

ارائه روشی جدید مبتنی بر مدل کوکومو بمنظور افزایش دقت تخمین تلاش در پروژه های نرم افزاری

مهديه سالاری* و حید خطیبی بردسیری** عمید خطیبی بردسیری**

*گروه علمی مهندسی کامپیوتر، موسسه آموزش عالی غیرانتفاعی غیر دولتی جاوید

**استادیار گروه مهندسی کامپیوتر، دانشگاه آزاد اسلامی، واحد بردسیر

تاریخ پذیرش: ۱۳۹۹/۰۹/۲۱

تاریخ دریافت: ۱۳۹۹/۰۱/۰۵

نوع مقاله: پژوهشی

چکیده

تخمین و برآورد معیارها یک فعالیت حیاتی در پروژه‌های نرم‌افزاری محسوب می‌شود. به طوری که تخمین تلاش در مراحل اولیه توسعه نرم‌افزار، یکی از مهم‌ترین چالش‌های مدیریت پروژه‌های نرم‌افزاری است. تخمین نادرست می‌تواند منجر به شکست پروژه گردد. لذا یکی از فعالیت‌های اصلی و کلیدی در توسعه مؤثر و کارآمد پروژه‌های نرم‌افزاری تخمین دقیق هزینه‌های نرم‌افزار است. از این رو در این پژوهش دو روش به منظور تخمین تلاش در پروژه‌های نرم‌افزاری ارائه شده است، که در این روش ها سعی شده با تجزیه و تحلیل محرک‌ها و استفاده از الگوریتم‌های فرا ابتکاری و ترکیب با شبکه عصبی راهی برای افزایش دقت در تخمین تلاش پروژه های نرم افزاری ایجاد شود. روش اول تأثیر الگوریتم فاخته جهت بهینه‌سازی ضرایب تخمین مدل کوکومو و روش دوم به صورت ترکیبی از شبکه عصبی و الگوریتم بهینه‌سازی فاخته جهت افزایش دقت برآورد تلاش توسعه نرم‌افزار ارائه شده است. نتایج بدست آمده روی دو پایگاه داده واقعی نشان دهنده عملکرد مطلوب روش ارائه شده در مقایسه با سایر روشهاست.

واژگان کلیدی: الگوریتم فاخته، تخمین هزینه، شبکه عصبی، کوکومو.

۱- مقدمه

یکی از روش‌های تخمین هزینه‌های نرم‌افزار، پیشگویی میزان تلاش لازم برای ساخت نرم‌افزار و زمان موردنیاز جهت توسعه آن است؛ که به شکل ساده می‌توان تلاش لازم را برحسب تعداد افرادی که در روز/هفته/ماه و یا حتی سال بر روی پروژه کار می‌کنند را تخمین زد [۱]. مدل‌های تخمین زمان و هزینه نرم‌افزار را به دودسته الگوریتمی و غیر الگوریتمی تقسیم می‌کنند [۲-۴]. مدل‌های الگوریتمی مدلهایی هستند که بر مدل‌های ریاضی استوارند و از نظر محاسبات ریاضی پیچیده می‌باشند. این مدل‌ها تلاش می‌کنند تا

تخمین زمان و هزینه نرم‌افزاری از فعالیت‌های علمی موردنیاز در زمینه مدیریت پروژه‌های نرم‌افزاری محسوب می‌شود. دقت تخمین نرم‌افزار دارای اهمیت بالایی است، زیرا می‌تواند به طبقه‌بندی و اولویت‌بندی پروژه به منظور معین کردن منابعی که در آینده موردنیاز پروژه خواهند بود، کمک به سزایی کند.

نویسنده مسئول: مهديه سالاری @mahdieh.salari60@gmail.com

مدل تخمین کوکومو نتایج دقیق تری در مقایسه با مدل کوکوموی اولیه با دو معیار عملکرد متوسط شدت خطای نسبی ۱/۲۶ و درصد پیش‌بینی ۱/۵۴ را ایجاد نموده است. همچنین این مدل در مقایسه با شبکه عصبی مصنوعی، شبکه عصبی فازی^۳ و فازی تجزیه و تحلیل رابطه‌ای گری^۴ نتایج بهتری را ارائه داده است [۶]. در مقاله [۷] نسخه جدیدی از مدل کوکومو با تکیه بر انتخاب محرک‌های هزینه و بهینه‌سازی ضرایب با استفاده از الگوریتم کلونی زنبور پرداخته است. در روش ارائه شده ضرایب کارآمدتری نسبت به حالت اولیه مدل کوکومو به دست آمده و انتخاب بهترین ضرایب بالاترین هدف این روش می‌باشد و نتایج شبیه‌سازی نمایانگر این برتری می‌باشد. در مقاله دیگری از الگوریتم رقابت استعماری برای تخمین هزینه نرم‌افزار استفاده شده است که روی پایگاه داده کوکومو ۸۱ صورت گرفت. نتایج تجربی نشان می‌دهند که در حالت کوکوموی مجزا، الگوریتم رقابت استعماری طوری پارامترهای مدل کوکومو ۸۱ را مقداردهی می‌کند که MMRE برابر با ۰,۲۸۶۳ و PRED(۰,۲۵) برابر با ۰,۳۵۷۱ می‌گردد. واضح است که در مقایسه با مدل کوکومو ۸۱ که MMRE برابر با ۰,۳۱۸۰ و PRED(۰,۲۵) برابر با ۰,۳۴۹۲ می‌باشد، الگوریتم رقابت استعماری توانسته است، با دقت بیشتری عملیات تخمین هزینه را انجام دهد [۸]. در یک مقاله، ACOT برای پیش‌بینی برآورد هزینه نرم‌افزار با استفاده از داده‌های تخمین تلاش مختلف، استفاده می‌کند. در طول مطالعه، اعتبار سنجی متقابل ده برابر را در مجموعه داده‌های ISBSG و اعتبار سنجی متقابل سه‌گانه در مجموعه داده‌های IBMDP و COCOMO ۸۱ انجام شده است؛ و عملکردهای ACOT را با SVR، GP، TREE NET، MLP، DENFIS، MARS و PSO را مقایسه کردند نتایج تجربی با توجه به ارزش RMSE^۵ بهتر از سایر تکنیک‌های مستقل بوده و از این رو، نتیجه می‌گیریم که پس از آزمایش‌های گسترده‌ای که مدل ACOT نسبتاً بهترین پیش‌بینی کننده در میان تمام تکنیک‌های دیگر است [۹]. در مقاله ی دیگر الگوریتم ژنتیک برای انتخاب ویژگی در زمینه تخمین تلاش‌های پروژه نرم‌افزار مورد استفاده قرار گرفته است این تکنیک بر روی مجموعه داده‌های شناخته شده ماکسول، کوکومو ۸۱ و مجموعه داده Desharnais آزمایش شده است. نتایج پیاده‌سازی نشان می‌دهد که زیرمجموعه حاصل،

ارتباطی بین تلاش و یک یا دو مشخصه پروژه برقرار کنند. شکل کلی این مدل‌ها به صورت رگرسیون خطی یا غیرخطی می‌باشد. در این گونه مدل‌ها میزان کار مورد نیاز برای انجام پروژه به صورت تابعی از متغیرهای متعدد تعریف می‌شود این متغیرها در واقع فاکتورها یا عوامل هزینه هستند که به صورت خطی^۱، ضربی و نمایی در محاسبه زمان و هزینه تأثیر می‌گذارند. معادله تخمین استفاده شده به شکل زیر است.

$$\text{Effort} = f(x_1, x_2, \dots, x_n) \quad (1)$$

که در آن x_1 تا x_n بردار فاکتورهای هزینه است. از سوی دیگر به دلیل عدم توانایی روش‌های الگوریتمی در مهار کردن رفتار پویای پروژه‌های نرم‌افزاری و نبود اطلاعات کامل از یک پروژه در مراحل اولیه، متدهای غیر الگوریتمی ارائه شدند. که این متدها بر اساس تجزیه و تحلیل پروژه‌های نرم‌افزاری بنا شده‌اند. در این مقاله با تمرکز بر روش کوکومو به عنوان متد پایه و الگوریتم بهینه‌سازی فاخته و شبکه عصبی به عنوان ابزار کمکی، یک مدل پویا برای تخمین مقدار تلاش ارائه خواهیم کرد. ادامه این مقاله در پنج قسمت سازمان‌دهی شده است: بخش دوم به مرور کارهای مرتبط می‌پردازد. در بخش سه به ترتیب مدل کوکومو، الگوریتم فاخته و شبکه عصبی و روش ارائه شده آورده شده و بخش چهارم نیز به ترتیب نحوه ارزیابی و نتایج به دست آمده را نشان می‌دهند. در نهایت بخش پنجم به نتیجه‌گیری و پیشنهاد کارهای آینده اختصاص دارد.

