



Semnan University

# Journal of Modeling in Engineering

Journal homepage: <https://modelling.semnan.ac.ir/>



Research Article

## A flexible effort estimator model based on ASO algorithm

Amin Moradbeiky<sup>1</sup>, Vahid Khatibi Bardsiri<sup>1,\*</sup>, Mehdi Jafari<sup>2</sup>

1. Department of Computer Engineering, Faculty of Science, Kerman Branch, Islamic Azad University, Kerman, Iran

2. Department of Electrical Engineering, Faculty of Engineering, Kerman Branch, Islamic Azad University, Kerman, Iran

\*Corresponding Author: Vahid Khatibi Bardsiri

---

### PAPER INFO

#### Article history:

Received: 11 December 2021

Revised: 21 March 2022

Accepted: 29 June 2022

#### Keywords:

Atom Search Optimization, Development Effort Estimation, Machine Learning, Software Project

---

### ABSTRACT

Accurate estimation of required effort for software development plays an important role in the success of the software project. This is always a challenging issue due to the intangible nature of the software project. Therefore, a large category of researches have been performed to develop accurate tools to estimate the required efforts for software development. According to the presented papers in related works, the adoption of methods to identify the types of relationship between software project features and features affecting the required effort for software development have a significant impact on effort estimation accuracy increment. In addition, the effectiveness of various features on the software development effort estimation is different. So, the feature effectiveness determination is advantageous in increasing the effort estimation accuracy. This paper presents a new model consisting of sub-models for project features analyzing and it uses a new and accurate heuristic algorithm called Atom Search Optimization (ASO) Algorithm to configure tools and data modeling methods. The presented model in this article is designed in multiple layers and the sub-models are organized in separate layers. The organizations of sub-models are in such a way to increase performance of other layers and ultimately increase the final estimate accuracy. In accuracy evaluation of the proposed model, 3 data sets from real projects are used and the comparisons of the results with different methods are presented. Based on the results, the proposed model leads to significant improvement of final effort estimation accuracy.

© 2023 Published by Semnan University Press.

DOI: <https://doi.org/10.22075/jme.2022.25440.2185>

---

### How to cite this article:

Moradbeiky, A., Khatibi, V., & Jafari, M. (2023). A flexible effort estimator model based on ASO algorithm. *Journal of Modeling in Engineering*, 21(73), 1-16. doi: 10.22075/jme.2022.25440.2185

## یک مدل بر آوردگر تلاش انعطاف پذیر بر اساس الگوریتم ASO

امین مرادبیکلی<sup>۱</sup>، وحید خطیبی بردسیری<sup>۲\*</sup>، مهدی جعفری<sup>۳</sup>

اطلاعات مقاله	چکیده
<p>نوع مقاله: پژوهشی دریافت مقاله: ۱۴۰۰/۰۹/۲۰ بازنگری مقاله: ۱۴۰۱/۰۱/۰۱ پذیرش مقاله: ۱۴۰۱/۰۴/۰۸</p>	<p>مدیریت موفق پروژه توسعه نرم افزار، نیازمند تخمین دقیق هزینه‌ها است. تخمین دقیق هزینه‌ها باعث برنامه‌ریزی بهتر، با حداقل هزینه خواهد شد. اما تخمین هزینه‌های پروژه نرم افزار، بدلیل ماهیت ناملموس نرم افزار، همواره با چالش‌های فراوانی روبه‌رو بوده است. از این رو، پژوهش‌های فراوانی در جهت توسعه ابزارهای دقیق برای تخمین تلاش مورد نیاز برای توسعه نرم افزار انجام شده است. بر اساس ادبیات تحقیق، استفاده از شیوه‌های شناسایی نوع ارتباط میان ویژگی‌های پروژه نرم افزار و ویژگی‌های موثر بر تلاش مورد نیاز برای توسعه نرم افزار، تاثیر بسزایی بر افزایش دقت تخمین مقدار این تلاش دارد. ایده این مقاله ارائه یک مدل جدید متشکل از زیرمدل‌هایی برای تحلیل ویژگی‌های پروژه و استفاده از یک الگوریتم هیوریستیک جدید و دقیق به نام الگوریتم بهینه‌سازی جستجوی اتم (ASO) برای پیکربندی ابزارها و روش‌های مدلسازی داده بوده است. سازماندهی زیرمدل‌ها به نحوی انجام شده است که موجب افزایش کارایی یکدیگر و در نهایت افزایش دقت تخمین نهایی شده‌اند. برای ارزیابی دقت مدل پیشنهادی، از سه مجموعه داده از پروژه‌های واقعی استفاده شده است و نتایج بدست آمده با نتایج روشهای مختلف بر اساس معیارهای MMRE، MdmMRE و Pred مقایسه شده است. در انتها برای اثبات برتری مدل پیشنهادی نسبت به سایر مدل‌ها و روش‌ها از تحلیل آماری wilcoxon استفاده شده است. نتایج بدست آمده، نشان‌دهنده دقت بالای مدل پیشنهادی است.</p>
<p><b>واژگان کلیدی:</b> الگوریتم بهینه سازی جستجوی اتم، پروژه نرم افزار، تخمین تلاش توسعه، یادگیری ماشین.</p>	

