دوره یازدهم، شماره اول، بهار و تابستان ۱۳۹۷

مجله فناوری اطلاعات در طراحی مهندسی

Information Technology in Engineering Design

http://ited.sinaweb.net

ارائه مدل جدیدی مبتنی بر روش نقطه مورد کاربری به‌منظور افزایش دقت تخمین تلاش در پروژه‌های نرم‌افزاری

وحید خطیبی بردسیری*(۱) الهام خطیبی(۱)

(۱) گروه مهندسی کامپیوتر، واحد بردسیر، دانشگاه آزاد اسلامی، کرمان، ایران.*

(تاریخ دریافت : ۱۳۹۷/۱/۱۱ تاریخ پذیرش : ۱۳۹۷/۲/۷)

چکیده

برآورد تلاش لازم برای پیشبرد یک پروژه نرم افزاری همواره یکی از دغدغه‌های اساسی مهندسین نرم افزار و مدیران پروژه بوده است. این موضوع در چند سال اخیر در کانون توجه محققین قرار گرفته و مدل‌های تخمین متنوعی در این حوزه ابداع شده است که عمدتا متکی بر ابزارهای هوش مصنوعی می‌باشند. اما در این زمینه پروژه‌های نرم‌افزاری شی‌گرا با وجود محبوبیت بسیاری که دارند کمتر مورد توجه قرار گرفته‌اند و مدل‌های بسیار محدودی برای این نوع پروژه‌ها در دسترس است. در این مقاله به منظور تخمین میزان تلاش به‌طور خاص در پروژه‌های نرم‌افزاری شی‌گرا، مدلی مبتنی بر وزن‌دهی به عوامل تاثیرگذار با الهام از روش نقطه مورد کاربری ارائه شده است. مدل پیشنهادی به منظور بهینه‌سازی و تغییر ساختار روش نقطه مورد کاربری از الگوریتم ازدحام ذرات استفاده می‌کند. به‌منظور افزایش کارایی، مدل پیشنهادی بر روی خودکارسازی فرآیند وزن‌دهی به عوامل کلیدی، تغییر نقش اندازه پروژه، تنظیم تخمین‌های موجود و انتخاب موثرترین ویژگی‌ها تکیه می‌کند. در مدل پیشنهادی با سازماندهی یک روند جدید، روش نقطه مورد کاربری بهبود داده می‌شود و در واقع سفارشی می‌شود. به‌منظور ارزیابی مدل پیشنهادی از پروژه‌های واقعی استفاده شده است و نتایج بدست آمده توسط مدل پیشنهادی با سایر مدل‌های رایج ارائه شده در این زمینه مقایسه شده است. نتایج مقایسه‌ها نشان دهنده برتری مدل پیشنهادی در اغلب معیارهای عملکردی است.

واژه‌های کلیدی: تخمین تلاش، روش نقطه مورد کاربری، پروژه شی گرا، الگوریتم ازدحام ذرات، وزن‌دهی

* **عهده‌دار مکاتبات:**

نشانی: گروه مهندسی کامپیوتر، واحد بردسیر، دانشگاه آزاد اسلامی، کرمان، ایران.

تلفن: ۰۹۱۳۱۴۱۳۹۲۴ **پست الکترونیکی:** kvahid2@live.utm.my

۱. مقدمه

بررسی‌های صورت گرفته نشان می‌دهد، که اکثر پروژه‌های نرم‌افزاری به دلیل برآورد غلط هزینه، برنامه‌ریزی و زمان‌بندی نادرست مدیران با شکست مواجه شده‌اند. در نتیجه برآورد صحیح پارامترهای پروژه نرم‌افزاری اهمیت زیادی پیدا کرده است [۱، ۲]. مدیریت پروژه، برنامه‌ریزی و هدایت پروژه در چهارچوب زمان، هزینه و کیفیت مشخص به منظور ایجاد نتایج مشخص است. مدیریت پروژه فعالیت‌های برنامه‌ریزی، سازماندهی، نظارت بر اجرا وهدایت اجرا را در برمی‌گیرد و سعی دارد تا با استفاده درست از منابع، نتایج مشخص و مورد انتظار را با هزینه توافق شده قبلی در موعد درست خود تحویل دهد. زمان و هزینه دو پارامتر مهم در هر پروژه نرم‌افزاری می‌باشند. در سال‌های اخیر تولید و توسعه نرم‌افزار در حال تبدیل شدن به یک سرمایه‌گذاری حیاتی برای بسیاری از سازمان‌ها است. بعنوان مهمترین فاکتور تعیین هزینه و زمان، تخمین تلاش لازم برای تولید و توسعه نرم‌افزار اهمیت روزافزونی در مدیریت موثر پروژه نرم‌افزاری پیدا کرده است. تخمین دقیق میزان تلاش لازم می‌تواند کمک‌های چشمگیری به فرآیند تصمیم‌گیری مدیران پروژه‌های نرم‌افزاری نماید. برای مثال تخمین دقیق تلاش می‌تواند به یک سازمان برای تحلیل بهتر، امکان‌سنجی پروژه و مدیریت موثر فرآیند توسعه نرم‌افزار کمک کند و در نتیجه ریسک تا حد زیادی کاهش می‌یابد. بنابراین خروجی‌های مدل تخمین تلاش بسیار مهم و تاثیرگذار است.

بر اساس گزارشی‌که توسط گروه استندیش در سال ۲۰۱۵ منتشر شده است تنها ۱۶/۲ ٪ پروژه‌های نرم‌افزاری با موفقیت به پایان رسیده‌اند و ۵۲/۷٪ پروژه‌ها با چالش‌های جدی روبرو شده و۳۱/۱٪ آنها نیز با شکست مواجه شده‌اند [۳]. این در حالی است که طبق گزارش ارائه شده توسط همین گروه در سال ۲۰۱۱ نرخ موفقیت، به چالش کشیده شدن و شکست پروژه‌های نرم‌افزاری در سال۳۷٪، ۴۲٪ و ۲۱٪ می‌باشد. مقایسه این آمار نشان از آن دارد که تعداد پروژه‌هایی که طی سال‌های ۲۰۱۱ تا ۲۰۱۵ به موفقیت رسیده‌اند ۵۷٪ کاهش یافته است. این عدد برای پروژه‌هایی که با چالش مواجه شده‌اند برابر با ٪ ۱۹ رشد بوده و برای پروژه‌های شکست خورده نیز برابر با ۳۲٪ رشد می‌باشد. این آمار بیانگر وضعیت بحرانی فرآیند مهندسی نرم‌افزار و بطور خاص مدیریت پروژه‌های نرم‌افزاری است.

هرچند عوامل متعددی مانند، برنامه‌ریزی ضعیف، عدم مشارکت مشتریان در تولید نرم‌افزار، عدم بیان دقیق نیازمندی‌ها از طرف مشتریان و توقعات غیر واقعی مشتریان را می‌توان در شکست پروژه‌های نرم‌افزاری دخیل دانست، اما اغلب محققان بر این باور هستند که دلیل اصلی شکست پروژه‌های نرم‌افزاری و یا به چالش کشیده شدن آنها، تخمین‌های نادقیق و غیر اصولی از هزینه و زمان لازم برای تولید نرم‌افزار می‌باشد که عموما از آن با عنوان تخمین تلاش نرم‌افزار یاد می‌شود [۴].

تخمین دقیق تلاش در پروژه‌های نرم‌افزاری ذاتا یک کار چالش‌برانگیز است. اگر چه تلاش‌های بسیاری برای حل این مشکل در دهه‌های گذشته صورت گرفته است، هیچ روش قطعی برای موفقیت موثر و مداوم در پیش‌بینی معیارهای مربوط به تخمین تلاش وجود ندارد.

نتایج تحقیقات اخیر نشان می‌دهد که مسأله تخمین تلاش در پروژه‌های نرم‌افزاری کماکان به عنوان یک مسأله حل نشده وجود دارد و محققان زیادی در تلاش‌اند تا در این زمینه بهبود لازم را ایجاد کنند [۵-۷]. در حوزه پروژه‌های نرم‌افزاری شی‌گرا با وجود اهمیت فزایندهای که دارند تحقیقات بسیار محدودی به منظور بهبود دقت تخمین تلاش انجام شده است و پارامترهای تاثیرگذار در پروژه‌های شی‌گرا کمتر مورد توجه قرار گرفته‌اند و روش‌های انگشت شماری در این زمینه معرفی شده است. لذا در این مقاله تلاش شده است تا با تمرکز بر روی یکی از روش‌های تخمین تلاش در پروژه‌های شی‌گرا بهبود قابل ملاحظه‌ای به لحاظ دقت و انعطاف‌پذیری از طریق معرفی یک مدل جدید ایجاد شود.

مقاله حاضر از ۷ بخش تشکیل شده است. پس از مقدمه، بخش دوم به پیشینه تحقیق می‌پردازد و ملزومات مدل پیشنهادی در بخش سوم ارائه می‌گردد. در بخش چهارم جزییات مدل پیشنهادی توضیح داده می‌شود و در بخش پنجم نتایج ارائه می‌شود. در بخش ششم به تحلیل و توجیه نتایج پرداخته می‌شود و نهایتا در بخش هفتم نتیجه‌گیری و جمع‌بندی ارائه می‌شود.

۱. کارهای پیشین

روش‌های تخمین تلاش در پروژه‌های نرم‌افزاری بسیار متنوع هستند به‌طوری‌که دسته‌بندی آنها قدری مشکل است. به‌صورت کلی می‌توان به روش‌های مبتنی بر تعداد خطوط کد [۹]، مدیریت چرخه حیات نرم‌افزاری [۱۰،۱۱]، روش‌های شی‌گرایی [۱۲]، نقطه تابع [۱۳] و مدل کوکومو ۱ و ۲ [۱۴،۱۵] اشاره کرد.