۲- پیشنهاد پژوهش

جهت تخمین تلاش نرم‌افزار، مدل‌های مختلفی معرفی شده است که کوکومو^۲ یکی از مشهورترین آن‌ها می‌باشد. کوکوموی ۱ یا کوکوموی ۸۱ یک مدل مبتنی بر رگرسیون می‌باشد که در سال ۱۹۸۱ توسط بوهم پیشنهاد شد [۵]. این مدل جزو دسته مدل‌های محسوب می‌شود که می‌تواند علاوه بر میزان تلاش، میزان زمان‌بندی یک پروژه نرم‌افزاری را نیز محاسبه کند. در مدل تخمین تلاش ترکیبی، بر پایه ترکیب تکنیک‌های هوش محاسباتی، مدل کوکومو و الگوریتم بهینه‌سازی ازدحام ذرات نتایج آزمایش نشان داد که ترکیب الگوریتم‌های خوشه‌بندی با الگوریتم بهینه‌سازی ازدحام ذرات در ارزیابی

^۴ Fuzzy Gray Relational Analysis (FGRA)

^۵ Root Mean Square Error

^۱ Linear Models

^۲ Cocomo

^۳ Fuzzy Neural Network (FNN)

Archive of SID

بینی) بهبود یافته است. در مقاله [۱۵] یک مدل ترکیبی از الگوریتم جستجوی کوکو و الگوریتم جستجوی هارمونی برای بهینه‌سازی چهار ضریب مدل کوکومو^۲ برای برآورد مطلوب پیشنهاد شده است. رویکرد پیشنهادی در مجموعه داده‌های ناسا ۹۳ و با استفاده از مقادیر خطای نسبی [۱۶] و اندازه خطای نسبی (MMRE) مورد ارزیابی قرار گرفته است. نتایج تجربی نشان می‌دهد که روش ارائه شده در برآورد تلاش و زمان توسعه پروژه نرم افزاری نسبت به کوکومو^۲ و الگوریتم جستجوی کوکو موثرتر است. امروزه برآورد هزینه نرم افزار (SCE) با تکنیک‌های یادگیری ماشین عملکرد بیشتری نسبت به سایر تکنیک‌های سنتی دارند در این مقاله، یک مدل ترکیبی جدید از چندین مدل ارائه شده که مدل‌ها عبارت‌اند از چندلایه پرسپترون (MLP)، شبکه عصبی مصنوعی (ANN) و بهینه‌سازی کلونی مورچه^۲ (ACO) و برای بالا بردن دقت در برآورد هزینه نرم افزار به نام بهینه‌سازی کلونی مورچه چندلایه پرسپترون (MLPACO) نامیده می‌شود. در این مقاله با استفاده از الگوریتم کلونی مورچه (ACO) از بین ویژگی‌های پروژه‌های نرم افزاری ویژگی‌هایی که باعث افزایش دقت تخمین می‌شود فیلتر می‌شوند و با استفاده از شبکه عصبی چندلایه پرسپترون بهینه‌سازی تخمین انجام می‌شود. نتایج نشان می‌دهد که در این رویکرد جدید افزایش دقت برای بیش از ۸۰٪ موارد بهتر از الگوریتم سازنده هزینه کوکومو می‌باشد همچنین نتایج الگوریتم پیشنهاد شده نشان می‌دهد که میانگین مقدار خطای نسبی (MMRE) در الگوریتم پیشنهاد شده است پایین‌تر از مدل کوکومو می‌باشد [۱۷]. در پژوهشی دیگر رویکردی پیشنهاد شد که شامل شبکه عصبی مصنوعی عملکردی و الگوریتم کلونی زنبور عسل مصنوعی^۳ که به عنوان الگوریتم آموزش شبکه و برای ارائه دقیق‌ترین تخمین هزینه نرم افزار استفاده شده است. FLANN پیچیدگی محاسباتی را در شبکه عصبی چندلایه کاهش می‌دهد و لایه پنهان ندارد و بنابراین یادگیری سریع‌تری دارد؛ و پس از ارزیابی نتایج نشان داده شده که آموزش FLANN با الگوریتم ABC برای رفع مسئله هزینه نرم افزار بسیار بهبود یافته است و علاوه بر این مدل به دلیل نداشتن لایه پنهان و محاسبات کمتر در طول آموزش از لحاظ ساختاری بسیار ساده می‌باشد؛ و یکی از اهداف مهم این تحقیق این است که آموزش FLANN با یک

در مقایسه با مجموعه داده‌های اصلی، نتیجه‌های بهتر را از لحاظ دقت برآورد تلاش به ارمغان آورده است. این مقاله نشان داد که الگوریتم‌های ژنتیک روش‌های ایده آل برای انتخاب یک زیرمجموعه از ویژگی‌ها و بهبود دقت تخمینی تلاش هستند [۱۰]. یکی از مشهورترین روش‌های تخمین تلاش، استفاده از قیاس و مقایسه یک سرویس با موارد مشابه قبلی است. متأسفانه روش قیاس بدون استفاده از وزن‌های مناسب و ارزشدهی به ویژگی‌های یک سرویس، نتایج خوبی نخواهد داشت. بنابراین در این مقاله سعی شده تا با ترکیب روش قیاس و الگوریتم تکامل تفاضلی یک مدل کارا و قابل اطمینان برای برآورد تلاش لازم جهت توسعه سرویس‌های نرم افزاری ایجاد شود. روش ارائه شده بر روی داده‌های واقعی مستخرج از پایگاه داده ISBSG و دو پایگاه داده مصنوعی مورد ارزیابی قرار گرفت و نتایج با روش‌های مشهور تخمین تلاش مقایسه گردید؛ مقادیر به دست آمده برای مخازن داده‌های ISBSG^۱، همگن و ناهمگن به ترتیب و به طور میانگین بهبود ترکیبی برای تخمینی تلاش‌های پروژه‌های نرم افزاری پیشنهاد شده است. روش ارائه شده ترکیبی از الگوریتم بهینه سازی ذرات و یک روش رگرسیون خطی است که در آن نتایج ضرایب بهینه انجام می‌شود. علاوه بر این، معادله برآورد با استفاده از اندازه پروژه اندازه گیری می‌شود به طوری که دقیق ترین برآورد به دست آمده است. مجموعه داده ای نسبتاً واقعی برای ارزیابی عملکرد مدل پیشنهاد شده و نتایج با مدل های دیگر مقایسه می‌شود. نتایج به دست آمده نشان داد که مدل هیبرید پیشنهادی می‌تواند صحت برآورد ها را بهبود بخشد. در مقاله [۱۳] یک روش جدید بر اساس ترکیبی از مدل های برآورد مختلف و ریاضی پاسخ خود را ارائه شده که این روش جدید در دو داده مجموعه کاملاً متفاوت واقعی مورد بررسی قرار گرفت. در مقاله [۱۴] یک مدل جدید برای برآورد هزینه های نرم افزاری ارائه می‌دهد که با استفاده از الگوریتم جستجوی کوکو (CS) برای پیدا کردن پارامتر مطلوب مدل تخمینی هزینه است. روش ارائه شده در مجموعه داده های پروژه نرم افزار ناسا مورد آزمایش قرار گرفته است. نتایج تجربی نشان می‌دهد که روش ارائه شده عملکرد نتایج برآورد شده با توجه به MMRE (میانگین خطای نسبی) و PRED (پیش

^۲ Bee Colony optimization

^۱ International Software Benchmarking Standards Group

^۳ Ant Colony Optimization

وزن المان ها توسط الگوریتم رقابته امپریالیست بهینه می شوند. برای مقایسه روش ارائه شده برای برآورد تلاش با مدل های پیشرفته ، سه مجموعه داده به عنوان معیار ، یعنی COCOMO ، Maxwell و Albrecht انتخاب شده اند. مجموعه داده ها استاندارد و در دسترس عموم برای ارزیابی هستند. آزمایش ها نتایج امیدوار کننده را نشان می دهد و عملکرد متوسط با الگوی پیشنهادی معیار عملکرد MMRE در مجموعه داده ها به ترتیب ۲۳٪ ، ۳۸٪ و ۳۵٪ بهبود یافته است.

۳- روش شناسی پژوهش

۳-۱- کوکومو

واژه کوکومو به طور اختصار از دو حرف اول کلمات جمله‌ی *Constructive Cost Modle* که به معنای مدل هزینه سودمند است گرفته شده است. این مدل ابتدا در سال ۱۹۷۰ توسط بری بوهم ارائه و بعداً در سال ۱۹۸۱ در کتاب ایشان به نام اقتصاد مهندسی نرم افزار چاپ و در اختیار عموم قرار گرفت [۲۲، ۲۳] این مدل در سه سطح پایه، متوسط، با جزئیات ارائه شده است که در ادامه به تحلیل هر یک از این ۳ سطح می پردازیم.

• کوکومو پایه^۱

کوکومو پایه میزان تلاش^۲ (و هزینه) مورد نیاز جهت توسعه نرم افزار را به صورت تابعی از اندازه‌ی برنامه محاسبه می کند. اندازه برنامه برآوردی از تعداد خطوط برنامه بر پایه ۱۰۰۰ است که ان را به اختصار *KLOC* و یا *SLOC* می نامند.

معادله کوکومو پایه به طور کلی:

تلاش کاربردی (بر اساس فرد در ماه)

$$Effort Applied (E) = a (KLOC)b \quad (2)$$

• کوکومو متوسط

کوکومو متوسط میزان تلاش (و هزینه) مورد نیاز جهت توسعه نرم افزار را به صورت تابعی از اندازه‌ی برنامه و مجموعه‌ای از