### ۱-مقدمه

مدیران هر پروژه همواره با مسئله محدودیت منابع مواجه هستند. بنابراین نیاز به مدیریت صحیح منابع، مسئله بسیار مهمی است. برای مدیریت پروژه نرم افزار و برنامه ریزی برای توسعه آن، نیازمند پیش‌بینی دقیق هزینه‌های مورد نیاز هستیم. در خصوص پروژه‌های نرم افزار این مسئله با چالش بیشتری همراه است زیرا ماهیت نرم افزار ناملموس است، به سادگی قابل ارزیابی نیست و شرایط تأثیرگذار در آن نیز بسیار زیاد است [۱]. در میان پژوهشگران حوزه مهندسی نرم افزار این چالش به عنوان "تخمین تلاش مورد

نیاز برای توسعه نرم افزار" مطرح است. پژوهشگران این حوزه با تکیه بر اطلاعات موجود از پروژه پیش رو که تحت عنوان ویژگی‌های پروژه از آن نام می‌برند اقدام به تخمین هزینه آن می‌کنند. برخی از محققان نیز با جمع‌آوری اطلاعات پروژه‌های انجام شده در گذشته و تحلیل ویژگی‌های آنها، سعی در ارائه شیوه دقیق، برای تخمین هزینه‌های پروژه نرم افزار دارند. در ابتدا برای این مسئله راه‌حلی با نام روش قضاوت متخصص ارائه شد [۲]. در سال‌های بعد تعداد زیادی از پژوهشگران روش‌های فرموله شده‌ای را ارائه دادند که با استفاده از یک شیوه‌ی مشخص و از قبل پیش‌بینی

\* پست الکترونیک نویسنده مسئول: int.uzah@gmail.com

۱. دکتری کامپیوتر، دانشکده علوم، گروه مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد کرمان، کرمان، ایران

۲. استادیار، دانشکده علوم، گروه مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد کرمان، کرمان، ایران

۳. استادیار، دانشکده مهندسی، گروه برق، دانشگاه آزاد اسلامی واحد کرمان، کرمان، ایران

منظور افزایش دقت تخمین پیشنهاد شده است. در مقاله [۲۴] با استفاده از الگوریتم کلونی مورچگان<sup>۴</sup> (ACO)، پارامترهای روش ABE تنظیم شده تا دقت تخمین این روش را افزایش دهد. در مقاله [۲۵] یک مدل مبتنی بر الگوریتم گرده افشانی گل<sup>۵</sup> (FPA) پیشنهاد شده است که در آن، این الگوریتم بر اساس ویژگی های پروژه در تکرارهای خود سعی می کند خطای تخمین را کاهش دهد. در [۲۶] مدلی مبتنی بر الگوریتم FA<sup>۶</sup> و الگوریتم PSO ارائه شده است.

گروهی از پژوهشگران در مدل پیشنهادی خود روی شبکه عصبی مطالعه کرده اند. آنها سعی کردند دقت مدل سازی توسط شبکه عصبی را افزایش دهند. آنها نشان دادند که شبکه عصبی در صورت پیکربندی مناسب به خوبی می تواند خود را با محیط مسائل پویا منطبق کند. در [۲۷] از شبکه عصبی برای تخمین تلاش توسعه نرم افزار استفاده شده است. در [۲۸] چهار معماری مختلف بر مبنای شبکه عصبی برای تخمین تلاش توسعه نرم افزار ارائه و این چهار معماری مقایسه شده است. در [۱۹] از شبکه عصبی برای یادگیری عمیق در تخمین تلاش توسعه نرم افزار استفاده شده است. در [۲۹] ساختار جدیدی برای پرسپترون شبکه عصبی ارائه و با این روش تلاش مورد نیاز برای توسعه نرم افزار تخمین زده شده است. در [۳۰] از ترکیب الگوریتم مرغ آلاچیق ساز<sup>۷</sup> (SBO) و شبکه عصبی - فازی برای تخمین تلاش استفاده شده است. در [۳۱] از الگوریتم ملخ برای پیکربندی شبکه عصبی استفاده شده است.