شناخته‌شده‌ترین تکنیک‌های تخمین تلاش، مدل کوکومو ۱ و کوکومو ۲ هستند. محققین بسیاری معیارهای ارزیابی خاصی را برای اعتبارسنجی مدل‌های تخمین مطرح کرده‌اند و به بررسی نقاط قوت و ضعف تکنیک‌های مختلف تخمین تلاش پرداخته‌اند. اما نتیجه این بررسی‌ها هرگز به یک توافق نهایی بر روی بهترین مدل تخمین منتهی نشده است [۱۶].

مدل کوکومو ۲ یک مدل عالی تا سال ۲۰۰۵ بود اما نیازهای جدید و روش‌های توسعه مدرن را در فرآیند تخمین تلاش برآورده نمی‌ساخت [۱۷]. مدل کوکومو ۲ متخصصین نرم‌افزاری را به سمت خلق و طراحی مدل‌های جدید از قبیل مدل دولت چینی و مدل هزینه ساختاری تجاری آزاد هدایت کرد.

محققان تأکید کرده‌اند که برای بازشدن افق‌های جدید در زمینه تخمین هزینه‌های تولید و توسعه نرم‌افزار نیاز به تحقیقات گسترده‌ای است. نقاط ضعف و قوت تکنیک‌های موجود تخمین تلاش به منظور فراهم کردن مبنایی برای تعیین میزان دقت تخمین‌ها در پروژه‌های نرم‌افزاری به طور گسترده در سال‌های اخیر مورد بحث و تحلیل قرار گرفته است [۱۸].

حساسیت تخمین تلاش در مدل کوکومو ۲ در مطالعات قبلی درک شده و دقت مورد نیاز با کمک سه روش به نام‌های آنالیز ریاضی تساوی تخمین، شبیه‌سازی مونت کارلو و انتشار خطاها به دست آمده است. محققین اثبات کرده‌اند که میزان حساسیت تلاش وابسته به سه فاکتور ضرب کننده تلاش، سایز و مقیاس وزن است و روش‌های فازی برای کمک به مساله عدم قطعیت و حساسیت در برآورد هزینه‌های نرم‌افزاری مورد استفاده قرار گرفتند [۱۳].

آنالیز نقطه تابع روشی است که فاکتورهای محیطی و تکنیکی مانند پیچیدگی، مهارت‌های توسعه دهنده و ریسک پروژه را مورد استفاده قرار می‌دهد [۱۹]. این روش به‌منظور تخمین تلاش در پروژه‌های نرم‌افزاری بسیار مورد استفاده قرار گرفته و در واقع مبنایی برای اندازه‌گیری سایز پروژه‌ها و نهایتا تخمین تلاش است.

تنها مدل موجود برای تخمین هزینه در پروژه‌های شی‌گرا روش نقطه مورد کاربری است که سال‌ها پیش معرفی شده است [۱۲]. در آزمایشی که برای اندازه‌گیری سایز با استفاده از نقطه مورد کاربری روی ۲۳ سیستم نرم‌افزاری با مقیاس بالا انجام شد، مشخص شد که روش نقطه مورد کاربری شاخصی مطمئن برای تعیین سایز پروژه است. در بحث طراحی شی‌گرا مشاهده شده است که مدل‌های مورد کاربری که در جزئیات با یکدیگر تفاوت دارند به لحاظ سایز نیز با هم کاملا متفاوت هستند. این ایده اولیه برای به‌کارگیری ابزارهای طراحی شی‌گرا در جهت تخمین تلاش و هزینه بوده است. زیرا با تخمین سایز قادر خواهیم بود به تخمین تلاش و هزینه دست پیدا کنیم [۲۰].

نتایج حاصل از یک پروژه آزمایشی که به منظور بهبود فرآیند برآورد هزینه و تلاش با استفاده از مدل مورد کاربری صورت گرفته است نشان می‌دهد که با مدل‌سازی مورد کاربری و بهبود آن مشتری و توسعه‌دهندگان درکی

یکسان از نیازها پیدا کرده و فرآیند تخمین تلاش بهبود قابل ملاحظه‌ای یافته است [۲۱].

استفاده از مبنای شی‌گرایی در فرآیند تخمین منجر می‌شود که پروژه‌های آزمایشی یک سربار را در تخمین‌ها تجربه کنند، اما این سربار کمتر از میانگین آن در پروژه‌هایی است که از روش‌های معمولی استفاده می‌کنند. بنابراین محققین دریافته‌اند که تحلیل موارد کاربری به عنوان پایه‌ای برای تخمین و طراحی در زمینه برآورد تلاش بسیار مفید است. در سال‌های گذشته روش‌های متعددی برای تخمین هزینه و تلاش بر اساس موارد کاربری معرفی شده است [۲۲].

در یکی از مدل‌های مبتنی بر شی‌گرایی، مدل مورد کاربری می‌تواند به عنوان پایه‌ای برای محاسبه نقاط تابع باشد که در جهت به دست آوردن تخمینی از تلاش مورد استفاده قرار می‌گیرد [۲۰]. در مدلی دیگر نقطه مورد کاربری برای تخمین تعداد خطوط کد در سیستم به کار گرفته می‌شود که این تعداد خطوط کد به عنوان ورودی مدل‌های تخمین استفاده می‌شود [۲۳].

در تلاش‌های صورت گرفته برای بهبود عملکرد روش نقطه مورد کاربری می‌توان نمونه‌هایی از بکارگیری روش‌های محاسبات نرم را مثال زد. روش‌های مبتنی بر رگرسیون [۲۲]، شبکه‌های عصبی [۲۰] و همینطور طبقه‌بندی [۲۱] از این دسته‌اند که توانسته‌اند تا حدی عملکرد روش نقطه مورد کاربری را بهبود بخشند.

همانطور که از ادبیات تحقیق پیداست، روش نقطه مورد کاربری با وجود توانمندی و پتانسیل بسیار بالا کمتر مورد توجه قرار گرفته است و وفور پروژه‌های شی‌گرا اهمیت بکارگیری این روش را دو چندان می‌کند. لذا در این مقاله تلاش شده است تا با بهینه‌سازی بخشی از فرآیند عملکردی روش نقطه مورد کاربری دامنه بکارگیری آن افزایش داده شده و دقت پیش‌بینی‌ها در آن بهبود یابد.

۲. روش‌های مورد استفاده

۳-۱. الگوریتم بهینه‌سازی ازدحام ذرات

الگوریتم بهینه‌سازی ازدحام ذرات (PSO) یک الگوریتم بهینه‌سازی فرااکتشافی است و به نام الگوریتم پرندگان نیز مشهور است و از حرکت گروهی پرندگان (و دیگر حیواناتی که به شکل گروهی زندگی می‌کنند) الگو گرفته است. این الگوریتم یک خانواده از روش‌های هوش جمعی و یکی از الگوریتم‌های موفق در زمینه بهینه‌سازی پیوسته و گسسته می‌باشد. این روش بهینه‌سازی اولین بار در سال ۱۹۹۵ با الهام از رفتار جمعی پرندگان و ماهی‌ها و به کارگیری مفاهیم الگوریتم‌های تکاملی معرفی شد [۲۴]. الگوریتم بهینه‌سازی ازدحام ذرات مشابه با الگوریتم‌های تکاملی یک الگوریتم جمعیتی بوده که در آن تعدادی ذره که راه‌حل‌های کاندیدای یک تابع یا یک مسئله هستند، یک ازدحام جمعیت را تشکیل می‌دهند. این ذرات در فضای مسئله حرکت کرده و براساس تجربیات فردی خود و تجربیات جمعی سعی می‌کنند تا راه حل بهینه در فضای جستجو را بیابند. این روش بوسیله ابعاد و غیرخطی بودن مسئله خیلی تحت تأثیر قرار نگرفته و نتایج خوبی در محیط‌های استاتیک و محیط‌های بطور پیوسته در حال تغییر می‌گیرد.

۳-۱-۱. مفاهیم اولیه

- هر ذره در حال جابه‌جایی است (در غیر این صورت نمی‌تواند جستجو کند).
- به دلیل این جابه‌جایی، دارای سرعت است.
- PSO بر مبنای حرکت و هوش ذرات کار می‌کند.
- PSO مفهوم تعامل اجتماعی را برای حل مسائل بهینه‌سازی به کار می‌برد.
- ذرات (پاسخ‌های مسأله) در فضای جستجو حرکت می‌کنند.

- هر ذره در هر مرحله موقعیتی که بهترین نتیجه را در آن داشته به خاطر می‌سپارد (بهترین موقعیت فردی هر ذره).
- ذرات در گروه ذرات با هم همیاری می‌کنند و اطلاعاتی دربارهٔ موقعیتی که در آن هستند را با هم تبادل می‌کنند.

حرکت ذره به ۳ عامل بستگی دارد:

۱.موقعیت فعلی ذره

۲.بهترین موقعیتی که تا کنون ذره داشته است (PBEST)

۳.بهترین موقعیتی که کل مجموعه ذرات تا کنون داشته‌اند (GBEST)

موقعیت جدید هر ذره از رابطه زیر به دست می‌آید:

$$Present[\,] = Present[\,] + v[\,] \qquad (۱)$$

در اینجا آخرین جمله سرعت (میزان تغییر مکان ذره) را مشخص می‌کند که بصورت زیر محاسبه می‌شود:

(۲)

$$V[\,] = C0*V[\,] + C1*rand(\,)*(Pbest[\,]-present) + C2*rand(\,)*(Gbest[\,]-present)$$

تابع rand مولد عدد تصادفی در بازهٔ [۰،۱] و C1 و C2 ثوابت اثر نامیده می‌شوند.

۳-۱-۲. پارامترهای الگوریتم بهینه‌سازی ذرات

در اجرای الگوریتم PSO پارامترهای زیادی دخیل هستند که تنظیم مناسب آن‌ها عملکرد الگوریتم را شدیداً تحت تأثیر قرار می‌دهد. برخی از این پارامترها به شرح زیراند:

تعداد ذرات: نشان داده شده است که معمولا ۱۰-۱۰۰ کافی است.