برنامه آموزش بهبود یافته جایگزین الگوریتم یادگیری استاندارد BP برای ارزیابی هزینه دقیق توسعه نرم افزار شده است و مزیت عمده روش ارائه شده این است که پیچیدگی محاسباتی را بدون هیچ گونه نقص در عملکرد آن برطرف می کند. نتایج به دست آمده از روش پیشنهادی با استفاده از معیارهایی مانند MRE ، MMRE و MdMRE ارزیابی شده است با توجه به نتایج به دست آمده می توان گفت که FLANN دقت تخمین هزینه را افزایش می دهد و به ویژه هنگامی که این الگوریتم با الگوریتم ABC آموزش داده می شود دقت هزینه برآورد شده بسیار بهتر می شود [۱۸]. در مقاله [۱۹]، اجرای مدل سازی غیر الگوریتمی با استفاده از تکنیک های محاسبات نرم مانند منطق فازی و الگوریتم های ژنتیکی انجام می شود. روش فازی برای طراحی یک مدل فازی برای هر درایور هزینه پیاده سازی شده است. مدل فازی دارای تعریف نادرست و مبهم از محدوده ورودی درایورهای هزینه است. انتخاب پارامترهای مشخص کننده مجموعه های فازی در مدل فازی پیشنهادی با استفاده از الگوریتم های ژنتیکی بیشتر بهینه می شود. مدل ارائه شده بر روی مجموعه داده های COCOMO NASA و مجموعه داده COCOMO NASA۲ با استفاده از MATLAB آزمایش شده است. بهبود عملکرد مدل بهینه سازی شده پیشنهادی از نظر میانگین بزرگی خطای نسبی (MMRE) و Pred (۲۵٪) اندازه گیری می شود. در مقاله [۲۰] شبکه عصبی مصنوعی با استفاده از الگوریتم بهینه سازی فاخته برای پیش بینی برآورد هزینه نرم افزار آموزش داده شده است. هدف اصلی در این مقاله استفاده از یک روش یادگیری جدید در شبکه عصبی برای پیش بینی بهتر تخمین پروژه های نرم افزاری می باشد. روش ارائه شده با مجموعه داده ISBSG تأیید شده و نتایج با مدل های موجود مقایسه شده است. نتایج نشان داده شده از نظر خطای میانگین مربع (RMSE) و اندازه متوسط خطای نسبی (MMRE) است. در مقاله [۲۱]، یک رویکرد ترکیبی سه فاز برای غلبه بر مشکل تخمین هزینه ارائه شده است. در مرحله اول ، ویژگی ها با استفاده از ترکیبی از الگوریتم ژنتیکی و شبکه عصبی پرسپترون انتخاب می شوند. در مرحله دوم ، عوامل ضربه با استفاده از روش های رگرسیون خطی چندگانه که به عنوان ضرایب تأثیر برای هر ویژگی عمل می کنند ، به هر ویژگی انتخابی مرتبط می شوند. در مرحله آخر و سوم ،

^۲ Effort

^۱ Basic COCOMO

تخم‌ها نجات یابند پارامتری خواهد بود که COA قصد بهینه‌سازی آن را دارد [۲۵].

فاخته‌ها برای بیشینه کردن نجات تخم‌های خود به دنبال بهترین منطقه می‌گردند. پس از آنکه جوجه‌ها از تخم درآمدند و به فاخته بالغ تبدیل شدند، جوامع و گروه‌هایی تشکیل می‌دهند. هر گروه منطقه سکونت خود را برای زیست دارد. بهترین منطقه سکونت تمام گروه‌ها مقصد بعدی فاخته‌ها در سایر گروه‌ها خواهد بود. تمام گروه‌ها به سمت بهترین منطقه موجود فعلی مهاجرت می‌کنند. هر گروه در منطقه‌ای نزدیک بهترین موقعیت فعلی ساکن می‌شود. با در نظر گرفتن تعداد تخمی که هر فاخته خواهد گذاشت و همچنین فاصله فاخته‌ها از منطقه بهینه فعلی برای سکونت تعدادی شعاع تخم‌گذاری محاسبه شده و شکل می‌گیرد. سپس فاخته‌ها شروع به تخم‌گذاری تصادفی در لانه‌هایی داخلی شعاع تخم‌گذاری خود می‌کنند. این پروسه تا رسیدن به بهترین محلی برای تخم‌گذاری (منطقه با بیشترین سود) ادامه می‌یابد. این محلی بهینه جایی است که بیشترین تعداد فاخته‌ها در آن گرد می‌آیند.

حل یک مسئله بهینه‌سازی لازم است تا مقادیر متغیرهای مسئله به فرم یک آرایه شکل گیرند. در GA و PSO این آرایه‌ها بانام‌های «کروموزوم» و «موقعیت ذرات» مشخص می‌شوند. ولی در الگوریتم بهینه‌سازی فاخته این آرایه $habitat$ یا «محل سکونت» نام دارند.

در یک مسئله بهینه‌سازی $Nvar$ بعدی یک $habitat$ یک آرایه $Nvar \times 1$ خواهد بود که موقعیت فعلی زندگی فاخته‌ها را نشان می‌دهد. این آرایه به شکل زیر تعریف می‌شود:

$$Habitat = [x_1, x_2, \dots, xvar] \quad (4)$$

میزان مناسب بودن (مقدار سود) در $habitat$ فعلی با ارزیابی تابع سود (fp) در $habitat$ به دست می‌آید؛ بنابراین:

$$Profit = fp(habitat) = f(x_1, x_2, \dots, xvar) \quad (5)$$

همان‌طور که دیده می‌شود COA الگوریتمی است که تابع سود را ماکزیمم می‌کند. برای استفاده از COA برای حل مسائل کمینه‌سازی کافی است یک علامت منفی در تابع هزینه ضرب

محرک‌ها که شامل ارزیابی محصول، سخت‌افزار، کارکنان و ویژگی‌های پروژه هستند را محاسبه می‌کند. این محرک‌ها هر یک دارای چند سطح با مقادیر مخصوص به خود هستند.

هر یک از این ۱۵ ویژگی، از خیلی کم تا بسیار زیاد در ۶ سطح امتیازدهی می‌شوند (بر اساس ارزش و یا درجه اهمیت). با استفاده از افزایشده‌ها می‌توان میزان مقیاس‌ها و اندازه‌ها را در جدول زیر تغییر داد. محصول همه‌ی این افزایشده‌ها منجر به فاکتور تعدیل و تنظیم تلاش می‌شود (EAF). ارزش‌های به‌دست‌آمده معمولاً مقداری بین ۰٫۹ تا ۱٫۴ می‌باشند.

فرمول محاسبه تلاش کوکومو متوسط

$$E = ai(KloC)(bi)(EAF) \quad (3)$$

برای محاسبه زمان توسعه و افراد موردنیاز می‌توان از همان فرمول‌های کوکومو پایه استفاده کرد.

• کوکومو با جزئیات

این مدل از کوکومو دارای تمامی ویژگی‌های مدل کوکومو متوسط به‌اضافه یک ارزیابی از تأثیر محرک‌های هزینه در هر مرحله از فرایند مهندسی نرم‌افزار است.

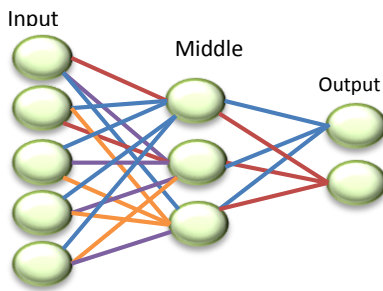
کوکومو با جزئیات میزان تلاش [۲۴] موردنیاز جهت توسعه نرم‌افزار را به‌صورت تابعی از اندازه‌ی برنامه و مجموعه‌ای از محرک‌ها بسته به چرخه حیات نرم‌افزار به دست می‌آورد.

۳-۲- الگوریتم فاخته

الگوریتم بهینه‌سازی فاخته^۱ (COA) همانند سایر الگوریتم‌های تکاملی هم با یک جمعیت اولیه کار خود را شروع می‌کند. جمعیتی متشکل از فاخته‌ها. این جمعیت از فاخته‌ها تعدادی تخم‌دارند که آن‌ها را در لانه تعدادی پرندگی میزبان خواهند گذاشت. تعدادی از این تخم‌ها که شباهت بیشتری به تخم‌های پرندگی میزبان دارند شانس بیشتری برای رشد و تبدیل شدن به فاخته بالغ خواهند داشت. سایر تخم‌ها توسط پرندگی میزبان شناسایی شده و از بین می‌روند. میزان تخم‌های رشد کرده مناسب بودن لانه‌های آن منطقه را نشان می‌دهند. هرچه تخم‌های بیشتری در یک ناحیه قادر به زیست باشند و نجات یابند به همان اندازه سود (تمایل) بیشتری به آن منطقه اختصاص می‌یابد؛ بنابراین موقعیتی که در آن بیشترین تعداد

^۱ Cuckoo optimization algorithm

است. بخش اول تابع ترکیب نام دارد که تمام ورودی ها را جمع می کند و نتیجه را به صورت یک مجموع وزن دار نشان می دهد. طبق این شکل برای هر نرون i تمام ورودی های I_j توسط فاکتور W_j وزن دار می شوند. سپس هر ورودی در وزن مربوطه ضرب شده و حاصل ضرب ها باهم جمع شده تا مجموع وزن دار u_i را نتیجه دهند:



شکل ۱. سلول ولایه های یک شبکه عصبی

شکل (۱). مدل ساده یک شبکه عصبی مصنوعی که از یک لایه ورودی، یک لایه میانی و یک لایه پنهان تشکیل شده است. شکل (۱) مدل ریاضی از فرایند پردازش در یک نرون.

$$u_i = \sum_{j=1}^i w_j I_j \quad (8)$$

بخش دوم تابع انتقال یا تابع تحریک نام دارد. وقتی مجموع وزندار به حد آستانه های رسید، تابع انتقال، تحریک شده تا خروجی را نتیجه دهد. توابع انتقال انواع مختلفی دارند که برخی از آنها عبارتند از: توابع انتقال *Tansig*, *Pureline* و *Logsig*.

