی استفاده می‌کند و می تواند از لحاظ تعداد مراحل تکامل سرعت ۲ تا چند برابری را به دست آورد[۶]. به دلیل آنکه که این الگوریتم در حل مسائل بهینه سازی پیچیده در دنیای واقعی توانسته است موفق عمل کند، در این مقاله نیز سعی می کنیم استفاده از این الگوریتم را در جهت بهبود تخمین هزینه نرم افزار به کار گیریم.

این مقاله به ۸ بخش تقسیم می شود. در بخش ۲ به معرفی اجمالی روش تخمین مبتنی بر تشابه پرداخته می شود و در ادامه در بخش ۳ کارهای مرتبط انجام شده در این حیطه مورد توجه قرار خواهد گرفت. در بخش ۴ الگوریتم مدل تکاملی یادگیر ارائه شده است. پس از آن در بخش ۵ درباره چارچوب پیشنهادی بحث خواهیم نمود و درباره معیارهای ارزیابی ، مجموعه داده ها و نحوه پیاده سازی آن در بخش ۶ صحبت خواهیم کرد. در بخش ۷ نتایج و آزمایشات تجربی با استفاده از مدل پیشنهادی ارائه خواهد شد. در نهایت در بخش ۸ نتیجه گیری ارائه خواهد شد.

۲- روش تخمین مبتنی بر تشابه

روش تخمین مبتنی بر تشابه بسیار ساده است. در این روش برای تخمین یک پروژه جدید باید آن را با پروژه های مشابهی که در گذشته انجام شده است که به آنها به مجموعه داده های تاریخی یا مخزن داده[3] می گویند مقایسه کنیم. فرایند تخمین مبتنی بر تشابه شامل گام‌های زیر است:

۱. به دست آوردن مجموعه داده تاریخی از طریق مجموعه داده های واقعی یا تولید داده های مصنوعی.
۲. به دست آوردن ویژگی های پروژه جدید به گونه ای که با مجموعه داده های تاریخی مطابقت داشته باشد
۳. استفاده از یک تابع شباهت تعریف شده مانند فاصله اقلیدسی و منهتن جهت بازیابی مشابه ترین پروژه ها.
۴. تخمین هزینه پروژه جدید با استفاده از تابع راه حل.

در زیر هر کدام از اجزای سیستم تخمین مبتنی بر تشابه به طور مجزا ارائه شده است.ت

۲-۱- تابع شباهت

تابع شباهت، هسته مرکزی روش تخمین مبتنی بر تشابه است و میزان تشابه بین دو پروژه مختلف را محاسبه می کند. فرم کلی تابع شباهت به شکل زیر می باشد:

$$sim(p.p') = \text{(۱)}$$
$$f\left(Lsim(f_1.f_1').Lsim(f_2.f_2').\ldots.Lsim(f_n.f_n')\right)$$

که در آن p و p' نشان دهنده دو پروژه جدید و پروژه قدیمی در مخزن داده است، f_i و f_i' نشان دهنده مقدار ویژگی i ام در پروژه ها می باشند، n نشان دهنده تعداد ویژگی‌ها در پروژه است

و تابع $Lsim(.)$ شباهت بین دو ویژگی متناظر از پروژه را محاسبه می نماید. توابع $Lsim(.)$ و $f(.)$ نشان دهنده ساختار عمومی تابع شباهت می باشند. شباهت دو پروژه با استفاده از فاصله اقلیدسی با استفاده از معادله (۲) به دست

[1] Jorgensen and Shepperd
[2] Sternberg
[3] Repository

می‌آید.

$$sim(p.p') = \frac{1}{\left[\sqrt{\sum_{i=1}^{n} w_i Dis(f_i.f_i')+\delta}\right]} \qquad \delta = 0.0001 \qquad (۲)$$

که در آن w_i وزن ویژگی iام است و مقدار آن بین ۰ و ۱ می باشد. همچنین $\delta = 0.0001$ یک عدد ثابت کوچک است که در فرمول قرار داده **می** شود تا مانع از تقسیم بر صفر شود.