برخی از پژوهشگران بر شناسایی ویژگی های مهم پروژه و ارتباط میان آنها بر مقدار تلاش مورد نیاز برای توسعه نرم افزار تاکید کردند. در [۳۲] بر شناسایی ویژگی های مرتبط با یکدیگر که روی مقدار تلاش اثر دارند تاکید شده است. در [۳۳] با استفاده از شبکه عصبی ویژگی های موثر بر مقدار تلاش شناسایی شده است. در [۲۳] از الگوریتم PSO و در [۳۴] از تکنیک بیژین برای شناسایی ویژگی های اثرگذار روی مقدار تلاش مورد نیاز برای توسعه نرم افزار استفاده شده است. در جدول ۱ تعدادی از پژوهش های اخیر بررسی شده است. ایده مدل ارائه شده در این پژوهش بر اساس تحلیل نوع

شده و با استفاده از فرمول های ریاضی اقدام به تخمین تلاش توسعه نرم افزار می کردند. از جمله این روش ها می توان به روش کوکومو [۳]، کوکومو ۲ [۴]، تحلیل نقاط تابع [۵] و مدل دوتی [۶] اشاره کرد. گروهی دیگر از پژوهشگران نیز از شیوه های مبتنی بر رگرسیون مانند رگرسیون خطی [۷، ۸] یا از شیوه رگرسیون غیر خطی [۸] یا از شیوه رگرسیون درختی [۹، ۱۰] استفاده کردند. از عمده ترین معایب روش های فرموله شده، عدم تطبیق با شرایط جدید است [۱۱]. بنابراین شیوه هایی ابداع شده که انعطاف پذیری بیشتری متناسب با شرایط، از خود نشان دهند. یکی از این روش ها، شیوه تخمین مبتنی بر مقایسه<sup>۱</sup> (ABE) است. روش ABE قابلیت این را دارد که هم بصورت مستقل و هم بصورت یک ماژول از یک ماشین تخمین زنده، مورد استفاده قرار گیرد. تاکنون در مقالات فراوانی روش ABE با الگوریتم های فرا اکتشافی ترکیب شده است. از موارد ترکیب شیوه های الگوریتم های فرا اکتشافی با روش ABE می توان به ترکیب منطق فازی [۱۲، ۱۳]، ترکیب منطق فازی و طبقه بندی [۱۴]، استفاده از شبکه بیژین [۱۵]، بکارگیری الگوریتم بهینه سازی ازدحام ذرات<sup>۲</sup> (PSO) برای افزایش دقت [۱۶] اشاره کرد. از دیگر روش های ترکیبی بکار رفته برای افزایش دقت می توان به ترکیب شبکه عصبی - فازی [۱۷]، استفاده از الگوریتم PSO برای افزایش دقت منطق فازی [۱۸] و بکارگیری یادگیری عمیق در شبکه عصبی اشاره کرد [۱۹]. در [۲۰] از الگوریتم DE<sup>۳</sup> برای افزایش دقت تخمین روش ABE استفاده شده است.

در برخی از پژوهش ها از الگوریتم های هوش مصنوعی برای افزایش دقت شیوه ها یا ابزارهای تخمینگر استفاده شده است. در این پژوهش ها، سازماندهی ابزارهای تخمین گر و الگوریتم هوش مصنوعی به صورت یک مدل ابداعی توسط پژوهشگران انجام می شود. این سازماندهی به صورتی است که دقت تخمین را افزایش می دهد. مقاله [۲۱] به بررسی پژوهش هایی که از این شیوه استفاده کرده اند پرداخته است. در مقاله [۲۲] یک مدل ارائه شده است که از الگوریتم ژنتیک برای افزایش دقت تخمین استفاده کرده است. در مقاله [۲۳] انتخاب ویژگی با استفاده از الگوریتم PSO به