معمولاً در یک سلول شبکه عصبی یک ورودی اضافی نیز وجود دارد که بایاس نامیده می شود. نقش بایاس افزایش یا کاهش مجموع وزن دار است [۲۶] بایاس به عنوان یک جبران کننده عمل می کند و به شبکه کمک می کند تا الگوهای موجود را بهتر بشناسد. یادگیری در یک شبکه عصبی مصنوعی توسط تعدیل (کموزیاد کردن) وزن ها صورت می گیرد [۲۷] الگوریتمهای آموزشی به دودسته نظارتی و غیر نظارتی تقسیم بندی میشوند. الگوریتم پس انتشار خطا نوعی از یادگیری نظارتی است که طی آن ورودیها وارد شبکه میشوند. وقتی که خروجی مطلوب به دست آید، خطای

کنیم. دیگر عادت هر فاخته حقیقی این است که آن ها در یک دامنه مشخص تخم های خود را می گذارند که با آن حداکثر دامنه تخم گذاری ELR ^۱ گفته می شود. در یک مسئله بهینه سازی هر متغیر دارای حد بالا Var_{hi} و حد پایین Var_{low} است که هر ELR با استفاده از این حدود قابل تعریف خواهد بود. ELR متناسب است با تعداد کل تخم ها، تعداد تخم های فعلی فاخته و همچنین حد بالا و پایین متغیرهای مسئله؛ بنابراین ELR به صورت رابطه ۶ تعریف می شود:

$$ELR = \alpha \times \frac{\text{number of current cuckoo's eggs}}{\text{total number of eggs}} \times (Var_{hi} - Var_{low}) \quad (6)$$

با توجه به تعداد تخم هر فاخته یک ELR برای آن مشخص می شود و سپس تخم گذاری شروع می گردد. فرمول عملگر مهاجرت در الگوریتم بهینه سازی فاخته به صورت رابطه است [۲۵]:

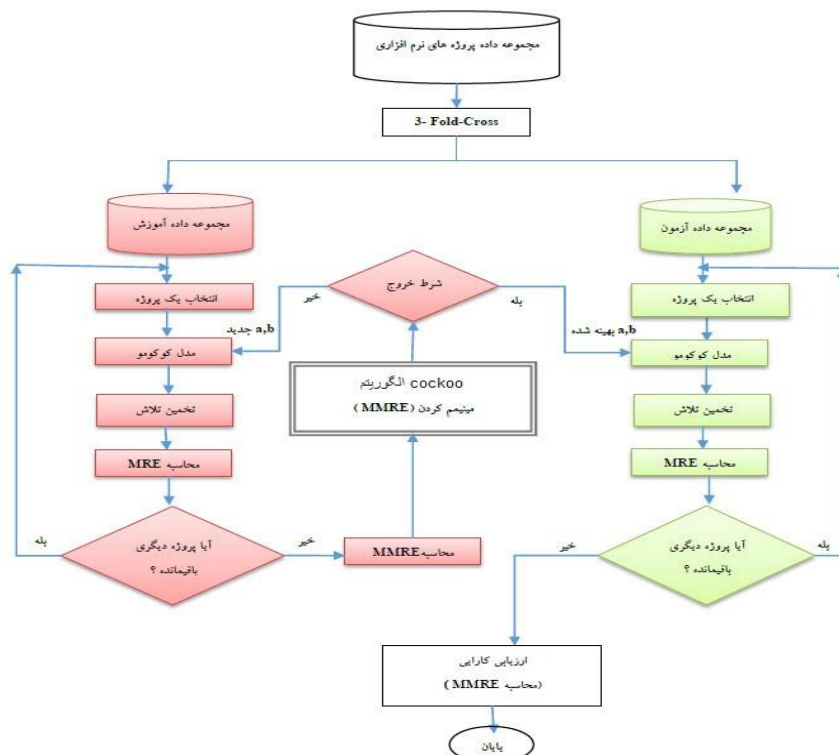
$$X_{NextHabitat} = X_{currentHabitat} + F(X_{GoalPoint} - X_{currentHabitat}) \quad (7)$$

F پارامتری است که باعث انحراف می شود.

۳-۳- شبکه عصبی

شبکه های عصبی یکی از کارآمدترین و گسترده ترین سیستم های هوشمند به شمار می آیند. شبکه های عصبی مصنوعی در واقع تقلید بسیار ساده ای از رفتار سلول های بیولوژیکی می باشد. اجزای تشکیل دهنده شبکه های عصبی مصنوعی شامل نرون ها و لایه ها می باشند. در حالت کلی یک شبکه عصبی از ۳ لایه تشکیل شده است شکل (۱): لایه ورودی که اطلاعات را دریافت می کند، لایه میانی یا پنهان که خود می تواند شامل چندین لایه باشد، یکی از مهم ترین پردازشگرهای اطلاعات است و لایه خروجی که این پردازش را دریافت کرده و خروجی نهایی را نتیجه می دهد. شکل (۱) مدل ساده ای از یک سلول شبکه عصبی را که نرون نامیده میشود نشان می دهد. بدنه این سلول از دو بخش تشکیل شده

^۱ Egg Laying Radius



شکل ۲. روش COA تخمین هزینه نرم افزار

بهترین مقادیر را برای ضرایب a و b یافته و در مدل کوکومو (پایه) جایگزین می‌نماید. در روش دوم (ANN-COA) الگوریتم فاخته از بین ۱۵ محرک هزینه مدل کوکومو محرک‌های مؤثرتر را پیشنهاد داده و به‌عنوان ورودی به همراه مقدار بهینه a, b به شبکه عصبی ارسال می‌کند و شبکه عصبی به‌عنوان خروجی تخمین تلاش نرم‌افزار را با توجه به محرک‌های مؤثر به دست می‌آورد.

۳-۴-۱ روش COA پیدا کردن ضرایب a و b بهینه برای مدل کوکومو با استفاده از الگوریتم‌های فرا ابتکاری

در این قسمت جزئیات مربوط به روش ارائه شده COA که در شکل (۲) نشان داده شده، بیان شده‌اند. این روش سعی بر آن دارد که تا حد امکان از طریق بهینه‌سازی ضرایب کوکومو بهترین تخمین‌ها را به دست آورد. فرایند بخش‌های روش ارائه شده به‌گونه‌ای است که میزان خطا در تخمین‌ها حداقل شود. این روش شامل ۲ بخش اصلی است که در ادامه به

محاسبه‌شده به‌صورت پس‌رو در شبکه منتشر میشود تا وزنها با توجه به آن تعدیل شوند. وقتی که خروجی محاسبه‌شده با خروجی واقعی تطابق خوبی داشته باشد، آموزش متوقف میشود. استفاده از شبکه عصبی در مهندسی هزینه و تخمین هزینه از سال ۱۹۹۳ شروع شده و نتایج مطلوبی را به همراه داشته است شبکه عصبی مصنوعی، معمول ترین روش مبتنی بر یادگیری مورد استفاده برای تخمین تلاش توسعه نرم افزار است؛ بنابراین روش ارائه شده بر اساس شبکه عصبی معرفی شده است.

۳-۴-۲ روش های ارائه شده

در توسعه نرم‌افزار ریسک‌های بالایی از خطا در تخمین هزینه پروژه‌ها با استفاده از مدل الگوریتمی کوکومو وجود دارد. در مدل های الگوریتمی مقادیر ثابت تخمین هزینه مقادیر تعریف‌شده‌ای نیستند و بنابراین پیدا کردن پاسخ‌های قابل اطمینان آسان نیست. روش ارائه شده در این پژوهش ترکیبی از الگوریتم فاخته و شبکه عصبی و مدل کوکومو (پایه) را جهت بهبود تخمین هزینه به کار گرفته است. دو روش ارائه شده عبارت‌اند از: در روش اول (COA)، الگوریتم فاخته

ابتکاری فاخته بر روی هر یک از پروژه‌ها درازای هر پروژه یک دسته از ضرایب a و b خواهیم داشت. حال باید از هر دسته آن ضرایبی که دارای فاصله کمتری تا هزینه واقعی نسبت به بقیه هستند استخراج و در آخربین ضرایبی که در یک دسته (ارگانیک - نیمه منفصل - تعبیه شده) مشترک هستند به طور مجزا میانگین بگیریم. وقتی که تخمین برای پروژه‌های یک دسته به پایان رسید باید پارامترهای عملکردی برای آن دسته محاسبه شوند. پارامترهای عملکردی موردنظری که در اینجا استفاده شده اند MMRE می باشند. برای محاسبه‌ی MMRE ابتدا لازم است تا MRE طبق رابطه (۱۲) محاسبه شود. پس از محاسبه MRE برای همه پروژه‌ها مقدار MMRE محاسبه می شود. مقدار MMRE برابر است با میانگین مقادیر به دست آمده برای MRE در دسته موردنظر که طبق رابطه (۱۳) محاسبه می شود. در ادامه MMRE به دست آمده درازای اجرای الگوریتم فرا ابتکاری استفاده شده در این پژوهش ارائه شده و در انتها بهترین آن‌ها انتخاب و به همراه ضرایب جدید ارائه خواهد شد.

۳-۴-۱-۲ آزمون در روش COA

در این بخش از نتایج حاصل از مرحله آزمون به منظور ارزیابی روش ارائه شده استفاده خواهد شد. داده‌های مورد استفاده در این بخش داده‌های آزمون هستند و همچنین از ضرایب بهینه استفاده خواهد شد. در ابتدا تمامی پروژه‌های آزمون به سه دسته (ارگانیک - نیمه منفصل و تعبیه شده) تقسیم می شوند سپس ضرایب مربوطه را به هر دسته استخراج و معادله کوکومو برای تک تک این پروژه‌ها اعمال می شود. به طور مثال اگر پروژه‌ای ارگانیک باشد ضرایب ارگانیک به آن اختصاص می یابد. سپس میزان تلاش پروژه انتخاب شده طبق معادله کوکومو محاسبه می شود (این کار برای تمامی ضرایب ایجاد شده توسط الگوریتم‌های فرا ابتکاری و روش ارائه شده در این مرحله انجام می شود). این فرایند برای تمامی پروژه‌های موجود در مرحله آزمون انجام می شود. سپس MRE مربوط به هر پروژه محاسبه می شود. به عبارت دیگر به تعداد تمامی پروژه‌ها MRE وجود خواهد داشت که میانگین

توضیح و تشریح هر یک از آن‌ها می پردازیم. بخش اول شامل بخش آموزش و بخش دوم شامل بخش آزمون می باشد.