فاصله منهتن یک نوع از فاصله اقلیدسی است با اندکی تغییر که در معادله (۳) نشان داده شده است.

$$sim(p.p') = \frac{1}{\left[\sum_{i=1}^{n} w_i Dis(f_i.f_i')+\delta\right]} \qquad \delta = 0.0001 \qquad (۳)$$

$$Dis(f_i.f_i') = \begin{cases} |f_i - f_i'| & if\ features\ are\ numeric \\ 1 & if\ features\ are\ numeric\ and\ f_i = f_i' \\ 0 & if\ features\ are\ numeric\ and\ f_i \neq f_i' \end{cases}$$

۱-۲-نزدیکترین همسایه (*KNN*):

الگوریتم *KNN* نمونه ای از یادگیری مبتنی بر نمونه است که در آن طبقه بندی برای یک رکوردی که هنوز طبقه بندی نشده است با مقایسه آن با شبیه ترین رکوردها در مجموعه داده آموزشی انجام میشود. *KNN* به عنوان یک پارامتر حیاتی بسیار موثر بر دقت شناخته می شود. در انتخاب مقدار k ، اگر k خیلی کوچک باشد تاثیر داده های پرت زیاد میشود و اگر مقدار k خیلی بزرگ باشد آنگاه رفتار محلی مورد نظر نادیده گرفته می شود. در اکثر مقالات مقدار k متغیر و بین ۱ تا ۵ می باشد [۷].

تابع پاسخ در این بخش مشخص می شود که چگونه پروژه های مشابه را با یکدیگر ترکیب نمود و هزینه پروژه جدید را تخمین زد. روش‌های ارزیابی مختلفی از جمله مشابه ترین پروژه[۴]، میانگین بدون وزن، میانه و میانگین وزن دار فاصله معکوس[۵]

به عنوان پایه ای برای تابع پاسخ در تحقیق محققان استفاده شده است.

میانگین به عنوان معیاری کلاسیک را زمانی می‌توان محاسبه نمود که در $k> 1$ باشد. میانه، نیز یک معیار کلاسیک دیگری است که زمانی می‌توان از ان استفاده کرد که در $k> 2$ باشد. میانه نسبت به میانگین معیار آماری قوی تری است زیرا میانه نسبت به داده های پرت حساس است و داده های پرت با افزایش تعداد پروژه ها افزایش می یابد. میانگین وزن دار فاصله معکوس به پروژه‌های مشابه اهمیت بیشتری نسبت به موارد کمتر مشابه دارند. فرمول میانگین وزن دار فاصله معکوس در معادله (۴) نشان داده شده است[۸].

$$\widehat{C_p} = \sum_{k=1}^{n} \frac{Sim(P.P_k)}{\sum_{i=1}^{n} Sim(P.P_k)} C_{pk} \qquad (۴)$$

که در آن P نشان دهنده پروژه ای است که باید هزینه آن تخمین زده شود. P_k نشان دهنده kامین پروژه مشابه است. $Sim(P.P_k)$ نشان دهنده شباهت بین پروژه های P و P_k است و C_{pk} نشان دهنده هزینه مشابه ترین پروژه به P_k است.

۳- کارهای مرتبط

تاکنون روش های مختلفی در راستای تخمین مبتنی بر تشابه ارائه شده است که تمرکز همگی آنها بر بهبود دقت تخمین بوده است. انتخاب ویژگی و وزن دهی ویژگی ها بیشترین روش‌هایی هستند که در تخمین مبتنی بر تشابه به آنها پرداخته شده است.

از روشهای کلاسیک[۹] و روش های یادگیری ماشین [۱۰, ۱۱] برای افزایش دقت تخمین نرم افزار به وفور استفاده شده است. ون و همکارانش در یک بررسی سیستماتیک ۸ مدل یادگیری ماشین را برای تخمین تلاش شناسایی کردند[۱۲]. در بیشتر تحقیقات انجام شده در این زمینه برای تابع شباهت از فاصله اقلیدسی، منهتن، مینکوفسکی و تعداد محدودی از مقالات از فاصله گری [۱۳] استفاده نموده اند.