<sup>۵</sup> Flower Pollination Algorithm

<sup>۶</sup> Firefly algorithm

<sup>۷</sup> Satin Bowerbird Optimization

<sup>۱</sup> Analogy Based Estimation

<sup>۲</sup> Particle Swarm Optimization

<sup>۳</sup> Differential evolution

<sup>۴</sup> Ant Colony Optimization

در این مدل هر لایه باعث افزایش دقت لایه بعد می‌گردد. به عبارت ساده‌تر خروجی هر لایه از این مدل به عنوان ورودی لایه بعد است که باعث تقویت عملکرد نهایی مدل پیشنهادی شده است. در این مدل، سازماندهی الگوریتم هیوریستیک به روشی بوده است که باعث هماهنگی و انعطاف‌پذیری لایه‌ها و هم‌افزایی آنها در راستای افزایش دقت تخمین شده است.

در ادامه این مقاله و در بخش ۲ روش ABE که برای تخمین تلاش توسعه نرم افزار بکار رفته توضیح داده شده است. در بخش ۳ معیارهایی برای محاسبه دقت مدل پیشنهادی معرفی شده است. در بخش ۴ مدل پیشنهادی معرفی شده است. در بخش ۵ یک روش برای ارزیابی میزان پایدار بودن نتایج روش پیشنهادی ارائه شده است. در بخش ۶ به معرفی ۳ مجموعه داده از پروژه‌های واقعی که مدل پیشنهادی توسط این مجموعه داده‌ها آزمایش شده است پرداخته شده است. در بخش ۷ نتایج آزمایش مدل پیشنهادی و در بخش ۸ تحلیل این نتایج آمده است.

مدل‌های ارائه شده در پژوهش‌های پیشین بود. در گروهی از پژوهش‌های گذشته، ابزارهای پالایش داده برای افزایش دقت تخمین، پیشنهاد شده بود. گروه دیگری از این مطالعات، بر متفاوت بودن اثربخشی ویژگی‌ها مختلف پروژه بر مقدار تلاش، تأکید داشته و سعی کردند مدلی ارائه نمایند تا بتواند مقدار تلاش را با توجه مقدار ویژگی‌ها و نیز بر اساس میزان موثر بودن آن نوع ویژگی، به شکل دقیق تخمین بزنند. عموماً در این مطالعات میزان تأثیر به شکل ضریب تعریف می‌شود. مدلسازی از داده اقدام دیگری بود که در مطالعات گذشته به شکل‌های مختلف انجام شده بود که در مجموعه‌ای از پژوهش‌ها به آن توجه شده بود. در این دسته از مطالعات به شکل‌های متنوع از روش‌های یاگیری ماشین استفاده شده است. با توجه به موارد اشاره شده، در مدل جدیدی که در این مقاله ارائه شده است تکنیک‌ها و ابزارهای متنوع که پیش از این به طور مجزا توسط محققان بکار گرفته شده بودند و باعث بهبود دقت تخمین شده بودند، به شکل لایه‌های مجزا و در یک مدل بکار رفته‌اند.

جدول ۱ - شیوه‌های مختلف تخمین تلاش توسعه نرم افزار

مرجع	روش	روش ارزیابی	مجموعه داده	سال	شماره
[۲۷]	ANN	MMRE, Pred, MSE	21 Project(1 Dataset)	۲۰۱۹	۱
[۲۸]	ANN	MAE, Pred, MMRE	COCOMO, NASA, Kemerer	۲۰۲۱	۲
[۳۰]	ANFIS	MMRE, Pred	ISBSG, Albrecht, Kemerer	۲۰۱۷	۳
[۳۵]	ABE	MMRE, Pred, MdmRE	CF, DPS	۲۰۰۷	۴
[۳۶]	PSO, ABE	MMRE, Pred	CF, DPS, ISBSG	۲۰۱۳	۵
[۳۷]	Firefly algorithm	MMRE, Pred	Desharnais, COCOMO	۲۰۱۹	۶
[۳۸]	Fuzzy	MMRE, VAF	4 Project(1 Dataset)	۲۰۱۹	۷
[۳۹]	Neuro fuzzy	MMRE	COCOMO	۲۰۱۸	۸
[۱۷]	ANFIS	MMRE, Pred	Kemerer, Albrecht	۲۰۲۱	۹
[۱۸]	PSO, Fuzzy	MMRE, Pred	COCOMO	۲۰۲۰	۱۰
[۱۵]	Bayesian network	MMRE	COCOMO	۲۰۱۶	۱۱
[۴۰]	SVR	MAE	ISBSG	۲۰۱۸	۱۲
[۴۱]	Cuckoo Search	MMRE	COCOMO	۲۰۱۷	۱۳
[۴۲]	DE	MMRE, Pred, MAE	COCOMO	۲۰۱۸	۱۴
[۲۴]	ACO, ABE	MMRE, Pred	ISBSG, Desharnais	۲۰۲۰	۱۵