C1: اهمیت مربوط به بهترین وضعیت هر ذره

C2: اهمیت مربوط به بهترین وضعیت همسایگی‌ها

$C1 + C2 = 4$ فقط به دلایل تجربی انتخاب شده است.

ضرایب C1 و C2، به ترتیب میزان تاثیر پذیری از موقعیت محلی و عمومی را تعیین می‌کنند. اگر C1=0 باشد، ذره تنها وابسته به حرکت گروهی ذرات می‌باشد و ذرات از ارزیابی فردی خود استفاده نمی‌کنند. اگر C2=0، ذره هیچگونه ارزیابی از جمعیت انجام نمی‌دهد و تنها با حرکت اکتشافی خودش به جستجوی جواب می‌پردازد. برای توابع چند جوابه، بهتر است تا خاصیت اکتشافی ذره بالاتر رود و بنابرین C1>C2 انتخاب می‌گردد. همچنین مقدار زیاد C1 و C2 باعث حرکت سریع ذرات و امکان جستجوی فضای بزرگتری را می‌دهد که با کاهش دقت جستجوی فضا همراه خواهد شد، درحالی که مقادیر کم C1 و C2 منجر به حرکت آهسته ذرات و افزایش دقت جستجوی فضا می‌گردد.

۳-۲. مدل نقطه مورد کاربری

روش تخمین نقطه مورد کاربری یکی از روش‌های تخمین تلاش می‌باشد که توسط کارنر در سال ۱۹۹۳ به‌عنوان تز دانشگاهی معرفی شد [۱۲]. این روش امروزه به عنوان ابزاری جهت تخمین هزینه و زمان پروژه‌های نرم‌افزاری که از uml برای مدل‌سازی استفاده می‌کنند به کار می‌رود. در این روش با تعریف نقطه مورد کاربری‌های سیستم و تخصیص نفر-ساعت لازم برای پیاده‌سازی آنها، حجم فعالیت لازم تخمین زده می‌شود. هر مورد کاربری شامل سناریو یا سناریوهایی است.

علاوه بر مورد کاربری‌های سیستم واسطه‌های ارتباطی مورد کاربری با دنیای بیرون برای مثال پنجره‌های ویندوز و یا صفحات وب نیز وجود دارند که طراحی و پیاده‌سازی آن خود حجم کار قابل‌توجهی را می‌طلبد. بنابراین قدم اول تشخیص مورد کاربری‌ها و تشریح سناریوهای آنهاست. فرآیند تشخیص و تشریح مورد کاربری‌های سیستم هر چه با دقت بیشتری انجام شود، برآوردهای واقعی‌تری را خواهد داشت. مورد کاربری‌ها به عنوان مدلی از فعالیت‌های سیستم به طور کامل انتزاعی بوده و بسته به آن که چه کسی و از چه زاویه‌ای آن را می‌نویسد سطوح و پیچیدگی‌های مختلفی می‌توانند داشته باشند. برای مثال می‌توان صدور چک را یک مورد کاربری تلقی کرد و همزمان می‌توان صدور چک را زیرسیستمی معرفی نمود

که خود شامل تعداد مشخصی موردکاربری است. نتیجه آن که سطوح نمونه کاربری‌ها می‌توانند مختلف باشند و بنابراین در تعیین تعداد نقطه مورد کاربری‌ها باید دقت بیشتری مبذول نمود. به هرحال بهتر است که سطوح انتزاع در تمامی‌سیستم از یک روال ثابتی پیروی کنند، در غیر این صورت باید ضریب سطح انتزاع نیز در معادلات مربوط به نقطه مورد کاربری در نظر گرفته شود.اندازه ی نقطه مورد کاربری بوسیله شمارش تعداد مورد کاربری‌ها و عامل‌ها که در وزن‌های پیچیدگی خود ضرب شده‌اند محاسبه می‌شوند. میزان هر یک از نقطه کاربری‌ها بوسیله شمارش تراکنش‌های موفق و گستردگی سناریوها در سناریو کاربری‌ها است [۲۵]. مراحل روش نقطه مورد کاربری برای تخمین به شرح ذیل می‌باشند:

۳-۲-۱. تعیین وزن عامل تنظیم نشده

دسته بندی عوامل سیستم در این بخش جزییات مشخص می‌شود. در جدول (۱) دسته بندی عوامل آمده است. ستون دوم راهنمای تصمیم گیری در مورد نوع عامل بوده و نشان می‌دهد که عامل در کدام دسته قرار داده می‌شود. آخرین ستون نیز ارزش پیچیدگی آن را نشان می‌دهد.

۳-۲-۲. تعیین وزن نقطه مورد کاربری تنظیم نشده

شمارش تعداد مورد کاربری و تعیین وزن آنها برحسب تعداد سناریوها و تراکنش‌های آنها در این بخش انجام می‌شود. در جدول ۲ نرخ پیچیدگی کاربری‌ها نمایش داده می‌شود.

جدول ۱- نرخ پیچیدگی عامل‌ها [۱۲]

گروه	تست لازم برای تشخیص	ارزش
عامل ساده	عواملی هستند که با سیستم از طریق APIها در ارتباط هستند.	۱
عامل متوسط	عواملی که با سیستم از طریق پروتکل‌های HTTP,FTP و گاهی پروتکل‌های تعریف شده توسط کاربران ارتباط دارند. عواملی که به عنوان نگهدارنده داده‌ها هستند (فایل‌ها و RDBS)	۲
عامل پیچیده	عواملی هستند که با سیستم از طریق GUIها در ارتباط هستند.	۳

جدول۲- نرخ پیچیدگی موارد کاربری [۱۲]

نوع موردکاربری	شرط لازم برای دسته‌بندی	ارزش
ساده	کمتر یا مساوی ۳ تراکنش دارند.	۵
متوسط	بین ۴ تا ۷ تراکنش دارند.	۱۰
پیچیده	بیشتر از ۷ تراکنش دارند.	۱۵

در ادامه تعیین مجموع UUCP[1] با استفاده از فرمول (۳) بصورت زیر انجام می‌شود:

$$UUCP = \text{مجموع } UAW + \text{مجموع } UUCW \qquad (۳)$$

۳-۲-۳. محاسبه عوامل تکنیکی و محیطی

آخرین قدم برای محاسبه پیچیدگی، تعیین و اندازه‌گیری عوامل تکنیکی و محیطی سیستم است. عوامل تکنیکی ۱۳ مورد شناخته شده دارند هر چند می‌توان عوامل دیگری را نیز به آن اضافه نمود.

[1] Unadjusted Use Case Point (UUCP)

به هر یک از عوامل تکنیکی مقادیر ۰ تا ۵ نسبت داده می‌شود. مجموع عوامل تکنیکی فاکتور پیچیدگی تکنیکی پروژه را تعیین کرده و با ضرب آن در ضریب پیچیدگی، میزان پیچیدگی پروژه محاسبه می‌شود. هر عامل تکنیکی وزنی نیز دارد که میزان تاثیر آن را مشخص می‌کند. جدول (۳) عوامل تکنیکی را نمایش می‌دهد.

فاکتور تکنیکی: برای محاسبه فاکتور تکنیکی پروژه از معادله (۴) استفاده می‌شود:

$$Tfactor = T_1 + T_2 + \dots\dots T_{12} + T_{13} \quad (۴)$$

پیچیدگی تکنیکی: محاسبه میزان پیچیدگی تکنیکی پروژه با فرمول (۵) انجام می‌شود:

$$TCF = 0.6 + (0.01 * Tfactor) \quad (۵)$$

عوامل دیگری نیز هستند که باید در نظر گرفته شوند از جمله عامل محیطی که اثر مستقیم بر روی زمان و هزینه پروژه خواهد داشت. عوامل دیگری نیز هستند که باید در نظر گرفته شوند از جمله عامل محیطی که اثر مستقیم بر روی زمان و هزینه پروژه خواهد داشت. مجموع عوامل محیطی (فاکتور محیطی) از جمع مقادیر محیطی محاسبه می‌شود:

$$Efactor = SUM(e1 \dots e8) \quad (۶)$$

برای محاسبه ضریب عامل محیطی از معادله (۷) استفاده می‌شود:

$$EF = 1.4 + (0.03 * Efactor) \quad (۷)$$

درنهایت مقدار نقاط کاربری تنظیم شده با استفاده از فرمول (۸) محاسبه می‌شود:

$$AUCP = UUCP * TCF * EF \quad (۸)$$

۳-۲-۴. پارامترهای مورد استفاده در مدل پیشنهادی

ارزیابی قابلیت اطمینان مدل‌های تخمین در پروژه‌های نرم‌افزاری بسیار مهم است. یک عامل اصلی در انتخاب الگوریتم تخمین تلاش، دقت پیش‌بینی آن در امر تخمین است. جهت بررسی میزان دقت مدل پیشنهادی از معیارهای زیر استفاده شده است:

(۹) (خطای نسبی) $$E = |Estimated - Actual|$$

(۱۰) (مقدار خطای نسبی)

$$MRE = |Estimated - Actual| / Actual$$

(۱۱) (میانگین مقدار خطای نسبی)

$$MMRE = \frac{\sum_{i=1}^{N} MRE_i}{N}$$

که در آن N تعداد تخمین‌ها می‌باشد. اساس محاسبه مقدار خطای نسبی و میانگین مقدار خطای نسبی استفاده از خطای نسبی می‌باشد. میانگین، با در نظر گرفتن مقدار تلاش واقعی و برآورد آن از مجموعه داده‌های به کار رفته محاسبه می‌شود. وقتی چندین پروژه با خطاهای نسبی بزرگ وجود دارد، ارزیابی قدرت برآورد مدل ممکن است منحرف شود. یک جایگزین برای میانگین، میانه می‌باشد که یک سنجش مرکزگرا را نشان می‌دهد بطوری‌که کمتر به وجود چندین خطای نسبی بزرگ حساس است. میانه مقدار خطای نسبی برای تعداد i مشاهده، میانه بزرگی خطای نسبی نامیده می‌شود. برای به دست آوردن میانه، داده‌ها به ترتیب صعودی لیست می‌شوند، اگر تعداد داده‌ها فرد باشد داده وسطی و اگر زوج باشد میانگین دو داده وسطی به عنوان میانه در نظر گرفته می‌شود. میانگین بزرگی خطای نسبی و میانه بزرگی خطای نسبی به عنوان معیارهای ارزیابی عملی در نظر گرفته می‌شود تا قدرت برآورد مدل‌های تخمین تلاش را اندازه‌گیری کنند.