در روش مبتنی بر تشابه اصلی هر کدام از ویژگی های پروژه مستقل در نظر گرفته می شوند و میزان تاثیر مشابهی دارند. *Auer* و همکارانش [۱۴] برای تخمین دقیق تر پیشنهاد دادند که ویژگی ها دارای تاثیر متفاوتی باشند. برای وزن دهی به ویژگی ها برخی از روشهای کلاسیک مانند رگرسیون[۱۵] و برخی از محاسبات نرم و الگوریتم های متاهیوریستسکی استفاده می نمایند[۱۶-۱۹]. در این بین برخی از محققان از سیستم های فازی[۲۰]، الگوریتم های تکاملی و شبکه های عصبی مصنوعی[۳, ۲۱] استفاده نموده اند.

۴- مدل تکاملی یادگیر (LEM)

تمام روش های رایج محاسبات تکاملی از اصول تئوری داروین الهام گرفته شده اند. محاسبات تکاملی داروین نیمه کورکورانه هستند. در آن برای تولید جمعیت جدید از پراتورهایی مانند جهش باز ترکیبی و انتخاب استفاده می شود. در این نوع تکامل افراد جمعیت جدید با استفاده از افراد آموزش دیده جمعیت های قبلی هدایت نمی شود بلکه یک فرآیند آزمون و خطا است که به صورت موازی انجام می شود. ایده اصلی مدل تکاملی یادگیر از ترکیب روشهای جستجوی تکاملی و یادگیری ماشین می باشد [۲۲]. این راهکار ، یک مدل یادگیری ماشین

[4]Closet analogy
[5]Inverse distance Weighted Mean

را اجرا میکند تا بتواند افرادی از جمعیت که در انجام وظایف از بقیه بهتر عمل می کنند را تشخیص دهد. این دلایل به عنوان فرضیه های استنتاجی بیان می شوند و سپس برای تولید جمعیت جدید این افراد مورد استفاده قرار میگیرند. در واقع جمعیت جدید را با استفاده از فرضیه های مربوط به افراد با تناسب بالا در جمعیت گذشته تولید می کند.

روند تکامل در *LEM* با تعدادی از اعضا از یک جمعیت اولیه آغاز می شود. در پردازش های تکاملی اعضا ممکن است راه حل های مسائل و یا دستورالعمل هایی تولید راه حل ها باشند در اینجا یک فرم کلی از فرآیند *LEM* بیان می شود:

۱. ایجاد جمعیت: جمعیت اولیه به صورت تصادفی یا بر اساس یک روش خاص ساخته می شود

۲. اجرای حالت یادگیری ماشین:

آ. استخراج اکسترما: از اعضای جمعیت حاضر دو گروه انتخاب می کنیم یک گروه با کارایی بالا که به طور خلاصه به آن *H-group* می گوییم در گروه دیگر با کارایی پایین را *L-group* می نامیم. مقدار این گروه ها با استفاده از مقدار تابع تناسب به دست می آید.

ب. ایجاد فرضیه ها: یک روش یادگیری ماشین برای توصیف *H-group* و *L-group* اعمال می‌کنیم که بتواند این دو گروه را از یکدیگر متمایز کند. هنگام یادگیری توصیف *H-group* (یا *L-group*) باید به تاریخچه تکامل یعنی جمعیت گذشته و یا توصیف جمعیت های گذشته نیز توجه کرد.

پ. ایجاد یک جمعیت جدید: برای تولید یک جمعیت جدید نمونه های یاد گرفته شده در*H-group* با جمعیت جدید ترکیب می‌شود. توصیف نمونه ها یا به صورت تصادفی یا به صورت قانون هایی از نمونه های توصیف شده انجام میشود.