## ۲- روش ABE

پیدا کند. سپس بر اساس میزان شباهت کلی پروژه‌ها، اندازه پارامتر مورد نظر را تخمین می‌زند. روش ABE برای تخمین مقدار نامعلوم یک ویژگی مهم (به عنوان مثال تلاش یا هزینه) از یک پروژه در مقابل چند پروژه دیگر است. این روش شامل چند مرحله است که در ادامه مورد بررسی قرار گرفته است.

اساس این روش بر این اصل استوار است که پروژه‌ها با ویژگی‌های مشابه در آینده شرایط مشابهی را تجربه خواهند کرد. بر اساس این اصل، روش ABE سعی می‌کند تا شبیه‌ترین پروژه‌ها به پروژه‌ای که قصد تخمین آن را داریم

## ۱-۲- تابع شباهت

تابع شباهت، میزان شباهت پروژه‌ها را بوسیله بررسی ویژگی‌های دارای مقدار مشخص، تعیین کرده است. به این منظور از ۲ روش تعیین شباهت اقلیدسی (فرمول (۱)) و منهن (فرمول (۲)) استفاده شده است. ویژگی‌های پروژه شامل دو گروه عددی و غیر عددی است. در روش اقلیدسی و منهن برای ویژگی‌های عددی، میزان فاصله ویژگی‌های عددی، برای محاسبه میزان تفاوت ۲ پروژه محاسبه شده است. همچنین برای ویژگی‌های غیر عددی نیز میزان تفاوت، بر اساس مقدار به اندازه ۰ یا ۱ در نظر گرفته شده است. تفاوت دو روش اقلیدسی و منهن در نحوه محاسبه فاصله مقادیر ویژگی‌های عددی است. در فرمول‌های (۱) و (۲) عبارت  $p$  و  $p'$ ، پروژه‌های مورد بررسی هستند.  $f_i$  و  $f'_i$  به ترتیب ویژگی  $i$  ام پروژه  $p$  و  $p'$  هستند. نتیجه نهایی بدست آمده میزان شباهت دو پروژه را بیان می‌کند.

$$\delta = 0.0001 \quad (1)$$

$$sim(p, p') = \frac{1}{\sqrt{\sum_{i=1}^n w_i Dis(f_i, f'_i) + \delta}}$$

$$Dis(f_i, f'_i) = \begin{cases} (f_i - f'_i)^2 & \text{if } f_i \text{ and } f'_i \text{ are numerical or ordinal} \\ 0 & \text{if } f_i \text{ and } f'_i \text{ are nominal and } f_i = f'_i \\ 1 & \text{if } f_i \text{ and } f'_i \text{ are nominal and } f_i \neq f'_i \end{cases}$$

$$\delta = 0.0001 \quad (2)$$

$$sim(p, p') = \frac{1}{\left[ \sum_{i=1}^n w_i Dis(f_i, f'_i) + \delta \right]}$$

$$Dis(f_i, f'_i) = \begin{cases} |f_i - f'_i| & \text{if } f_i \text{ and } f'_i \text{ are numerical or ordinal} \\ 0 & \text{if } f_i \text{ and } f'_i \text{ are nominal and } f_i = f'_i \\ 1 & \text{if } f_i \text{ and } f'_i \text{ are nominal and } f_i \neq f'_i \end{cases}$$

## ۲-۲- تابع راه حل

تابع راه‌حل برای تخمین مقدار ویژگی تلاش یک پروژه بر اساس اندازه آن ویژگی در  $K$  پروژه شبیه‌تر استفاده شده است. این تابع از مقدار ویژگی که قصد تخمین آن را داریم در  $K$  پروژه شبیه‌تر استفاده کرده است. در تابع راه‌حل بر اساس میزان شباهت پروژه هدف با یکی از  $K$  پروژه، به اندازه ویژگی که قصد تخمین آن را داریم ضریب ارزشی داده می‌شود. تاکنون مقادیر مختلفی برای مقدار  $K$  در تابع راه‌حل ارائه شده است. تابع راه‌حل در فرمول (۳) نشان داده