(۱۲) (میانه خطای نسبی)

$$MdMRE = Median(MRE)$$

(۱۳) (درصد پیش بینی)

$$PRED(X) = A/N$$

علاوه بر این، درصد پیش‌بینی معیار دیگری است که باید حداکثر شود. در این معیار A تعداد پروژه‌ها با MRE کمتر یا مساوی X وN تعداد کل پروژه‌های مورد نظر است. معمولا در حوزه تخمین تلاش مقدار ۰/۲۵ برای X در نظر گرفته می‌شود [۲۶].

جدول ۳- عوامل تکنیکی [۱۲]

	عامل تکنیکی	وزن	توصیف
T_1	سیستم توزیع شده	۲	آیا سیستم معماری توزیع شده دارد یا متمرکز است؟
T_2	زمان پاسخگویی/ هدف عملکرد	۱	آیا مشتری و یا کارفرما به سیستم پر سرعت نیاز دارد؟
T_3	کارایی کاربران نهایی	۱	میزان کارایی و قابلیت‌های کاربران نهایی چگونه است؟
T_4	پروسس‌های پیچیده داخلی	۱	آیا فرآیندهای کاری بسیار پیچیده‌اند؟
T_5	کد قابل استفاده مجدد	۱	آیا هدف این است که قابلیت استفاده مجدد اجزای سیستم بالا نگهداشته شود
T_6	راحتی نصب	۰.۵	آیا مشتری نصب آسان را دنبال میکند؟
T_7	استفاده آسان	۰.۵	طراحی صفحات و پنجره‌ها جهت استفاده آسان اولویت بالا دارد؟
T_8	قابلیت نقل و انتقال	۲	آیا مشتری به دنبال محیط اجرا و پیاده سازی چندگانه است؟
T_9	آسانی تغییرات	۱	آیا مشتری به دنبال تغییرات فراوان در آینده است؟
T_{10}	همزمانی	۱	آیا تعداد کاربرانی که همزمان از سیستم استفاده میکنند زیاد است؟
T_{11}	امنیت	۱	آیا مشتری به دنبال رعایت امنیت سنگین روی برنامه بوده وکدگذاری اهمیت دارد
T_{12}	دسترسی مستقیم به شرکت‌های ثالث	۱	آیا سیستم به کنترل شرکت‌ها و کمپانی‌های ثالث یا سیستم‌های مکانیزه آنها مرتبط است بنابراین تلاش قابل توجهی نیاز است
T_{13}	امکانات آموزشی‌کاربر	۱	آیا نرم‌افزار از دیدگاه کاربر آنقدر پیچیده است آموزش مجزایی در نظر گرفته شود

جدول۴- عوامل محیطی [۱۲]

	عامل محیطی	وزن	توصیف
E_1	آشنایی با محیط پروژه	۱.۵	آیا همه دست‌اندرکاران پروژه با حوزه سیستم و جزییات تکنییکی پروژه آشنا هستند در غیر این صورت به احتمال زیاد زمان زیادی روی توصیف و شرح نیازهای حوزه سیستم صرف خواهد شد
E_2	تجربه تیم	۰.۵	تجربه تیم در زمینه برنامه‌های کاربردی چقدر است؟
E_3	تجربه شی‌گرایی	۱	از آنجا که شرح یوزکیس به عنوان منبعی برای طراحی قلمداد می‌شود بنابراین بسیار مهم است که تیم مفاهیم اساسی مهندسی نرم‌افزار شی‌گرا را به خوبی بداند.
E_4	قابلیت هدایت تحلیل‌گر	۰.۵	وضعیت تحلیل‌گر سیستم از نظر آشنایی با محیط سیستم چگونه است؟
E_5	انگیزه	۱	آیا هدف این است که قابلیت استفاده مجدد اجزای سیستم بالا نگهداشته شود آیا برنامه نویسان انگیزه لازم برای کار روی پروژه را دارند؟ وضعیت بخش تولید نرم‌افزار چگونه است؟
E_6	ثابت بودن نیازهای سیستم	۲	آیا مشتری یا کاربر به درستی می‌داند چه می‌خواهد؟ واضح است که نیازهای کاربران مهم‌ترین فاکتور ثابت بودن نیازهای یک سیستم است کاربری که خود نمی‌داند چه می‌خواهد و یا طبیعتی تغییرپذیر دارد این فاکتور را افزایش می‌دهد.
E_7	عضو پاره وقت	۱	آیا افراد پاره‌وقتی مانند مشاور و غیره در پروژه وجود دارد؟
E_8	زبان برنامه‌نویسی مشکل	۲	پیچیدگی زبان برنامه‌نویسی انتخاب شده چگونه است؟

۳. ارائه مدل پیشنهادی

مدل پیشنهادی از دو گام آموزش و تست تشکیل شده است که در ادامه به آنها بیشتر پرداخته خواهد شد. در گام آموزش هدف یافتن هشت وزن مناسب برای اختصاص دادن به عوامل محیطی و همینطور یافتن یک توان مناسب برای تغییر مقدار سایز پروژه است که در نهایت منجر به تغییر ساختار UCP می‌شود. علاوه بر این یک عامل تنظیم کننده نیز به‌منظور تنظیم نهایی مقدار تلاش باید شناسایی شود. در مرحله تست ۹ وزن و همینطور یک مقدار تنظیم کننده یافت شده در مرحله آموزش مورد استفاده قرار می‌گیرند.

۴-۱. گام آموزش در مدل پیشنهادی

در ابتدا تعدادی پروژه نرم‌افزاری وجود دارد که هر یک از این پروژه‌ها شامل تعدادی ویژگی می‌باشند. ویژگی‌های هر پروژه شامل مقادیر عوامل محیطی و تکنیکی و همینطور سایز و میزان تلاش است. کل پروژه‌ها با نسبت ۷۰ به ۳۰ بصورت تصادفی به دو مجموعه آموزشی و تست تقسیم می‌شوند. یک پروژه به صورت تصادفی از میان پروژه‌های آموزشی‌انتخاب می‌شود و به الگوریتم ازدحام ذرات ارسال می‌گردد. در مرحله اول الگوریتم ازدحام ذرات یک رشته صفر و یک به تعداد ویژگی‌ها پیشنهاد می‌کند که صفر به معنای عدم انتخاب ویژگی متناظر و یک به معنی انتخاب آن است. تلاش پروژه انتخابی بر اساس ویژگی‌های انتخابی پیش‌بینی می‌شود و این کار برای همه پروژه‌های آموزشی انجام می‌شود. با پایان تکرارهای الگوریتم، بهترین مجموعه ویژگی‌ها انتخاب می‌شود. در هر تکرار تابع شایستگی MMRE-PRED خواهد بود و مجموعه‌ای از ویژگی‌ها که با محاسبه تلاش بر مبنای UCP کمترین مقدار را در تابع شایستگی تولید می‌کند انتخاب می‌شود. در واقع در هر تکرار از الگوریتم یک مجموعه از ویژگی‌ها بررسی می‌شود. در گام بعد مجددا یک پروژه انتخاب شده و الگوریتم ازدحام ذرات برای پروژه انتخاب شده ۹ وزن و یک مقدار تنظیم کننده پیشنهاد می‌دهد که مقدار هشت وزن اول بین۰ تا ۱ است. دلیل انتخاب بازه صفر تا یک استانداردسازی وزن‌دهی به همه ویژگی‌ها و یکسان‌سازی مقیاس تاثیرگذاری است. بازه مقدار وزن نهم از طریق سعی و خطا و باتوجه به مقادیر تلاش پروژه‌های آموزشی تعیین می‌شود که در مقاله حاضر این مقدار بین ۰ تا ۲ تعیین شده است. نهایتا بازه مقدار متغیر تنظیم‌کننده بین صفر و میانگین تلاش پروژه‌های آموزشی تعیین شده است. شکل (۱) مراحل گام آموزش را نمایش می‌دهد. در فرآیند وزن‌دهی به هر کدام از عوامل محیطی (e_1 تا e_8) که تاثیر مستقیم بر تخمین پروژه دارند و قبلا مورد بررسی قرار گرفته‌اند یک وزن اختصاص داده می‌شود و مقدار M با استفاده از رابطه (۱۴) از مجموع حاصل‌ضرب هر عامل محیطی در وزن خود محاسبه می‌شود.