ت. برو به مرحله *a* و حالت یادگیری ماشین را تکرار کن تا زمانی که به شرط خاتمه برسد: هنگامی که شرط خاتمه در حالت یادگیری ماشین برآورده شد می‌توان یکی از اقدامات زیر را انجام داد:

۱. فرایند تکامل را خاتمه داد

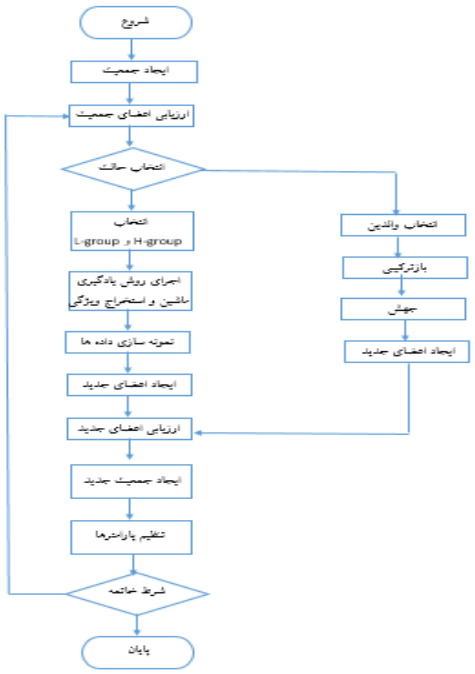
۲. تکرار فرایند از مرحله ۱ که به این عملیات شروع مجدد از اول گفته میشود.

۳. رفتن به مرحله ۳

۳. اجرای حالت تکامل داروینی: در این قسمت یکی از روشهای تکامل داروینی را اجرا می‌کنیم به این معنا که یک جهش با ترکیبی و انتخاب را برای تولید جمعیت جدید اعمال می‌کنیم این عملیات را تا زمانی انجام می‌دهیم که حالت تکامل داروین به شرط خاتمه برسد.

۴. تبادل کردن حالتها : در این حالت به مرحله دو می رویم سپس انتقال بین مراحل ۲ و ۳ را آنقدر انجام می‌دهیم که شرط خاتمه الی ان برآورده شود. بهترین فرد به دست آمده رسیدگی تکامل است.

شرط خاتمه در حالت یادگیری ماشین در روز در صورت رسیدن به یک سطح از کارایی به دست می‌آید اگر در این مرحله شرط خاتمه*LEM* هنوز برآورده نشده باشد، عملیات شروع مجدد را اجرا میکند در شکل۱ فلوچارت الگوریتم *LEM* نشان داده شده است.



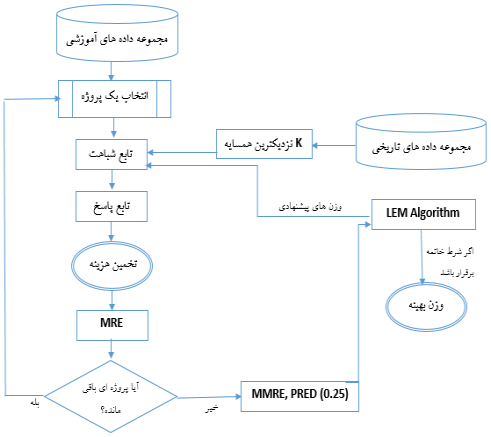
(شکل-۱): فلوچارت الگوریتم*LEM*

۵- مدل پیشنهادی

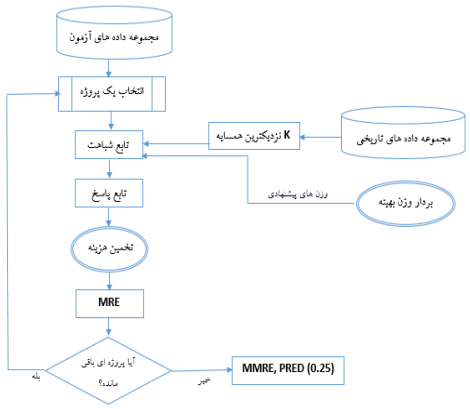
در این تحقیق سعی شده است که چارچوبی مبتنی بر *LEM* در جهت ارائه تخمین مناسبتر در پروژه های نرم افزاری ارائه گردد. چارچوب پیشنهادی از دو فاز آموزش و آزمون تشکیل شده است.

۵-۱- فاز آموزش

در این فاز مجموعه ای از داده های آموزشی به مدل ارائه می شود و سیستم تخمین هزینه مبتنی بر تشابه توسط وزن ویژگی ها تنظیم می شود و الگوریتم *LEM* برای به حداقل رساندن خطاها به بررسی بردار وزن در نمونه های آموزشی می‌پردازد. در شکل ۲ معماری این فاز نشان داده شده است.