۳-۴-۱-۱ آموزش در روش COA

روش ارائه شده در بخش آموزش ساخته شده و در واقع پارامترهای روش ارائه شده در این بخش تنظیم می شود. در این بخش سعی شده تا با تولید پارامترهای a و b جدید ضرایبی جدید جهت بهینه کردن مدل کوکومو ارائه دهیم. در ابتدا کل پروژه‌ها به دودسته آموزش و آزمون تقسیم می شوند. این تقسیم بندی کاملاً تصادفی بوده و این عمل با نسبت ۸۰ به ۲۰ انجام خواهد گرفت؛ که ۸۰ درصد از پروژه‌ها برای بخش آموزش و بقیه (۲۰٪) برای بخش آزمون انتخاب می شوند. در ادامه داده‌های آموزشی باید طبق رابطه کوکومو به ۳ دسته تقسیم شوند. این ۳ دسته شامل داده‌های ارگانیک، نیمه منفصل و تعبیه شده می باشند. این دسته بندی در مدل‌ها به دلیل درجه بالای ناهمگنی در پروژه‌های کوکومو است. حال با استفاده از الگوریتم‌های فرا ابتکاری سعی می کنیم تا ضرایب a و b را بهینه کنیم. برای بهینه کردن این ضرایب در هر یک از الگوریتم‌های فرا ابتکاری نیاز به یک تابع شایستگی داریم. تابعی که در این پژوهش مورد استفاده قرار گرفته شده در رابطه ۹ بیان شده است. منظور از تابع شایستگی (فیتنس) مقداری است که سعی داریم آن را حداقل کنیم.

$$\text{effort} = |\text{RealCost} - (a_i \times (K/\text{loc})b_i \times \text{EAF})| \quad (9)$$

منظور از RealCost در تابع بالا میزان هزینه واقعی پروژه است. در این پژوهش سعی شده با استفاده از تابع بالا میزان فاصله بین هزینه تولید شده توسط مقدار پیشنهادی الگوریتم فرا ابتکاری فاخته را با هزینه واقعی نرم افزار به حداقل برسانیم. هنگام اجرای این الگوریتم محدوده ضرایب a و b بین ۰ تا ۵ قرار گرفته اند؛ زیرا تعداد نقاط بهینه جواب در بازه‌های بالاتر از ۵ و کمتر از ۰ زیاد است (باید در نظر داشت به دلیل ساختار فرمول کوکومو در صورت مشخص نمودن یک بازه امکان وجود یک جواب بهینه برای تمامی اعداد حقیقی وجود خواهد داشت زیرا همواره درازای هر عدد در ضریب a یک عدد در اعداد حقیقی وجود خواهد داشت که با استفاده از آن بتوان فاصله بین جواب به دست آمده توسط مدل را با هزینه واقعی به حداقل رساند و یا حتی صفر کرد) بعد از اجرای الگوریتم فرا

آزمون)؛ که در بخش آموزش با استفاده از الگوریتم تکاملی فاخته از بین ۱۵ فاکتور تخمین تلاش مدل کوکومو محرک‌های مؤثرتر را بر اساس تابع برازش الگوریتم فاخته که میزان دقت تخمین (MRRE) می‌باشد انتخاب می‌شوند؛ و محرک‌های مؤثر به‌عنوان ورودی شبکه عصبی پیشنهاد داده می‌شود و خروجی شبکه عصبی مقدار تلاش توسعه می‌باشد؛ و پس از محاسبه MRE های همه پروژه ها، معیار کارایی MMRE محاسبه و به الگوریتم فاخته فرستاده می‌شود. بر پایه مقدار MMRE الگوریتم، ضرایب a و b و فاکتورهای انتخاب‌شده تخمین تلاش را به‌گونه‌ای تنظیم می‌کند که MMRE حداقل گردد و تخمین مجدداً بر اساس ضرایب جدید آغاز می‌شود. این فرآیند تا زمانی که معیار توقف الگوریتم فاخته رضایت‌بخش باشد ادامه می‌یابد؛ و در بخش آزمون مراحل بالا با یک تفاوت که دیگر به الگوریتم فاخته

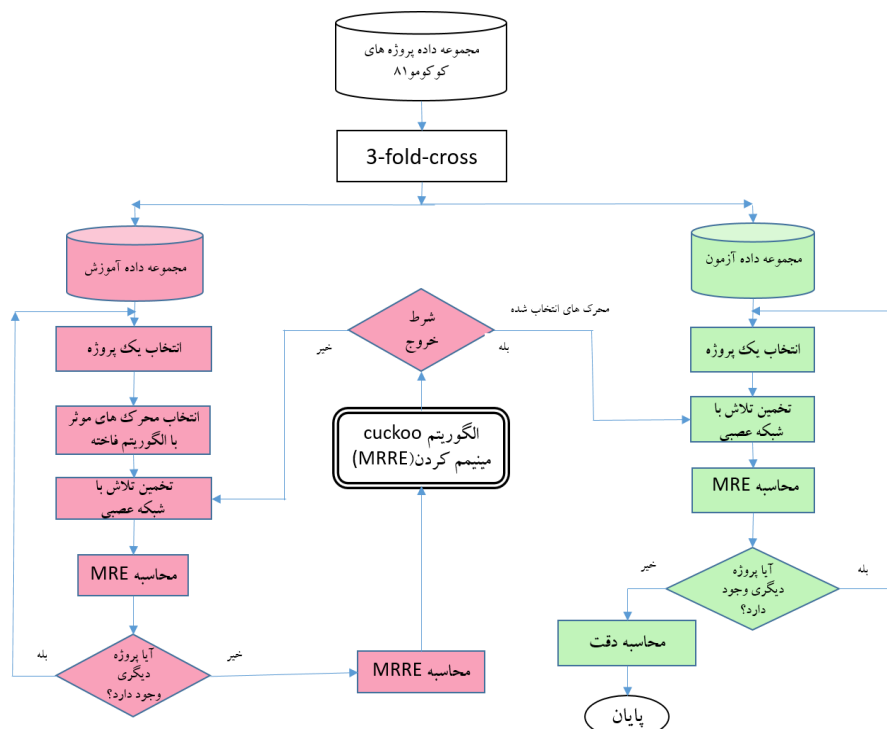
آن‌ها MMRE را تولید می‌کند. در انتها نتایج با مدل کوکومو اصلی مقایسه می‌شوند.

۳-۴-۲ روش ANN-COA انتخاب کردن محرک‌های مؤثرتر مدل کوکومو با استفاده از الگوریتم‌های فرا ابتکاری فاخته و شبکه عصبی

در این قسمت جزئیات مربوط به روش ارائه شده ANN-COA که در شکل (۳) نشان داده شده، بیان شده‌اند. این روش سعی بر آن دارد که تا حد امکان از طریق الگوریتم بهینه‌سازی فاخته محرک‌های مؤثر مدل کوکومو را انتخاب کنیم تا بهترین تخمین‌ها به کمک شبکه عصبی را به دست آورد. فرایند بخش‌های روش ارائه شده به‌گونه‌ای است که میزان خطا در تخمین‌ها حداقل شود. این روش شامل ۲ بخش اصلی است که در ادامه به توضیح و تشریح آن‌ها می‌پردازیم. بخش اول شامل بخش آموزش و بخش دوم شامل بخش آزمون می‌باشد.

۳-۴-۲-۱ آموزش و آزمون در روش ANN-COA

در این روش ابتدا داده‌ها را به دو بخش آموزش و آزمایش تقسیم می‌شوند (به‌صورت تصادفی ۸۰٪ آموزش و ۲۰٪



شکل ۳. روش ANN-COA تخمین هزینه نرم افزار

هدف اصلی همه تکنیک‌های تخمین استفاده شده در زمینه تلاش توسعه نرم‌افزار است.

برای انتخاب محرک مؤثرتر نیازی نیست چون در مرحله آموزش محرک‌ها شناسایی شده‌اند، تکرار می‌شوند.

۴- بحث و نتایج

۳-۵ پارامترهای عملکردی در روش ارائه شده

۴-۱- نتایج در روش COA

دقت تخمین تلاش یک اصل مهم در موفقیت پروژه‌های نرم‌افزاری است رایج‌ترین روش‌ها که در روش ارائه شده مورد استفاده قرار گرفته است عبارت‌اند از [۲۸]:

روش ارائه شده COA بر روی دو مدل کوکوموی پایه و کوکوموی میانه به کار گرفته شد. در هر دو مدل جهت ارزیابی نتایج از دو مجموعه داده کوکوموی ۸۱ و ناسا ۹۳ استفاده شده است.

$$MRE = \frac{|Estimated_i - Actual_i|}{Actual_i} \quad (10)$$

$$MMRE = \frac{\sum_{i=1}^N MRE_i}{N} \quad (11)$$

۴-۱-۱- نتایج با در نظر گرفتن مدل کوکوموی پایه

که $Actual_i$: تلاش واقعی پروژه i ام، $Estimate_i$: تلاش تخمین زده شده که با دو معادله کوکومو (پایه، میانه) زیر، به صورت جداگانه مقاداردهی می‌شود.

جدول (۱) نتایج روش ارائه شده بر روی مجموعه داده کوکوموی ۸۱ را نشان می‌دهد این جدول ضرایب a ، b و همچنین معیار عملکرد MMRE برای سه دسته مختلف را نشان می‌دهد. همان‌طور که مشاهده می‌شود بهترین MMRE برای دسته ۲ حاصل شده است و میانگین آن‌ها ۰/۱۳۴۳ می‌باشد.

$$Estimated_i = a \times size^b \quad (12)$$

جدول ۱. نتایج مدل کوکوموی پایه، با مجموعه داده کوکوموی ۸۱

	A	B	MMRE	PRED
ارگانیک	۵	۰/۴۳۷۵	۰/۱۸۱۸	۰/۶۶۶۶
تعبیه شده	۰/۰۲۶۴	۱/۴۹۵۹	۰/۰۹۶۳	۰/۹۲۸۵
نیمه منفصل	۰/۴۹۰۶	۰/۹۵۰۸	۰/۱۲۴۸	۰/۹۰۹۰
میانگین			۰/۱۳۴۳	۰/۸۳۴۷

$$Estimated_i = a \times size^b \times \prod_{i=1}^{15} EAF_i \quad (13)$$

MRE: میزان خطای نسبی، MMRE متوسط شدت خطای نسبی، N تعداد پروژه‌ها، a و b : ضرایب ثابت می‌باشند.