$$M=W_1*F_1+W_2*F_2+.....W_8*F_8 = \sum_{i=1}^{8} W_i F_i \qquad (۱۴)$$

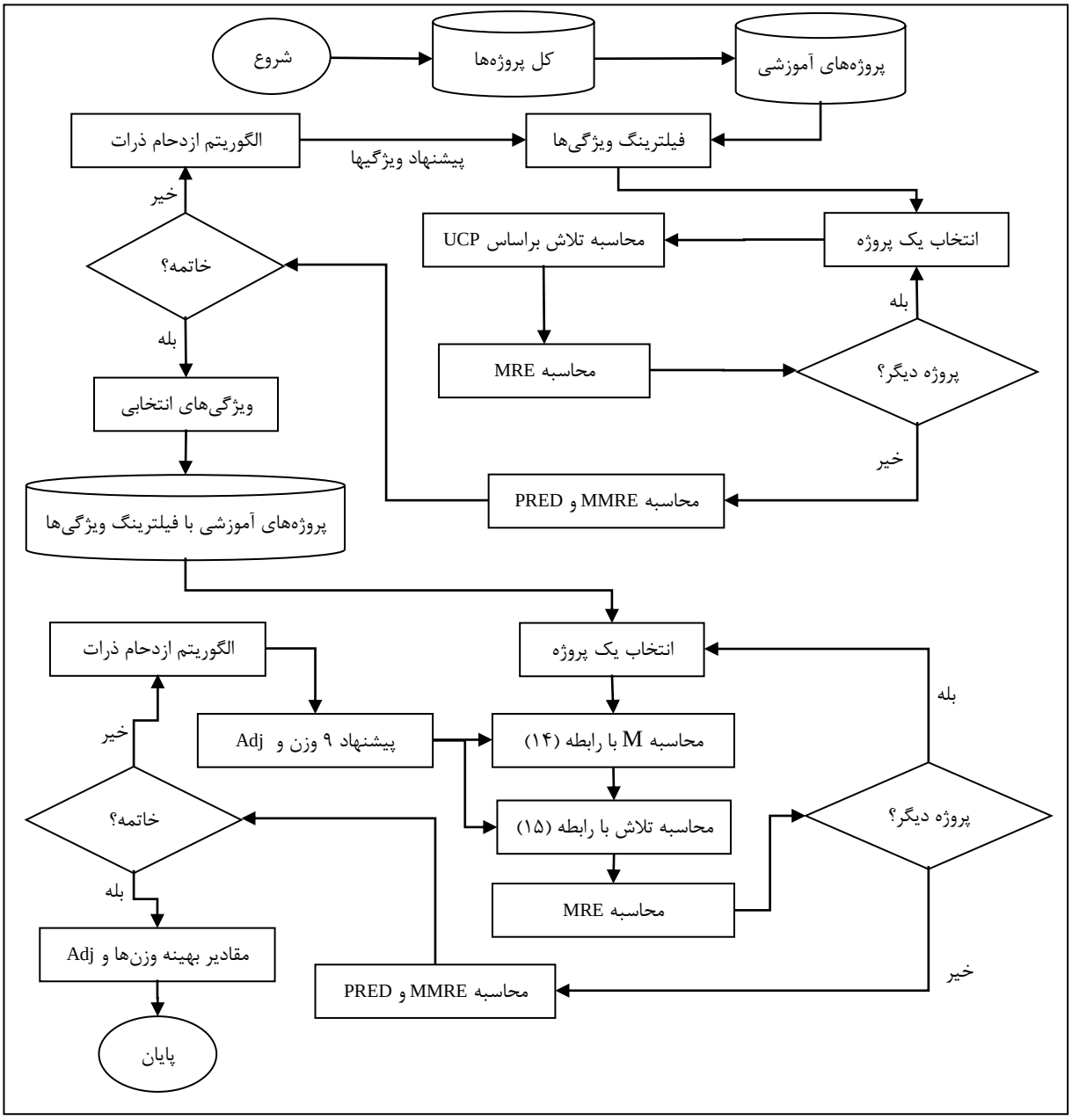
از آنجا که در محاسبه M تعداد ۸ وزن به عوامل محیطی اختصاص داده شد در این مرحله برای محاسبه تلاش از نهمین وزن استفاده می‌شود که این وزن به عنوان توان سایز پروژه درنظر گرفته می‌شود زیرا سایز پروژه باید طوری تغییر کند که تاثیر عمیقی در محاسبه تلاش پروژه داشته باشد.

در نهایت به‌منظور محاسبه میزان تلاش، یک فاکتور تنظیم کننده بنام Adj در نظر گرفته می‌شود که مقدار آن بین صفر و میانگین تلاش پروژه‌های آموزشی تعیین می‌شود. هدف از درنظر گرفتن این فاکتور کمک به رابطه ابداعی برای نزدیک شدن تخمین‌ها به مقادیر واقعی است. بهترین مقدار برای Adj توسط الگوریتم بهینه‌سازی ازدحام ذرات تعیین می‌شود. نهایتا میزان تلاش با استفاده از رابطه (۱۵) محاسبه می‌شود.

$$Effort = Size^{w9} * M + Adj \qquad (۱۵)$$

پس از محاسبه تلاش در مرحله بعد به محاسبه میزان MRE پروژه پرداخته می‌شود. پس از محاسبه MRE اگر پروژه دیگری برای بررسی وجود داشت این فرآیند تا محاسبه MRE مربوط به همه پروژه‌های آموزشی با یک

مجموعه ۹تایی از وزن‌ها و یک مقدار تنظیم کننده ادامه می‌یابد. یعنی برای همه پروژه‌های انتخاب شده MRE با ۹ وزن محاسبه می‌شود که تعداد هشت وزن در محاسبه M و یک وزن به همراه مقدار تنظیم کننده Adj در محاسبه تلاش استفاده شده است و در غیر این صورت اگر پروژه دیگری وجود نداشت مقدار MMRE و PRED با استفاده از MREهای محاسبه شده در کل پروژه موجود محاسبه می‌شود. هم اکنون یک تکرار الگوریتم ازدحام ذرات به پایان رسیده و مقدار تابع شایستگی باید محاسبه شده و به تکرار بعدی الگوریتم فرستاده شود که در مدل پیشنهادی این تابع مقدار تفاضل MMRE و PRED محاسبه شده در مرحله قبل است. دلیل این تفاضل این است که همیشه باید مقدار MMRE مینیمم و PRED ماکزیمم باشد و تابع ارسال شده که تابع هدف یا شایستگی الگوریتم ازدحام ذرات است باید به صورت مینیمم باشد چون در این صورت MMRE مینیمم و PRED ماکزیمم می‌شود. اگر تابع شایستگی منحصر به یکی از دو پارامتر ارزیابی باشد، احتمال انحراف در جواب‌ها بسیار زیاد خواهد بود چون MMRE و PRED مکمل یکدیگر هستند.



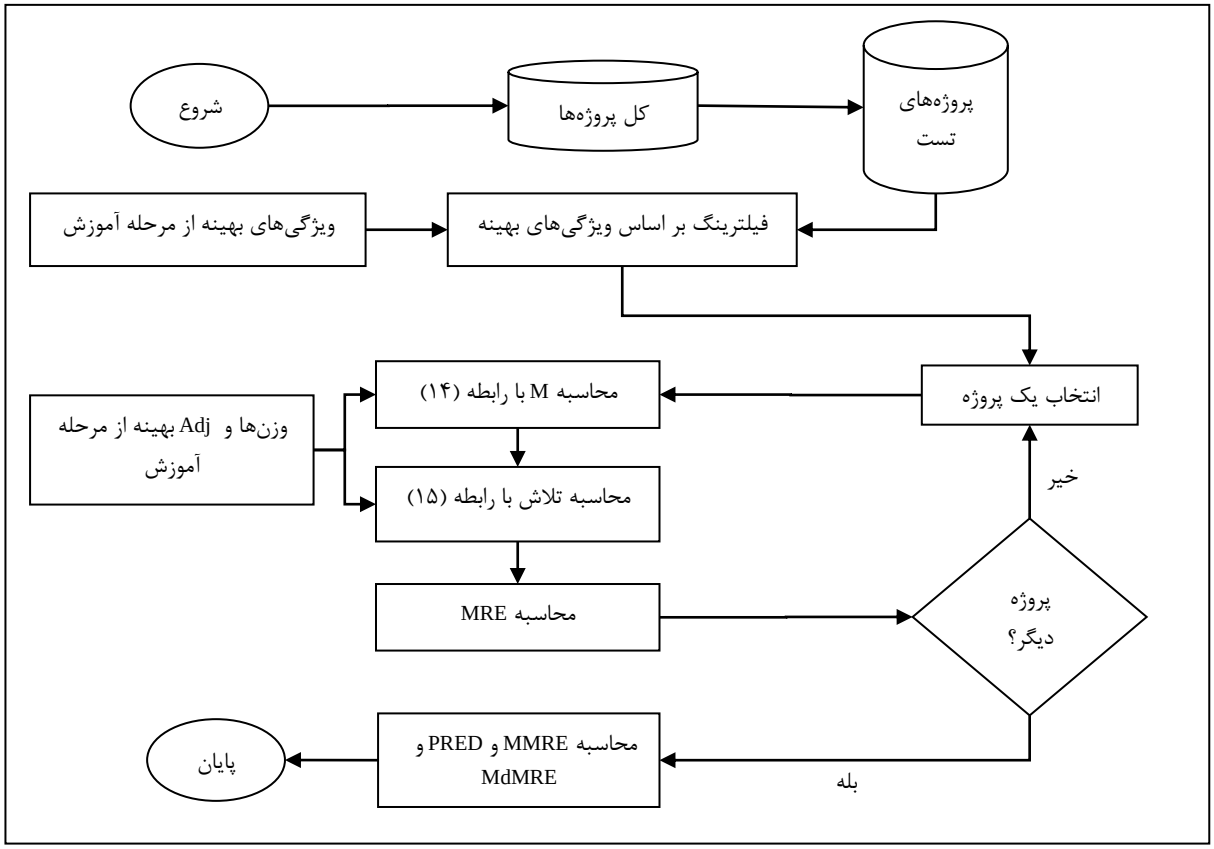
شکل ۱- فلوچارت روش پیشنهادی (گام آموزش)

بنابراین به ازای ۹ وزن و یک مقدار تنظیم کننده که الگوریتم ازدحام ذرات پیشنهاد می‌دهد یک مقدارMMRE و PRED وجود دارد و در هر تکرار الگوریتم ازدحام ذرات ۹ وزن و یک Adj جدید پیشنهاد می‌دهد و این روند تا زمان رسیدن به شرط خروج الگوریتم ادامه می‌یابد. زمان رسیدن به شرط خروج الگوریتم، بهترین ۹ تایی از وزن‌ها به‌همراه بهترین Adj ذخیره می‌شود و مرحله آموزش به پایان می‌رسد.