(شکل-۲): معماری سیستم داده های آموزشی



(شکل-۳): معماری سیستم داده های آزمون

۶- راه اندازی آزمایش

۶-۱- معیار های ارزیابی

از آنجا که پژوهش‌های مختلف، توابع متفاوتی را برای آزمایش روش خود مورد استفاده قرار می دهند، انتخاب معیارهای ارزیابی مناسب جهت مقایسه با روش‌های دیگر کاری مشکل است. زیرا در مقالات متفاوت معیارهای ارزیابی به کار گرفته شده، مختلف هستند و در اکثریت موارد سورس کد برنامه در دسترس نیست. به همین دلیل در این تحقیق سعی شده است تا از عمومی ترین معیارهای ارزیابی که در اکثریت قریب به اتفاق مقالات مورد استفاده قرار گرفته شده است استفاده شود. شاخص عملکردی که معمولاً برای اندازه گیری کارایی مدل‌های پیش‌بینی نرم افزار استفاده می شود اندازه خطای نسبی[۶] (*MRE*) است که درصد خطای مطلق را نسبت به تلاش واقعی محاسبه می‌کند و در معادله ۵ نشان داده شده است. میانگین *MRE* ها را *MMRE* می نامیم که در معادله ۶ نشان داده شده است. همچنین میانه *MRE* ها ، *MdMRE* نامیده می شود.

$$MRE_i = \frac{|\text{Estimated value} - \text{actual value}|}{\text{actual value}} \qquad (۵)$$

$$MMRE = \frac{\sum_{i=1}^{N} MRE_i}{N} \qquad (۶)$$

شاخص دیگری که وجود دارد شاخص عملکرد $PRED(X)$ است که درصد پیش بینی هایی که مقدار X را درست تشخیص میدهند، را نشان میدهد در معادله (۷) تعریف می‌شود.

$$PRED(x) = \sum_{i=1}^{N} D_i * \frac{100}{N} \qquad (۷)$$

$$D_i = \begin{cases} 1 & \text{if MMRE} < \frac{x}{100} \\ 0 & \text{otherwise} \end{cases}$$

when $x = 25$ the PRED metric is defind as $PRED(0.25)$

به دلیل اینکه اکثریت مقالات برای ارزیابی روش های پیشنهادی خود از این سه معیار استفاده نموده‌اند در این تحقیق نیز از این معیارها استفاده خواهیم نمود.

۶-۱- مجموعه داده ها

در این تحقیق برای ارزیابی روش پیشنهادی از ۲ مجموعه داده اندازه‌گیری نرم افزار استفاده خواهد شد که عبارتند از:

- *Desharnais*: شامل پروژه‌های نرم‌افزاری کانادایی است.
- *Maxwell*: شامل پروژه های نرم افزاری فنلاند است

۶-۲- راه اندازی آزمایش

برای طراحی آزمایش و ارزیابی روش پیشنهادی با استفاده از آزمایشات مختلف ابتدا باید عملیات پیش پردازش بر روی داده ها اجرا شود که برای این کار با استفاده از معادله *min-max* داده ها در بازه صفر تا یک نرمال خواهند شد. عملیات نرمال سازی به جهت حذف تاثیرات متفاوت ویژگی ها انجام می شود. پس از آن برای آموزش مدل پیشنهادی باید مجموعه داده های آموزشی را به دو مجموعه داده های آموزشی و داده های آزمون تقسیم بندی نمود که در این تحقیق، این کار را با استفاده از روش *10 - fold cross validation* انجام شد.

همانطور که قبلا بیان شد روش تخمین مبتنی بر تشابه دارای سه پارامتر کنترلی است که عبارتند از: تابع شباهت، k نزدیک‌ترین همسایه و تابع پاسخ. در طراحی این آزمایش، جهت تابع شباهت از فواصل اقلیدوسی و منهتن و برای به دست آوردن مقدار تخمین پیشنهادی از رایج ترین توابع پاسخ که میانه و میانگین می باشند، استفاده خواهد شد. همچنین تعداد نزدیک ترین پروژه به یکدیگر را به صورت متغیر و در بازه ۱ تا ۵ فرض خواهیم کرد.