جدول (۲) نتایج روش ارائه شده بر روی مجموعه داده ناسا ۹۳ را نشان می‌دهد این جدول ضرایب a ، b و همچنین معیار عملکرد MMRE برای سه دسته مختلف را نشان می‌دهد. همان‌طور که مشاهده می‌شود بهترین MMRE برای دسته ۳ حاصل شده است و میانگین آن‌ها ۰/۳۴۷۲ می‌باشد.

EAF: ۱۵ فاکتور تنظیم تلاش که در هم ضرب می‌شوند.

پارامتر دیگری که برای ارزیابی عملکرد، استفاده می‌شود درصد پیش‌بینی، مطابق فرمول زیر می‌باشد.

$$PRED(x) = A/N \quad (14)$$

جدول ۲. نتایج مدل کوکوموی پایه، با مجموعه داده ناسا

	A	B	MMRE	PRED
ارگانیک	۴/۹۹۳۳	۰/۱۶۶۳	۰/۸۱۹۱	.
تعبیه شده	۵	۱/۱۷۴۵	۰/۱۴۸۷	۰/۸۰۹۵
نیمه منفصل	۰/۵۹۰۶	۱/۰۲۵۷	۰/۰۷۳۹	۰/۹۲۵۶
میانگین			۰/۳۴۷۲	۰/۵۷۸۶

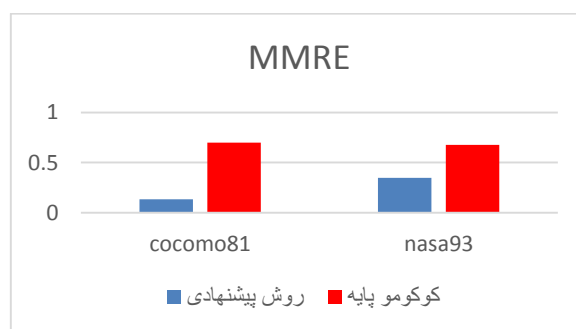
که A تعداد پروژه‌های با $MRE \leq x$ و N تعداد پروژه‌های در نظر گرفته شده می‌باشد. معمولاً سطح قابل قبول x در متدهای تخمین هزینه نرم‌افزار ۰/۲۵ [۲۹] متدهای مختلفی بر اساس این سطح مقایسه شده‌اند. کاهش MRE و افزایش PRED

۴-۱-۲- تحلیل نتایج مدل کوکوموی پایه

جدول (۳) و شکل (۴) مقادیر مربوط MMRE و PRED حاصل از مدل کوکوموی پایه و روش ارائه شده را نشان می‌دهد. نتایج از آزمایش بر روی دو مجموعه داده ناسا ۹۳ و کوکوموی ۸۱ حاصل شده است. همان‌طور که نتایج نشان می‌دهد بر روی مجموعه داده ناسا در روش ارائه شده متوسط شدت خطای نسبی: ۰٫۳۴۷۲ به دست آمده که نسبت به مدل کوکوموی پایه با مقدار ۰٫۶۷۷۵ بهبود ۵۱٫۲۴٪ را ایجاد نموده است و بر روی مجموعه داده کوکومو ۸۱ در روش ارائه شده متوسط شدت خطای نسبی ۰٫۱۳۴۳ به دست آمده که نسبت به مدل کوکوموی پایه با مقدار ۰٫۶۹۸۴ بهبود ۱۹٫۲۲٪ را ایجاد نموده است. همچنین بهبود عملکرد بر روی مجموعه داده ناسا در مقایسه با مجموعه داده کوکوموی ۸۱ بیشتر بوده است که یکی از دلایل آن را می‌توان تعداد بیشتر پروژه‌های ناسا در مجموعه داده آزمایش دانست.

جدول ۳: نتایج کلی دو مجموعه داده کوکومو و ناسا بر روی مدل کوکوموی ۸۱

	مجموعه داده کوکومو ۸۱		مجموعه داده ناسا ۹۳	
	MMRE	PRED	MMRE	PRED
کوکومو پایه	۰٫۶۹۸۴	۰٫۵۶	۰٫۶۷۷۵	۰٫۵
روش ارائه شده	۰٫۱۳۴۳	۰٫۸۳	۰٫۳۴۷۲	۰٫۵۷



شکل ۴: نمودار PRED, MMRE مدل کوکوموی پایه

۴-۱-۳- نتایج با در نظر گرفتن مدل کوکوموی میانه

جدول (۴) نتایج روش ارائه شده بر روی مجموعه داده کوکوموی ۸۱ را نشان می‌دهد این جدول ضرایب a، b و همچنین معیار عملکرد MMRE برای سه دسته مختلف را نشان می‌دهد. همان‌طور که مشاهده می‌شود بهترین MMRE برای دسته ۲ حاصل شده است و میانگین آن‌ها ۰٫۱۴۱۵ می‌باشد.

جدول ۴: نتایج مدل کوکوموی میانه، با مجموعه داده کوکومو

	A	B	MMRE	PRED
ارگانیک	۴/۹۹۸۹	۰/۵۶۷۷	۰/۱۴۸۸	۰/۸۳۳۳
تعبیه شده	۲/۵۵۰۶	۰/۴۰۰۰	۰/۰۹۹۳	۰/۹۲۸۵
نیمه منفصل	۰/۰۰۷۷	۱/۱۷۷۲	۰/۱۳۲۲	۰/۹۰۹۰
میانگین			۰/۱۴۱۵	۰/۸۹۰۲

جدول ۵: نتایج مدل کوکوموی میانه، با مجموعه داده ناسا

	A	B	MMRE	PRED
ارگانیک	۵	۰/۰۰۸۶	۰/۸۶۰۲	۰
تعبیه شده	۴/۹۴۲۱	۰/۸۲۹۲	۰/۱۲۷۱	۰/۹۰۴۷
نیمه منفصل	۴/۹۹۹۹	۰/۲۴۹۹	۰/۰۸۳۳	۰/۸۸۲۳
میانگین			۰/۳۵۶۸	۰/۵۹۵۶

جدول (۵) نتایج روش ارائه شده بر روی مجموعه داده ناسا ۹۳ را نشان می‌دهد این جدول ضرایب a، b و همچنین معیار عملکرد MMRE برای سه دسته مختلف را نشان می‌دهد. همان‌طور که مشاهده می‌شود بهترین MMRE برای دسته ۳ حاصل شده است و میانگین آن‌ها ۰٫۳۵۶۸ می‌باشد.

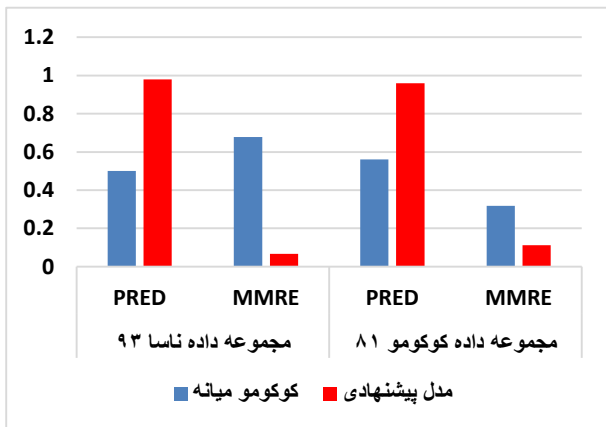
۴-۱-۴- تحلیل نتایج مدل کوکوموی میانه

جدول (۶) و شکل (۵) مقادیر مربوط به MMRE و PRED حاصل از مدل کوکوموی پایه و روش ارائه شده را نشان می‌دهد. نتایج از آزمایش بر روی دو مجموعه داده ناسا ۹۳ و کوکوموی ۸۱ حاصل شده است. همان‌طور که نتایج نشان می‌دهد بر روی مجموعه داده ناسا در روش ارائه شده متوسط شدت خطای نسبی: ۰٫۳۵۶۸ به دست آمده که نسبت به مدل کوکوموی پایه با مقدار ۰٫۶۷۷۵ بهبود ۵۲٫۶۶٪ را ایجاد نموده است و بر روی مجموعه داده کوکومو ۸۱ در روش ارائه شده متوسط شدت خطای نسبی: ۰٫۱۴۱۵ به دست آمده

۴-۱-۶- مقایسه روش COA با سایر روش‌ها

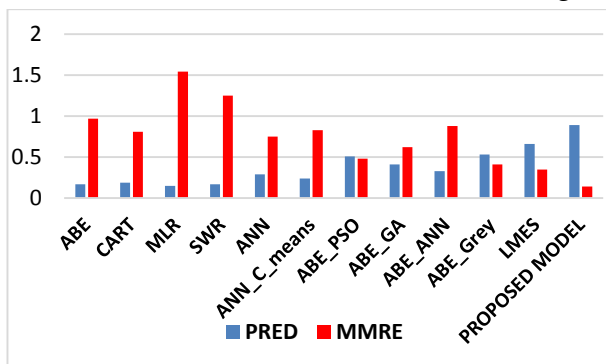
به منظور ارزیابی بهتر عملکرد روش ارائه شده، روش مورد نظر با نتایج یکی از مقالات جامع و معتبر که از همه روش‌های پیش‌بینی تلاش و از چندین دیتاست مختلف در آن استفاده شده و همچنین از نظر ارزیابی با مقاله حاضر یکسان هست، مورد مقایسه قرار گرفته است [۳۰].

شکل (۷) مقایسه عملکرد روش‌های مختلف را نشان می‌دهد. همان‌طور که مشاهده می‌شود بهترین MMRE به دست آمده



شکل ۶: مقایسه روش COA با سایر الگوریتم‌ها

مربوط به روش ارائه شده می‌باشد. کمترین درصد بهبود نسبت به مدل LMES با مقدار ۴۰/۴۲٪ حاصل شده و بیشترین درصد بهبود نسبت به مدل MLR با مقدار ۹/۱۸٪ حاصل شده است.