۴-۲. گام تست در مدل پیشنهادی

در این مرحله نیز تعدادی پروژه تست وجود دارد که از میان این پروژه‌ها، یک پروژه به صورت تصادفی انتخاب می‌شود. در این گام تنها ویژگی‌هایی مورد استفاده قرار می‌گیرند که در گام آموزش بعنوان ویژگی‌های بهینه در گام اموزش تعیین شده‌اند. در ادامه ۹ وزن به پروژه انتخاب شده اختصاص داده می‌شود که همان وزن‌های بهینه ذخیره شده در مرحله آموزش است. علاوه بر این مقدار Adj ارسال شده از مرحله آموزش نیست در گام تست استفاده می‌شود.

با اختصاص ۸ وزن به هر یک از عوامل محیطی و با استفاده از رابطه (۱۴) مقدار M محاسبه می‌شود و پس از آن با استفاده از نهمین وزن مقدار تلاش براساس رابطه (۱۵) محاسبه می‌شود. بعد از محاسبه تلاش پروژه انتخابی، مقدار MRE محاسبه می‌شود. در مرحله بعد اگر پروژه دیگری وجود داشت این فرآیند تا محاسبه MRE مربوط به همه پروژه‌های تست ادامه می‌یابد و در غیر این صورت اگر پروژه دیگری وجود نداشت با استفاده از MRE های محاسبه شده مقدار MMRE و PRED و MDMRE محاسبه شده و مرحله تست به پایان میرسد. در واقع مقادیر این سه پارامتر عملکرد مدل پیشنهادی را در گام تست مشخص می‌کند. شکل (۲) مراحل گام تست را نمایش می‌دهد.



شکل ۲- فلوچارت روش پیشنهادی (گام تست)

۴. ارائه نتایج

در این بخش مقایسات انجام شده مربوط به ۱۱۰ پروژه نرم‌افزاری کانادایی توضیح داده شده است. در ابتدا پایگاه داده و محتویات آن بررسی می‌شود، سپس مروری بر روش ارزیابی و روش‌های مقایس‌های رگرسیون صورت می‌گیرد و بعد از آن نمودارهای مقایسه‌های مربوطه نمایش و توضیح داده می‌شود و نتایج به‌دست آمده با روش‌های دیگر مقایسه می‌شود.

۵-۱. معرفی پایگاه داده

جهت بررسی عملکرد واقعی و بررسی کارایی مدل برآورد باید از داده‌های واقعی استفاده شود. در این راستا از اطلاعات ۱۱۰ پروژه‌ی نرم‌افزاری در قالب دو پایگاه داده که در کانادا جمع‌آوری شده است استفاده شده است [۲۷]. پایگاه داده اول شامل ۶۵ پروژه، ۸ عامل محیطی، یک‌اندازه پروژه بر اساس UCP و در نهایت تلاش پروژه می‌باشد و پایگاه داده دوم نیز شامل ۴۵ پروژه، ۸ عامل محیطی، یک اندازه پروژه و تلاش پروژه می‌باشد. در واقع مجموع داده‌های گردآوری شده مربوط به ۱۱۰ پروژه نرم‌افزاری در قالب ۲ پایگاه داده می‌باشد که برای اعمال تنظیمات و در نهایت مقایسه مدل پیشنهادی با مدل‌های دیگر استفاده می‌شود.

۵-۲. نتایج تجربی

به‌منظور ارزیابی عملکرد مدل پیشنهادی، نتایج بدست آمده با پنج مدل رایج تخمین تلاش مقایسه می‌شود که عبارتند از: CART[1]، SWR[2]، MLR[3]، ABE[4] و KARNER. روش CART ایده درخت‌های تصمیم‌گیری را برای طبقه‌بندی استفاده کرده و آن را بر اهداف پیوسته منطبق می‌کند [۳۰]. این روش قادر است بطور خودکار رگرسیون و یا طبقه‌بندی را بر اساس طیف ورودی‌ها انجام دهد و بدین جهت از قابلیت بسیار بالایی برخوردار است. SWR یک روش رگرسیون گام‌به‌گام آماری است که در آن مدلی پیش‌بینی کننده ساخته شده و رابطه بین متغیرهای مستقل و متغیرهای وابسته به‌صورت تدریجی و مرحله به مرحله با افزایش و یا کاهش ویژگیها تعیین می‌شود [۳۰]. MLR رگرسیون خطی چندگانه است که در بین روش‌های رگرسیون بدلیل سرعت و دقت قابل قبول بسیار متداول است [۳۰]. ABE یا برآورد مبتنی برقیاس رایج‌ترین روش غیرالگوریتمی تخمین تلاش است که به عنوان مدل مقایس‌های ارائه شده است [۳۱]. این روش به وفور در سال‌های اخیر مورد استفاده قرار گرفته است. نهایتا روش KARNER یک مدل شی‌گرا است که قبلا توضیح داده شد.

لازم به توضیح است که نتایج ذکر شده در این مقاله میانگین ۳۰ اجرای مستقل می‌باشد و جمعیت اولیه و تعداد تکرار بترتیب ۱۰۰ و ۲۰۰ در نظر گرفته شده است و پارامترهای C_1 و C_2 و W به‌ترتیب مقادیر ۲و۲و۱ را دریافت کرده‌اند.

۵-۲-۱. نتایج روی پایگاه داده اول

همان‌گونه که مشاهده می‌شود جدول (۵) دارای ۴ ستون (روش‌ها، MMRE، MDMRE، PRED) می‌باشد که نتایج به‌دست آمده از مدل پیشنهادی را نشان می‌دهد. این مقادیر برای به‌دست آوردن میزان دقت مدل پیشنهادی، در مقایسه با سایر مدل‌ها ارائه شده‌اند. بر اساس نتایج، مقدار به‌دست آمده برای پارامتر PRED توسط مدل پیشنهادی نسبت به سایر مدل‌های ذکر شده دقت بالاتری دارد. مقدار پارامتر MMRE برای مدل پیشنهادی نسبت به مقادیر روش‌های دیگر بسیار کمتر است. همچنین مقدار به‌دست آمده برای پارامتر MdMRE توسط مدل پیشنهادی از بقیه‌ی مدل‌ها کمتر است. مجموع نتایج به‌دست آمده از عملکرد قابل قبول روش پیشنهادی حکایت دارد.

[1] Classification and Regression tree (CART)
[2] Step Wise Regression (SWR)
[3] Multiple Linear Regression (MLR)
[4] Analogy Based Estimation(ABE)

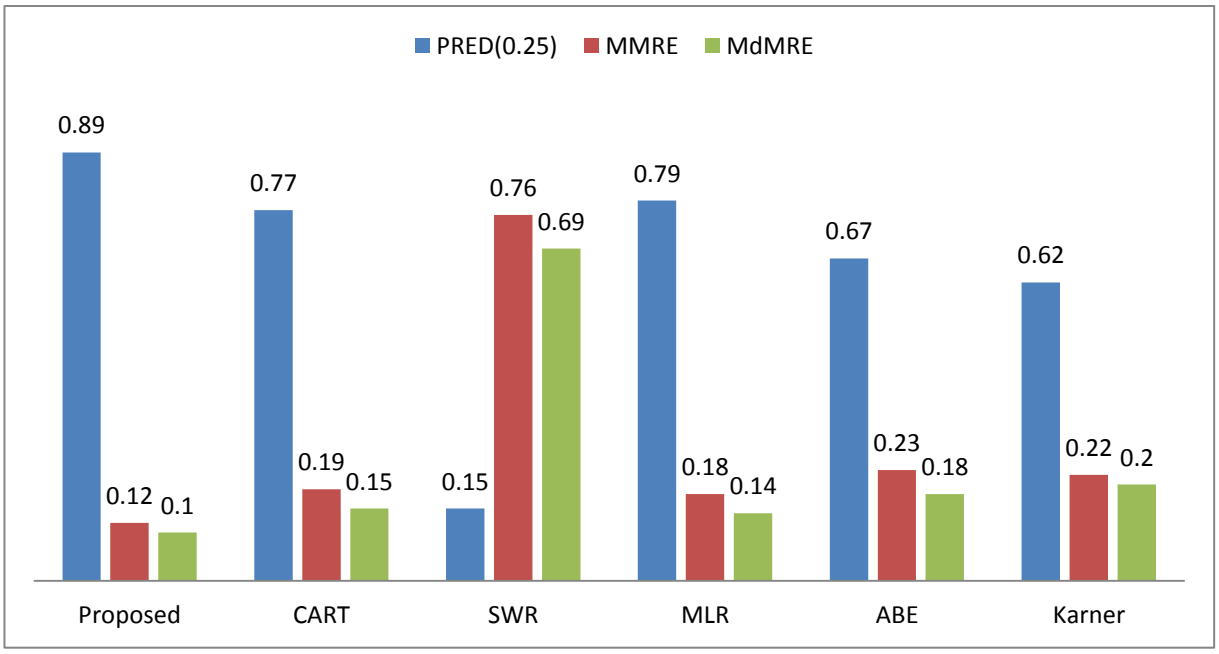
جدول ۵- نتایج بدست آمده روی پایگاه داده اول

مدل‌ها	PRED(25)	MMRE	MdMRE
Proposed	*0.89*	*0.12*	*0.10*
CART	0.77	0.19	0.15
SWR	0.15	0.76	0.69
MLR	0.79	0.18	0.14
ABE	0.67	0.23	0.18
Karner	0.62	0.22	0.2

همان‌گونه که ملاحظه می‌شود مقدار PRED برای مدل پیشنهادی ۰/۸۹ است که با توجه به اختلاف زیاد نسبت به سایر مدل‌ها نشان از برتری مدل پیشنهادی نسبت به سایر مدل‌ها دارد. MLR با مقدار ۰/۷۹ در رتبه دوم قرار گرفته است و مدل SWR با مقدار ۰/۱۵ پایین‌ترین جایگاه را به خود اختصاص داده است.

از سوی دیگر همان‌طور که در جدول مشاهده می‌شود بهترین MMRE به‌دست آمده توسط مدل پیشنهادی با مقدار ۰/۱۲ می‌باشد که بیانگر کارایی بالا مدل پیشنهادی است و بدترین میزان مربوط به مدل SWR با مقدار ۰/۷۶ است. بعداز مدل پیشنهادی، مدل MLR کمترین MMRE را به خود اختصاص داده است و همچنین مدل ABE بعد از مدل SWR بیشترین مقدار MMRE را نمایش می‌دهد.