[6] Magnitude of relative error

۷- نتایج علمی

۹-۱- آزمایش روش پیشنهادی

بر اساس چارچوب پیشنهادی، مناسب ترین مدلی که در هر دو مجموعه داده مناسب باشد و بتواند به خوبی پاسخگو باشد مربوط به فاصله اقلیدسی و سه نزدیکترین همسایه و همچنین تابع پاسخ میانه می باشد. همچنین کمترین مقدار MMRE و MdMRE مربوط به فاصله اقلیدسی با ۵ نزدیکترین همسایه و نیز تابع پاسخ میانه است ولی مقدارPRED آنها با این که مقادیر نسبتا مناسبی دارد اما بیشترین مقدار در میان مدل ها را شامل نمی شود و بر طبق نتایج به دست آمده بیشترین مقدارPRED مربوط به فاصله منهتن و سه نزدیک ترین همسایه و تابع پاسخ میانه می باشد.

۹-۲- مقایسه کارایی روش پیشنهادی با سایر روش ها

برای مقایسه چند روش یکی از مواردی که باید رعایت شود این است که معیار کیفیت توابع ارزیابی و بازه های انتخاب شده مجموعه داده ها یکسان باشد تا بتوان به درستی مقایسه را انجام داد. در این تحقیق میزان کارآیی پیشنهادی (New Model) با دیگر الگوریتم های تکاملی از جمله ژنتیک (GA)، تکامل تفاضلی (DE) و الگوریتم ازدحام ذرات (PSO) مقایسه می شود و می‌توان عملکرد آنها را با یکدیگر مقایسه نمود. زیرا این الگوریتم ها ساختاری مشابه با الگوریتم پیشنهادی را دارا هستند. نتایج این مقایسات که بر روی دیتاست های مختلف پیاده سازی شده، در جداول ۱ و ۲ نشان داده شده است.

(جدول-۱): مقایسه مدل پیشنهادی با الگوریتم های DE و PSO در مجموعه داده Desharnais

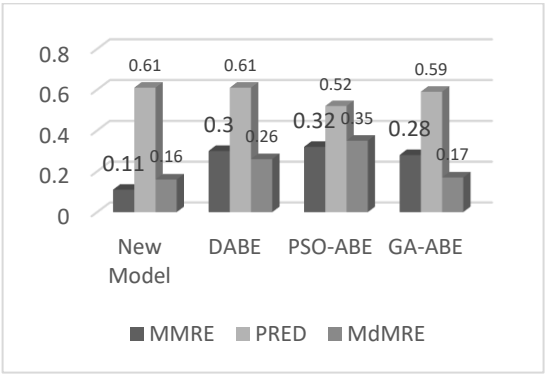
Model	Training Set			Testing Set		
	MMRE	PRED	MdMRE	MMRE	PRED	MdMRE
New Model	0.77	0.58	0.35	0.11	0.61	0.16
PSO-ABE	0.33	0.30	0.26	0.30	0.61	0.26
DABE	0.63	0.58	0.23	0.32	0.52	0.35
GA-ABE	0.60	0.58	0.30	0.28	0.59	0.17

(جدول-۲): مقایسه مدل پیشنهادی با الگوریتم های DE و PSO در مجموعه داده Maxwell

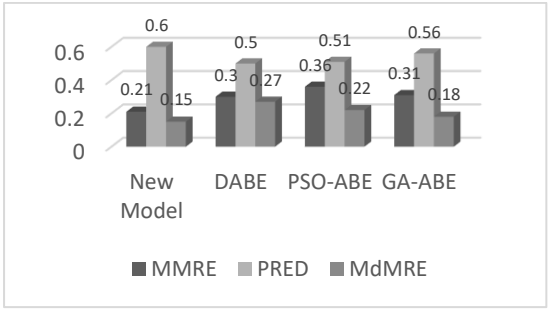
Model	Training Set			Testing Set		
	MMRE	PRED	MdMRE	MMRE	PRED	MdMRE
New Model	0.24	0.55	0.17	0.21	0.60	0.15
PSO-ABE	0.31	0.5	0.27	0.30	0.50	0.27
DABE	0.33	0.52	0.22	0.36	0.51	0.22
GA-ABE	0.33	0.55	0.24	0.31	0.56	0.18