شکل ۷: نمودار مقایسه MMRE مدل‌های دیگر با روش ارائه شده

و همچنین به منظور ارزیابی روش ارائه شده از نتایج مقاله معتبر که در سال‌های اخیر به چاپ رسیده است استفاده شده است. در این مقاله از چندین روش برای تخمین تلاش نرم افزارهای تلفن همراه و معیارهای عملکردی (MMRE) و

که نسبت به مدل کوکوموی پایه با مقدار ۰/۳۱۷۹ بهبود ۴۴/۵۱٪ را ایجاد نموده است. همچنین بهبود عملکرد بر روی مجموعه داده ناسا در مقایسه با مجموعه داده کوکوموی ۸۱ بیشتر بوده است که یکی از دلایل آن را می‌توان تعداد بیشتر پروژه‌های ناسا در مجموعه داده آزمایش دانست.

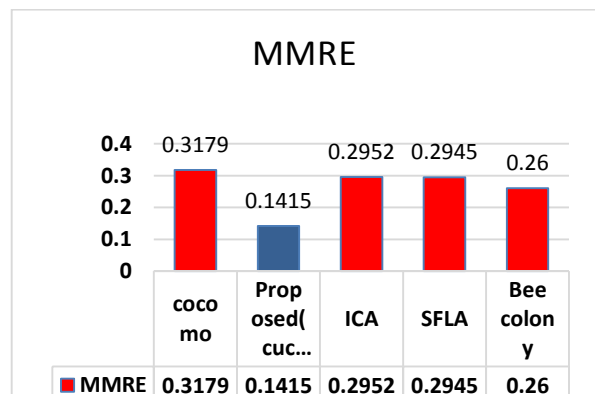
جدول ۶: نتایج کلی دو مجموعه داده کوکومو و ناسا بر روی مدل کوکوموی میانه

	مجموعه داده کوکومو ۸۱		مجموعه داده ناسا ۹۳	
	MMRE	PRED	MMRE	PRED
کوکومو میانه	۰/۳۱۷۹	۰/۵۶	۰/۶۷۷۵	۰/۵۰
روش ارائه شده	۰/۱۴۱۵	۰/۸۹	۰/۳۵۶۸	۰/۵۹

۴-۱-۵- مقایسه روش COA با نتایج سایر الگوریتم‌ها

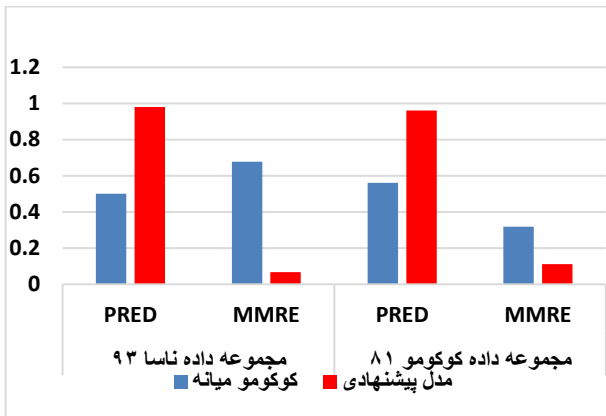
ها و با مجموعه داده کوکومو ۸۱

به منظور ارزیابی روش ارائه شده از نتایج مقالات معتبر که در سال‌های اخیر به چاپ رسیده است استفاده شده است. در این مرحله نیز از معیارهای عملکردی (MMRE) و (PRED) جهت ارزیابی استفاده شده است، که در شکل (۶) نشان داده شده است.



شکل ۵: نمودار PRED, MMRE مدل کوکوموی میانه

(PRED) جهت ارزیابی استفاده شده است، که در شکل (۸) نشان داده شده است [۳۱].



شکل ۹: نمودار نتایج مربوط به روش ANN-COA بر روی مجموعه داده کوکومو و ناسا ۹۳

۴-۲-۱- تحلیل نتایج در مدل ANN-COA

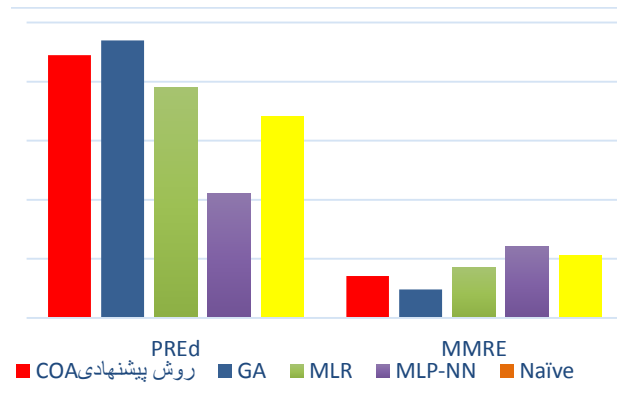
نتایج به دست آمده بر روی هر دو مجموعه داده بهبود داشته است و ما در این قسمت به تحلیل نتایج به دست آمده می‌پردازیم.

جدول (۸) و شکل (۷، ۸) مقادیر مربوط به MMRE حاصل از مدل کوکومو میانه و روش ارائه شده را نشان می‌دهد. نتایج از آزمایش بر روی مجموعه داده ناسا ۹۳ و کوکومو ۸۱ حاصل شده است. همان‌طور که نتایج نشان می‌دهد بر روی مجموعه داده ناسا در روش ارائه شده متوسط شدت خطای نسبی: $0/0669$ به دست آمده که نسبت به مدل کوکومو میانه با مقدار $0/6775$ بهبود $9/87\%$ را ایجاد نموده است؛ و بر روی مجموعه داده کوکومو ۸۱ در روش ارائه شده متوسط شدت خطای نسبی: $0/1118$ به دست آمده که نسبت به مدل کوکومو میانه با مقدار $0/3179$ بهبود $35/16\%$ را ایجاد نموده است.

۴-۲-۲- مقایسه روش ANN-COA با سایر مدل‌ها

با مجموعه داده ناسا ۹۳

به منظور ارزیابی روش ارائه شده از نتایج مقالات معتبر که در سال‌های اخیر به چاپ رسیده است استفاده شده است. در این مرحله نیز از معیارهای عملکردی (MMRE) و (PRED)



شکل ۸: مقایسه روش COA با سایر روش‌ها

همان‌طور که مشاهده می‌شود بهترین MMRE به دست آمده مربوط به روش الگوریتم ژنتیک می‌باشد. و روش ارائه شده در مرتبه دوم از نظر بهترین MMRE قرار گرفته است.

۴-۲-۳- نتایج در روش ANN-COA

در روش ارائه شده ابتدا داده‌ها را به دو بخش آموزش و آزمایش تقسیم می‌شوند (به صورت تصادفی و معمولاً 80% آموزش و 20% آزمون)؛ که در بخش آموزش با استفاده از الگوریتم تکاملی فاخته از بین ۱۵ فاکتور تخمین تلاش مدل کوکومو محرک‌های مؤثرتر را بر اساس تابع برازش الگوریتم فاخته که میزان دقت تخمین (MRRE) می‌باشد انتخاب می‌شوند؛ و محرک‌های مؤثر به عنوان ورودی شبکه عصبی پیشنهاد داده می‌شود و خروجی شبکه عصبی مقدار تلاش توسعه می‌باشد. این روش بر روی دو مجموعه داده کوکومو و ناسا پیاده‌سازی شده است. جدول (۷) و شکل (۸) نتایج حاصل از روش ارائه شده را بر روی مجموعه داده ناسا ۹۳ و کوکومو ۸۱ را نشان می‌دهند.

	مجموعه داده کوکومو ۸۱		مجموعه داده ناسا ۹۳	
	MMRE	PRED	MMRE	PRED
کوکومو میانه	$0/3179$	$0/56$	$0/6775$	$0/5$
روش ارائه شده	$0/1118$	$0/96$	$0/0669$	$0/98$

جدول ۷: نتایج روش ANN-COA بر روی دو مجموعه

۴-۳- نتیجه گیری و کارهای آینده

توسعه پروژه‌های نرم‌افزاری به دلیل ناملموس بودن، نیازمندی‌های مبهم، تولید انعطاف‌پذیر، دشوار پیچیده می‌باشد. مدیریت پروژه به شدت به تخمین دقیق و قابلیت اطمینان وابسته است. ماهیت خلاق پروژه‌های نرم‌افزاری و انتزاعی بودن آن تخمین هزینه و زمان انجام آن را بی‌نهایت مشکل می‌کند. در این تحقیق ابتدا مدل‌های استاندارد ارائه شده برای تخمین تلاش پروژه‌های نرم‌افزاری تشریح شده و سپس مدل‌های مبتنی بر الگوریتم‌های فراابتکاری مورد بحث و بررسی قرار گرفته است و در نهایت یک روش جدید تلاش لازم برای توسعه نرم‌افزار با استفاده از شبکه عصبی و الگوریتم بهینه‌سازی فاخته را ارائه شده است. مقاله حاضر تأثیر ترکیب الگوریتم بهینه‌سازی فاخته و شبکه عصبی را با مدل کوکومو پایه و میانه و با دو مجموعه داده واقعی کوکوموی ۸۱ و ناسا ۹۳ مورد بحث و بررسی قرار داده است. به منظور بررسی عملکرد روش ارائه شده معیار عملکرد MMRE محاسبه شده است. نتایج روش ارائه شده در کل باعث جدول ۱۰: نتایج به دست آمده از رویکرد ترکیبی شبکه‌های عصبی، الگوریتم رقابتی امپریالیستی [۲۱]، مقالات معتبر که در سال‌های اخیر منتشر شده است مورد مقایسه قرار گرفته است نتایج حاکی از آن است که روش ارائه شده در مقایسه با سایر مدل‌ها کمترین MMRE را به دست آورده است. دستاوردهای مانند به دست آوردن دقت قابل مقایسه با مدل‌های تخمین موجود، سرعت و همگرایی سریع الگوریتم در دست یافتن به جواب بهینه، یافتن بهترین ضرایب با استفاده از الگوریتم فاخته، به دست آوردن یک مدل انعطاف‌پذیر مبتنی بر شبکه عصبی از این مدل حاصل شد. با توجه به جدید بودن حوزه تخمین تلاش نرم‌افزار، روش ارائه شده قابلیت توسعه در زمینه‌هایی مانند بررسی میزان سازگاری مدل‌های پیشنهادی تخمین تلاش با پروژه‌های مختلف نرم‌افزاری، تمرکز بر مدل کوکومو ۲، به کارگیری سایر الگوریتم‌های بهینه‌سازی جدید جهت بهینه کردن هرچه بیشتر ضرایب و محرک‌ها می‌باشد.