در خصوص پارامتر MdMRE، مدل پیشنهادی با میزان ۰/۱۰ کمترین مقدار را دارد و مدل SWR بیشترین خطا را شامل می‌شود. شکل (۳) سه معیار اصلی را برای تمام مدل‌ها برای مقایسه‌ی بهتر در کنار یکدیگر قرار داده است.



شکل۳- نمودار مقایسه عملکرد روش‌ها در پایگاه داده اول

۵-۲-۲. نتایج روی پایگاه داده دوم

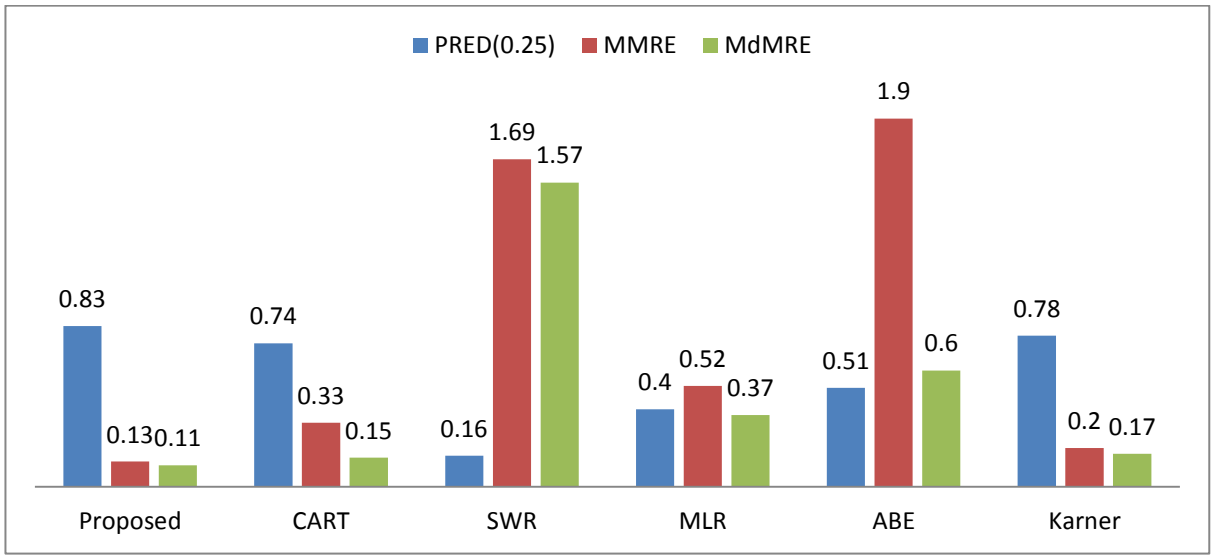
جدول (۶) نتایج بدست آمده بر روی پایگاه داده دوم را نمایش می‌دهد. این پایگاه داده مشابه با پایگاه داده اول و حاوی پروژه‌های واقعی می‌باشد که به تجزیه و تحلیل آن به صورت دقیق‌تر پرداخته می‌شود. مقیاس PRED که یکی از مهم‌ترین معیارهای مقایسه در اینجا می‌باشد با نمایش مقدار ۰.۸۳ برای مدل پیشنهادی موفقیت این مدل را به نمایش گذاشته است. معیار بعدی MMRE یا میانگین خطای نسبی می‌باشد که با نمایش مقدار ۰/۱۳ کمترین خطای نسبی را دارد و در نهایت معیار MdMRE نیز با نمایش مقدار ۰/۱۱ برای مدل پیشنهادی دقت بالای مدل را به اثبات می‌رساند.

جدول ۶- نتایج بدست آمده روی پایگاه داده دوم

مدل‌ها	PRED(25)	MMRE	MDMRE
Proposed	0.83	0.13	0.11
CART	0.74	0.33	0.15
SWR	0.16	1.69	1.57
MLR	0.4	0.52	0.37
ABE	0.51	1.9	0.6
Karner	0.78	0.2	0.17

معیار PRED جایگاه روش پیشنهادی در مقایسه با مدل‌های دیگر را نمایان می‌سازد. همان‌گونه که ملاحظه می‌شود میزان PRED در مدل پیشنهادی بیشترین مقدار در مقایسه با دیگر مدل‌ها می‌باشد. بعد از مدل پیشنهادی مدلKARNER بهترین مقدار در این پارامتر را دارد. لازم به ذکر است که مدل SWR با مقدار ۰/۱۶ بدترین عملکرد را در مقیاس PRED داشته است. با توجه به اهمیت معیار PRED در همین ابتدای کار مشخص می‌شود که مدل پیشنهادی عملکرد قابل قبولی داشته است.

درخصوص پارامتر MMRE، بعد از مدل پیشنهادی مدل KARNER با ثبت مقدار۰/۲ رتبه دوم را دارد و ABE با میزان ۱/۹ بدترین عملکرد را دارد. نهایتا در معیار MdMRE، روش پیشنهادی با میزان ۰/۱۱ بهترین عملکرد را دارد و CART در رتبه دوم قرار دارد و همچنین SWR با میزان ۱/۵۷ بدترین عملکرد را دارد که در این معیار نیز روش پیشنهادی عملکردی مناسب نسبت به سایر روش‌ها دارد. شکل (۴) هر سه معیار MMRE، MDMRE و PRED پایگاه داده دوم را به نمایش می‌گذارد که مقایسه وتحلیل عملکرد روش پیشنهادی را ساده‌تر می‌کند.



شکل ۴- نمودار مقایسه عملکرد روش‌ها در پایگاه داده دوم

در این بخش به‌منظور بررسی آماری تفاوت‌های موجود بین مدل‌های مختلف تخمین از یک تست اماری بر روی مقادیر MRE بدست آمده توسط هر مدل استفاده می‌شود. به‌منظور پیشبرد این آزمایش، روش آماری غیر پارامتریک Wilcoxon Signed-Rank در این مقاله استفاده شده است و سطح معنی داری ۰/۹۵ به‌منظور ارزیابی‌ها لحاظ گردیده است.

بدین معنا که در شرایطی که مقدار P-Value مربوط به مقایسه مقادیر MRE دو مدل کمتر از ۰/۰۵ باشد، وجود یک تفاوت معنادار بین نتایج اثبات می‌شود. جداول (۷) و (۸) مقادیر P-Value مربوط به مقایسه نتایج MRE مدل پیشنهادی با سایر مدل‌ها را برای پایگاه داده اول و دوم نمایش داده شده است.

جدول ۷- مقادیر P-Value مربوط به مقایسه MRE مدل پیشنهادی و سایر مدل‌ها در پایگاه داده اول

CART	SWR	MLR	ABE	Karner
0.03	0.00001	0.04	0.02	0.03

جدول ۸- مقادیر P-Value مربوط به مقایسه MRE مدل پیشنهادی و سایر مدلها در پایگاه داده دوم

CART	SWR	MLR	ABE	Karner
0.03	0.000005	0.02	0.003	0.04

نتایج نمایش داده شده در جداول (۷) و (۸) تفاوت معنادار نتایج را در همه مقایسه‌ها تایید می‌کند. بنابراین اصل متفاوت بودن عملکرد مدل‌های مورد مقایسه بر اساس معیار MRE اثبات می‌شود و فرضیه عدم وجود تفاوت معنادار در بین نتایج رد می‌شود.

۵. تحلیل و توجیه نتایج

از بررسی نتایج مختلف با سه معیار ارزیابی اصلی، پیشرفت چشمگیر روش پیشنهادی نسبت به بقیه‌ی روش‌ها نمایان است. دلیل اصلی آن نیز بهینه کردن وزن‌ها، انتخاب ویژگی‌های مناسب و تنظیم نتایج است که منجر به دقت بیشتر مدل پیشنهادی گردیده است. به عبارت دیگر دلیل بهبود پارامترهای عملکردی در مدل پیشنهادی وزن‌دهی مناسب به عوامل محیطی است که به‌صورت پویا و کارآمد انجام شده است و علاوه بر این افزایش میزان نقش سایز پروژه در تخمین تلاش از طریق تغییر رابطه محاسباتی تاثیر چشمگیری در بهبود نتایج داشته است. از سوی دیگر انتخاب ویژگی‌های موثر و تنظیم مقادیر نهایی بدست آمده نقش کلیدی در افزایش دقت نتایج داشته است. هر دو پایگاه داده مشابه و برگرفته از پروژه‌های نرم‌افزاری واقعی است که بوسیله مدل‌های مختلف تخمین مورد استفاده قرار گرفته‌اند. لذا می‌توان امیدوار بود که نتایج حاصل از مدل پیشنهادی قابل تعمیم به سایر پروژه‌ها نیز باشد و انعطاف‌پذیری مدل پیشنهادی برای بکارگیری آن در پروژه‌های واقعی قابل قبول باشد. در اینجا از پتانسیل زیاد الگوریتم‌های فراابتکاری جهت بهینه‌سازی استفاده شده و نتیجه به‌دست آمده نیز راضی کننده و قابل قبول است. می‌توان چنین نتیجه گرفت که تمرکز بر روی عوامل محیطی و تلاش برای بهینه‌سازی فرآیند وزن‌دهی به آنها می‌تواند تأثیر چشمگیری در بهبود عملکرد روش نقطه مورد کاربری داشته باشد. خودکارسازی فرآیند وزن‌دهی می‌تواند خطای انسانی را تا حد زیادی کاهش دهد. در خصوص قابلیت اطمینان نتایج، با توجه به انجام تست‌های آماری لازم، تهدیدهای احتمالی برطرف گردیده و صحت اختلاف عملکرد مدل‌های مورد مقایسه به اثبات رسیده است. نهایتا در خصوص به‌کارگیری سایر الگوریتم‌های بهینه‌سازی باید گفت که سرعت و عملکرد مطلوب الگوریتم ازدحام ذرات دلیل انتخاب این الگوریتم بوده است. هرچند به‌کارگیری سایر الگوریتم‌ها نیز ممکن است منجر به بهبود بیشتری گردد و می‌توان این مهم را مورد توجه قرار داد. در پایان ذکر این نکته ضرروی است که عملکرد مطلوب روش پیشنهادی در دو پایگاه داده متفاوت قابل اعتماد آن را افزایش داده است. هرچند به‌منظور افزایش دامنه کاربرد آن، لازم است پایگاه داده‌های بیشتری مورد مطالعه و بررسی قرار گیرند که در حال حاضر در دسترس نویسندگان نبوده است.