همان‌طور که در جداول فوق نمایش داده شده است، مقدار معیار های *MMRE* و *MdMRE* کمتر از مقادیر مشابه آنها نسبت به دیگر الگوریتم ها می باشد. این مساله در هر دو مجموعه داده صدق می کند. این مساله به طور واضح تر در شکل های ۴ و ۵ نشان داده شده است.

همان طور که در این شکل ها نشان داده شده است روش پیشنهادی می تواند نتایج بهتری را تولید کند.



(شکل-۴): نمودار بررسی مقدار معیار های MMRE و MdMRE و PRED در مجموعه داده Maxwell



(شکل-۵): نمودار بررسی مقدار معیار های MMRE و MdMRE و PRED در مجموعه داده Desharnais

۸- نتیجه گیری

در این مقاله ما یک تکنیک بهینه سازی وزن بر اساس مدل تکاملی یادگیر در تخمین مبتنی بر تشابه پیشنهاد کردیم. مطالعات تجربی در این مقاله نشان می دهد که در بیشتر موارد نتایج حاصل از مدل پیشنهادی در معیارهای ارزیابی متفاوت رضایت بخش بوده است. در ادامه تحقیق چارچوب پیشنهادی با دیگر الگوریتم های تکاملی مورد مقایسه قرار گرفت که در این موارد نیز نتیجه مطلوب تری ارائه شده است. به دلیل اینکه مدل تکاملی یادگیر اعضای جمعیت جدید را توسط یک سری فرآیند تولید و ایجاد فرضیه می سازد، می تواند مشکل تصادفی بودن تولید جمعیت در الگوریتم های مبتنی بر مدل داروین را تا حدودی مرتفع سازد. این امر می تواند باعث جذب شدن بسیاری از محققین حوزه تخمین هزینه به این الگوریتم شود.

۹- مراجع

[1]S. Keaveney and K. Conboy, "Cost estimation in agile development projects," in ECIS, 2006, vol. 169, pp. 183-197.
[2]M. Jorgensen and M. Shepperd, "A systematic review of software development cost estimation studies," IEEE Transactions on software engineering, vol. 33, no. 1, 2007.
[3]Y.-F. Li, M. Xie, and T. Goh, "A study of the non-linear adjustment for analogy based software cost estimation," Empirical Software Engineering, vol. 14, no. 6, pp. 603-643, 2009.
[4]V. K. Bardsiri, D. N. A. Jawawi, S. Z. M. Hashim, and E. Khatibi, "A PSO-based model to increase the accuracy of software development effort estimation," Software Quality Journal, vol. 21, no. 3, pp. 501-526, 2013.
[5]A. Tosun, B. Turhan, and A. B. Bener, "Feature weighting heuristics for analogy-based effort estimation models," Expert Systems with Applications, vol. 36, no. 7, pp. 10325-10333, 2009.
[6]J. Wojtusiak and R. S. Michalski, "The LEM3 implementation of learnable evolution model and its testing on complex function optimization problems," in Proceedings of the 8th annual conference on Genetic and evolutionary computation, 2006, pp. 1281-1288.
[7]N.-H. Chiu and S.-J. Huang, "The adjusted analogy-based software effort estimation based on similarity distances," Journal of Systems and Software, vol. 80, no. 4, pp. 628-640, 2007.
[8]G. Kadoda, M. Cartwright, L. Chen, and M. Shepperd, "Experiences using case-based reasoning to predict software project effort," in Proceedings of the EASE 2000 Conference, Keele, UK, 2000: Citeseer.
[9]E. Mendes, N. Mosley, and S. Counsell, "A replicated assessment of the use of adaptation rules to improve Web cost estimation," in 2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings., 2003, pp. 100-109: IEEE.
[10]T. R. Benala and R. Bandarupalli, "Least square support vector machine in analogy-based software development effort estimation," in 2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE), 2016, pp. 1-6: IEEE.
[11]T. R. Benala and R. Mall, "DABE: Differential evolution in analogy-based software development effort estimation," Swarm Evolutionary Computation, vol. 38, pp. 158-172, 2018.
[12]J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, "Systematic literature review of machine learning based software development effort estimation models," Information Software Technology, vol. 54, no. 1, pp. 41-59, 2012.
[13]M. Azzeh, D. Neagu, and P. I. Cowling, "Fuzzy grey relational analysis for software ffort estimation," Empirical Software Engineering, vol. 15, no. 1, pp. 60-90, 2010.
[14]M. Auer, A. Trendowicz, B. Graser, E. Haunschmid, and S. Biffl, "Optimal project feature weights in analogy-based cost estimation: Improvement and limitations," IEEE Transactions on Software Engineering, vol. 32, no. 2, pp. 83-92, 2006.
[15]A. B. Nassif, M. Azzeh, A. Idri, and A. Abran, "Software development effort estimation using regression fuzzy models," Computational intelligence and neuroscience, vol. 2019, 2019.
[16]S.-J. Huang, N.-H. J. I. Chiu, and s. technology, "Optimization of analogy weights by genetic algorithm for software effort estimation," vol. 48, no. 11, pp. 1034-1045, 2006.
[17]D. Wu, J. Li, and C. Bao, "Case-based reasoning with optimized weight derived by particle swarm optimization for software effort estimation," Soft Computing, vol. 22, no. 16, pp. 5299-5310, 2018.
[18]P. S. Kumar and H. Behera, "Role of soft computing techniques in software effort estimation: an analytical study," in