مراجع

[۱] Huang, G., "Cost Modeling Based on Support Vector Regression for Complex products During the Early Design Phases". Requirements for the Degree of Doctor of Philosophy in

جهت ارزیابی استفاده شده است، که در جدول‌های (۸ و ۹) و شکل (۹) نشان داده شده است.

جدول ۸: نتایج به دست آمده از ترکیب شبکه عصبی و الگوریتم بهینه‌سازی فاخته [۲۰]

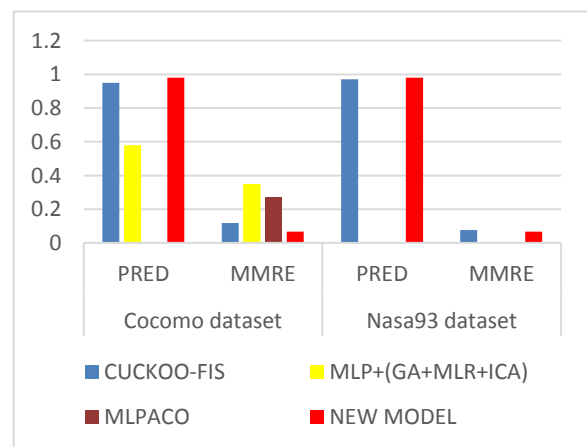
	MMRE
ANN-Cuckoo	۰/۱۴۸۲
روش ارائه شده	۰/۱۱۱۸

جدول ۹: نتایج به دست آمده از ترکیب شبکه عصبی MLP و الگوریتم بهینه‌سازی کلونی مورچگان [۱۷]

	مجموعه داده ناسا ۹۳	
	MMRE	PRED
MLPACO	۰/۲۶۹۱	-
روش ارائه شده	۰/۰۶۶۹	۰/۹۸

جدول ۱۰: نتایج به دست آمده از رویکرد ترکیبی شبکه‌های عصبی، الگوریتم رقابتی امپریالیستی [۲۱]

	مجموعه داده کوکومو	
	MMRE	PRED
MLP+GA+MLR+ICA	۰/۳۵	۰/۵۸
روش ارائه شده	۰/۱۱۱۸	۰/۹۶



شکل ۹: نمودار مقایسه روش ارائه شده ANN-COA با سایر مقالات

Archive of SID

- [۱۱] Khatibi Bardsiri, A., S.m. hashemi, and M. Razzazi, *A Novel Model for Software Services Development Effort Estimation*. Journal of Modeling in Engineering, ۲۰۱۷. ۱۵(۴۹): p. ۲۴۵-۲۶۱.
- [۱۲] Khatibi, E. and V. Khatibi Bardsiri, *An Improved Algorithmic Method for Software Development Effort Estimation*. Journal of Advances in Computer Research, ۲۰۱۸. ۹(۱): p. ۴۱-۴۹.
- [۱۳] Khatibi Bardsiri, A., *A new combinatorial framework for software services development effort estimation*. International Journal of Computers and Applications, ۲۰۱۸. ۴۰(۱): p. ۱۴-۲۴.
- [۱۴] Kumari, S. and S. Pushkar. *Software Cost Estimation Using Cuckoo Search*. in *Advances in Computational Intelligence*. ۲۰۱۷. Singapore: Springer Singapore.
- [۱۵] Puspaningrum, A. and R. Sarno, *A Hybrid Cuckoo Optimization and Harmony Search Algorithm for Software Cost Estimation*. Procedia Computer Science, ۲۰۱۷. ۱۲۴: p. ۴۶۱-۴۶۹.
- [۱۶] Çelik, E., et al. *Software test automation and a sample practice for an enterprise business software*. in *Computer Science and Engineering (UBMK)*, ۲۰۱۷ *International Conference on*. ۲۰۱۷. IEEE.
- [۱۷] Ebrahimpour, N., F. Soleimanian Gharehchopogh, and Z. Abbasi Khalifehlou, *New Approach with Hybrid of Artificial Neural Network and Ant Colony Optimization in Software Cost Estimation*. Journal of Advances in Computer Research, ۲۰۱۶. ۷(۴): p. ۱-۱۲.
- [۱۸] Wani, Z.H. and S. Quadri. *Artificial Bee Colony-Trained Industrial and Systems Engineering*, ۲۰۰۷.
- [۲] Khatibi, v. and D.N.A. awawi, *Software Cost Estimation Methods: A Review*. Journal of Emerging Trends in Computing and Information Sciences, ۲۰۱۱ ۲: p. ۲۱-۲۹.
- [۳] Abdullah, T., et al., *ANALYSIS OF SOFTWARE COST ESTIMATION MODELS*. IJEIR, ۲۰۱۲ ۱: p. ۲۰۶-۲۱۲.
- [۴] Binish Zahra, S. and M. Nazir, *A Review of Comparison among Software Estimation Techniques*. Bahria University Journal of Information & Communication Technology, ۲۰۱۲ ۵: p. ۳۹-۴۵.
- [۵] Boehm, B., "Software Engineering Economics". IEEE Transaction on Software Engineering, vol. SE-۱۰, ۱۹۸۴ p. ۴-۲۱.
- [۶] Lin, J.-C., et al., *Using Computing Intelligence Techniques to Estimate Software Effort*. International Journal of Software Engineering & Applications, ۲۰۱۳ ۴(۱): p. ۴۳.
- [۷] khatibi bardsiri, v. and m. dorosti, *An Improved COCOMO based Model to Estimate the Effort of Software Projects*. ۲۰۱۶.
- [۸] Sadeghi, B., et al., *A Novel ICA-based Estimator for Software Cost Estimation*. Journal of Advances in Computer Engineering and Technology, ۲۰۱۵. ۱(۴): p. ۱۵-۲۴.
- [۹] Venkataiah, V., et al., *Application of ant colony optimization techniques to predict software cost estimation*, in *Computer Communication, Networking and Internet Security*. ۲۰۱۷, Springer. p. ۳۱۵-۳۲۵.
- [۱۰] Shahpar, Z., et al., *Improvement of effort estimation accuracy in software projects using a feature selection approach*. Journal of Advances in Computer Engineering and Technology, ۲۰۱۶. ۲(۴): p. ۳۱-۳۸.

- [۲۵] Rajabioun, R., *Cuckoo optimization algorithm*. Applied soft computing, ۲۰۱۱. ۱۱(۸): p. ۵۵۰۸-۵۵۱۸.
- [۲۶] Karaboga, D., B. Akay, and C. Ozturk, *Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks*. MDAI, ۲۰۰۷. ۷: p. ۳۱۸-۳۱۹.
- [۲۷] Hippert, H.S. and J.W. Taylor, *An evaluation of Bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting*. Neural networks, ۲۰۱۰. ۲۳(۳): p. ۳۸۶-۳۹۵.
- [۲۸] Khatibi, E. *Investigating the effect of software project type on accuracy of software development effort estimation in COCOMO model*. in *Fourth International Conference on Machine Vision (ICMV ۱۱)*. ۲۰۱۱ International Society for Optics and Photonics.
- [۲۹] Khatibi, V. and D.N. Jawawi, *Software cost estimation methods: A review ۱*. ۲۰۱۱
- [۳۰] Bardsiri, V.K., et al., *LMES: A localized multi-estimator model to estimate software development effort*. Engineering Applications of Artificial Intelligence, ۲۰۱۳. ۲۶(۱۰): p. ۲۶۲۴-۲۶۴۰.
- [۳۱] Pandey, M., R. Litoriya, and P. Pandey, *Validation of existing software effort estimation techniques in context with mobile software applications*. Wireless Personal Communications, ۲۰۲۰. ۱۱۰(۴): p. ۱۶۵۹-۱۶۷۷.
- [۲۹] Chhabra, S. and H. Singh, *Optimizing design parameters of fuzzy model based COCOMO using genetic algorithms*. International Journal of Information Technology, ۲۰۱۹.
- [۲۰] Desai, V.S. and R. Mohanty. *ANN-Cuckoo Optimization Technique to Predict Software Cost Estimation*. in *۲۰۱۸ Conference on Information and Communication Technology (CICT)*. ۲۰۱۸. IEEE.
- [۲۱] Khazaiepoor, M., A. Khatibi Bardsiri, and F. Keynia, *A Hybrid Approach for Software Development Effort Estimation using Neural networks, Genetic Algorithm, Multiple Linear Regression and Imperialist Competitive Algorithm*. International Journal of Nonlinear Analysis and Applications, ۲۰۲۰. ۱۱(۱): p. ۲۰۷-۲۲۴.
- [۲۲] Boehm, B., C. Abts, and S. Chulani, *Software development cost estimation approaches—A survey*. Annals of software engineering, ۲۰۰۰. ۱۰(۱-۴): p. ۱۷۷-۲۰۵.
- [۲۳] Stutzke, R.D. and M. Crosstalk, *Software estimating technology: A survey*. ۱۹۹۷: Los. Alamitos, CA: IEEE Computer Society Press.
- [۲۴] جهانشاهی، س. و. خطیبی بردسیری، و همکاران. کاربرد روشهای محاسبات نرم در بهبود عملکرد مدل کوکومو بمنظور تخمین هزینه پروژه های نرم افزار. همایش ملی مهندسی کامپیوتر و فناوری اطلاعات، اسفند ۱۳۹۲