۶. نتیجه‌گیری

آغاز نمودن یک پروژه بدون داشتن طرح و چشم‌اندازی از اتمام آن مانند شروع شکست یک پروژه است. تخمین تلاش در پروژه‌های نرم‌افزاری قبل از آغاز پروژه امری حیاتی است، چون نقش راه را برای تصمیم‌گیران در رابطه با تخصیص منابع به شکلی کارآمدتر ممکن می‌سازد. در پروژه‌های نرم‌افزاری به دلیل پیچیدگی ذاتی که در آن‌ها وجود دارد، عمل تخمین تلاش با مشکلات عدیده‌ای روبروست. روش تخمین مناسب امکان استفاده موثر از زمان و هزینه ساخت سیستم را فراهم می‌کند، به علاوه برآورد تلاش، شرکت‌ها را قادر می‌سازد که قبل از پیاده سازی یک نرم‌افزار مقدار

تلاش مورد نیاز، زمان و بودجه معین برای مدیریت پروژه را بهتر شناسایی کرده و فازهای مختلف را برنامه‌ریزی کنند.

پیشینه تحقیق نشان می‌دهد که در حوزه تخمین تلاش کمتر به پروژه‌های شی‌گرا توجه شده است و این در حالی است که این نوع پروژه‌ها بسیار مورد توجه توسعه دهندگان قرار دارند. در این مقاله تلاش شد تا با معرفی یک مدل ترکیبی جدید روش شی‌گرای UCP بهبود داده شود. در این فرآیند بهبود، موثرترین ویژگی‌های پروژه‌ها انتخاب شد و نحوه وزن دهی به متغیرهای محیطی برای نخستین بار تغییر کرد و مکانیزم به‌کارگیری سایز پروژه نیز تغییر داده شد. علاوه بر این یک مکانیزم تنظیم نتایج نیز در پیش گرفته شد. بمنظور خودکارسازی اعمال تغییرات از الگوریتم ازدحام ذرات استفاده شد و مدل پیشنهادی بر روی ۱۱۰ پروژه واقعی نرم‌افزاری در قالب دو پایگاه داده شبیه سازی شد. نتایج حاصل از مدل پیشنهادی با ۵ مدل رایج در این زمینه مورد مقایسه قرار گرفته و برتری آن به اثبات رسید. علاوه بر این تفاوت معنادار نتایج مدل‌های مختلف از طریق پیشبرد یک آزمون آماری اثبات گردید. نهایتا این‌گونه می‌توان نتیجه گرفت که وزن‌دهی پویا و هدفمند عوامل محیطی در روش UCP و همین‌طور تغییر نحوه به‌کارگیری پارامتر سایز پروژه از طریق الگوریتم‌های فراابتکاری و البته انتخاب موثر ویژگی‌ها و تنظیم نتایج در کنار هم می‌تواند تاثیر بسیاری در عملکرد این روش داشته باشد. به‌دلیل استفاده از داده‌های پروژه‌های واقعی و همین‌طور به‌کارگیری دو پایگاه داده متفاوت تعمیم نتایج مدل پیشنهادی و انعطاف پذیری آن قابل اثبات است. البته در کارهای آینده لازم است جامعه آماری بزرگتری بررسی شده و همین‌طور تاثیر به‌کارگیری سایر الگوریتم‌های بهینه‌سازی نیز مورد مطالعه قرار گیرد.

۷. مراجع

[1] Shepperd, M., (2014), Cost prediction and software project management, in software project management in a changing world, G. Ruhe and C. Wohlin, Editors., Springer Berlin Heidelberg: Berlin, Heidelberg, pp. 51-71.

[2] Løhre, E. and M. Jørgensen, (2016), Numerical anchors and their strong effects on software development effort estimates. Journal of Systems and Software, Vol. 116, pp. 49-56.

[3] Standish Group, CHAOS manifesto, (2015), West Yarmouth, Massachusetts .

[4] Bansal, A.J.i.R.G.A., (2016), Ranking of software effort estimation selection criteria based on fuzzy set theory, International Journal of Advance Research and Innovation, Vol. 4, pp. 43-47.

[5] Tanveer, B., L. Guzmán, and U.M. Engel, (2017), Effort estimation in agile software development: Case study and improvement framework, Journal of Software: Evolution and Process, doi: 10.1002/smr.1862

[6] Phannachitta, P., J. Keung, A. Monden, and K. Matsumoto, (2017), A stability assessment of solution adaptation techniques for analogy-based software effort estimation. Empirical Software Engineering, Vol.22, pp. 474-504.

[7] Di Martino, S., F. Ferrucci, C. Gravino, and F. Sarro (2016), Web effort estimation: function point analysis vs. cosmic. Information and Software Technology Vol.72, pp. 90-109.

[8] Verner, J. and G, Tate., (1992), A Software Size Model, Software Eng., IEEE, Vol.18 , pp. 265-278 .

[9] Albrecht, A. and J.E., Gaftney, (1983), Software function, source lines of code, and development effort prediction: a software science validation, Transactions on Software Engineering, IEEE, Vol.9, pp.639-948.

[10] Putnam, L.H., (1978), A general empirical solution to the macro software sizing and estimation problem, Transactions on Software Engineering, IEEE, Vol.62, pp. 345-361.

[11] Putnam L.H. and Myers, (1992), Measures for Excellence, Yourdon Press Computing Series.

[12] Karner, G., (1993), Resource estimation for object oriented projects, Objective Systems SF AB .

[13] Abualkishik, A.Z., et al., (2017), A study on the statistical convertibility of IFPUG Function Point, COSMIC Function Point and Simple Function Point, Information and Software Technology, Vol. 86, pp. 1-19.

[14] Boehm, B.W., (1981), Software Engineering Economics, Prentice Hall, Springer.

[15] Boehm, B.W., (1999), An overview of the cocomo 2.0 software cost model, IEEE Software Technology Conference.

[16] Samareh Moosavi, S.H. and V. Khatibi Bardsiri, (2017), Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation. Engineering Applications of Artificial Intelligence, Vol. 60, pp. 1-15.

[17] Kumar, G. and P.K. Bhatia, (2016), Empirical assessment and optimization of software cost estimation using soft computing techniques, in Advanced Computing and Communication Technologies: Proceedings of the 9th ICACCT, 2015, R.K. Choudhary, et al., Editors. Springer Singapore: Singapore. pp. 117-130.

[18] Sehra, S.K., Y.S. Brar, N. Kaur, and S.S. Sehra, (2017), Research patterns and trends in software effort estimation. Information and Software Technology, doi.org/10.1016/j.infsof.2017.06.002

[19] Kazemifard, M., A. Zaeri, N. Ghasem-Aghaee, M.A. Nematbakhsh, and F. Mardukhi, (2017), Fuzzy Emotional COCOMO II Software Cost Estimation (FECSCE) using Multi-Agent Systems. Appl. Soft Comput., Vol.11, pp. 2260-2270.

[20] Park, B.K., S.Y. Moon, and R.Y.C. Kim, (2016), Improving Use Case Point (UCP) Based on Function Point (FP) Mechanism, , International Conference on Platform Technology and Service (PlatCon)

[21] Sarwosri, M.J.A. Haiyan, M. Husein, and A.P. Ferza, (2016), The development of method of the enhancement of Technical Factor (TF) and Environmental Factor (EF) to the Use Case Point (UCP) to calculate the estimation of software's effort, International Conference on Information & Communication Technology and Systems (ICTS).

[22] Silhavy, R., P. Silhavy, and Z. Prokopova, (2017), Analysis and selection of a regression model for the Use Case Points method using a stepwise approach, Journal of Systems and Software, Vol. 125, pp. 1-14.
[23] Iskandar, K., F.L. Gaol, B. Soewito, H.L.H.S. Warnars, and R. Kosala, (2016), Software size measurement of knowledge management portal with use case point, International Conference on Computer, Control, Informatics and its Applications (IC3INA).
[24] Kennedy, J and Eberhart, R., (2001), Swarm Intelligence, Academic Press, Is ted., San Diego, CA .
[25] Carroll, E.R. (2005), Estimating software based on use case points. Companion to the 20th annual Conference on Object-oriented programming, systems, languages, and applications, ACM SIGPLAN, pp.257-265.
[26] Khatibi, E. and V.K. Bardsiri, (2015), Model to estimate the software development effort based on in-depth analysis of project attributes. IET Software, Vol.9, pp.109-118.
[27] Nassif. A. B, D. Ho and L. F. Capretz, (2011), Regression model for software effort estimation based on the use case point method, International Conference on Computer and Software Modeling, Singapore, pp. 117-121 .
[28] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J. (1984), Classification and regression trees, Chapman and Hall, 1-368
[29] Nagpal1, G., Uddin, M., Kaur, A., (2014), Grey relational effort analysis technique using regression methods for software estimation, The International Arab Journal of Information Technology, Vol.11, pp. 437-446.
[30] Fedotova, O., Teixeira, L., Alvelos, H., (2013), Software effort estimation with multiple linear regression: Review and practical application, Journal of Information Science and Engineering, Vol.29, pp. 925-945.
[31] Khatibi Bardsiri, V., Jawawi, A.D.N., Mohd Hashim, S.Z., Khatibi, E., (2012), A pso-based model to increase the accuracy of software development effort estimation, Software Qual J, Vol.21, pp. 501-526.