روش ارجاع به مقاله : ت. جاودانی گندمانی، م. دشتی. ارائه راهکاری نوین جهت بهبود تخمین هزینه توسعه نرم افزار مبتنی بر تشابه. دوفصلنامه محاسبات و سامانه های توزیع شده. سال پنجم، شماره اول، شماره پیاپی ۹، صفحه ۵۶ تا ۶۳، سال ۱۴۰۱.

How to cite: Taghi Javdani Gandomani, Maedeh Dashti. Improving Analogy-based Software Development Cost Estimation Using Learnable Evolution Model. Journal of Distributed Computing and Systems(JDCS), Vol 5, Issue 1, Page 56-63, 2022.

Computational Intelligence in Pattern Recognition: Springer, 2020, pp. 807-831.

[19]P. S. Kumar, H. S. Behera, A. Kumari, J. Nayak, and B. Naik, "Advancement from neural networks to deep learning in software effort estimation: Perspective of two decades," Computer Science Review, vol. 38, p. 100288, 2020.

[20]M. Azzeh, D. Neagu, and P. Cowling, "Improving analogy software effort estimation using fuzzy feature subset selection algorithm," in Proceedings of the 4th international workshop on Predictor models in software engineering, 2008, pp. 71-78.

[21]M. Azzeh, "A replicated assessment and comparison of adaptation techniques for analogy-based effort estimation," Empirical Software Engineering, vol. 17, no. 1-2, pp. 90-127, 2012.

[22]R. S. Michalski, "Learnable evolution model: Evolutionary processes guided by machine learning," Machine learning, vol. 38, no. 1-2, pp. 9-40, 2000.

تقی جاودانی گندمانی، لیسانس، فوق لیسانس و دکتری خود را به ترتیب از دانشگاه های صنعتی اصفهان، اصفهان و پوترا مالزی اخذ نموده و هم اینک به عنوان عضو هیات علمی دانشگاه شهرکرد فعالیت می نماید. زمینه های تحقیقاتی ایشان متدولوژی های نرم افزار و مهندسی نرم افزار تجربی می باشد.

مائده دشتی، فارغ التحصیل مهندسی کامپیوتر- نرم افزار از دانشگاه آزاد اسلامی واحد اصفهان می باشد و و هم اینک به عنوان محقق در واحد علوم داده دانشگاه شهرکرد فعالیت می نماید. زمینه های تحقیقاتی ایشان تخمین نرم افزار و بدهی های فنی می باشد.