شده است.

$$C_p = \frac{\sum_{k=1}^K Sim(p, p_k)}{\sum_{i=1}^K Sim(p, p_i)} C_{p_k} \quad (3)$$

در این معادله  $P$  پروژه‌ای است که قصد تخمین مقدار تلاش آن را داریم.  $P_i$  پروژه  $i$  ام از مجموعه  $K$  پروژه شبیه‌تر است.  $C_{p_i}$  مقدار ویژگی معلوم که قصد تخمین آن را داریم از  $i$  امین پروژه شبیه‌تر است.

۳-۲- انتخاب مقدار  $K$ 

مقدار  $K$  که در تخمین تلاش بکار رفته است تأثیر زیادی در دقت تخمین تلاش دارد. مقدار مناسب  $K$  وابستگی زیادی به پروژه‌های مورد بررسی دارد. چنانچه پروژه‌های مورد بررسی تفاوت بالایی با یکدیگر داشته باشند مقدار  $K$  بالا باعث کاهش دقت خواهد شد. علت بروز این مشکل آن است که پروژه‌های موثر در مرحله نهایی تخمین، تفاوت زیادی با یکدیگر خواهند داشت. از سوی دیگر چنانچه پروژه‌های مورد بررسی به یکدیگر نزدیک باشند مقدار  $K$  کوچک باعث عدم بررسی پروژه‌های مشابه خواهد شد در حالی که وجود این پروژه‌ها در مرحله نهایی بررسی می‌تواند تأثیر مثبتی بر دقت نتایج حاصل داشته باشد. این تأثیر مثبت، از طریق کاهش نویز در محاسبات اتفاق خواهد افتاد. با توجه به مطالب مطرح شده در بالا مقدار ثابتی برای  $K$  نمی‌توان در نظر گرفت. بنابراین بهتر است مقدار  $K$  به صورت پویا تعیین شود.

## ۳- معادلات محاسبه خطای تخمین

در این پژوهش برای نشان دادن خطای حاصل از تخمین، از معادلات مشخصی استفاده شده است که مورد استفاده بسیاری از پژوهشگران این حوزه است. استفاده از این معادلات موجب می‌شود تا نتایج بدست آمده از این پژوهش با نتایج بدست آمده از سایر پژوهش‌های مشابه، قابل مقایسه و ارزیابی باشد. معادلات مورد استفاده در این مقاله خطای نسبی<sup>۱</sup> (RE)، بزرگی خطای نسبی<sup>۲</sup> (MRE)، میان بزرگی خطای نسبی<sup>۳</sup> (MdMRE)، درصد پیش‌بینی<sup>۴</sup> (PRE) و میانگین بزرگی خطای نسبی<sup>۵</sup> (MMRE) هستند که در معادلات ۴ تا ۸ نمایش داده شده است [۴۳].

<sup>4</sup> Prediction percentage

<sup>5</sup> Mean Magnitude of Relative Error

<sup>1</sup> Relative Error

<sup>2</sup> Magnitude of Relative Error

<sup>3</sup> Median Magnitude of Relative Error

فاز آموزش مدل پیشنهادی دارای ۲ لایه است. مدل لایه اول این فاز در شکل (۱) نمایش داده شده است. لایه اول این فاز به بررسی ویژگی‌های پروژه‌ها می‌پردازد. این لایه به دنبال آن است که موثرترین ویژگی‌ها بر روی تلاش مورد نیاز برای توسعه نرم افزار را پیدا کند. این مجموعه از ویژگی‌ها تحت عنوان features\_list در مدل این لایه نمایش داده شده است. در این لایه از الگوریتم ASO برای جستجوی بهترین مجموعه از ویژگی‌ها استفاده شده است. الگوریتم ASO در هر بار تکرار خود مجموعه‌ای از ویژگی‌ها را به روش ABE پیشنهاد می‌دهد. روش ABE بر مبنای ویژگی‌های پیشنهادی اقدام به تخمین تلاش مورد نیاز برای توسعه تک تک پروژه‌های مجموعه قسمت آموزش می‌کند. پس از تخمین همه این پروژه‌ها، دقت تخمین این پروژه‌ها محاسبه و به عنوان بازخورد به الگوریتم ASO ارسال می‌شود. الگوریتم ASO بر مبنای بازخورد دریافت شده، اقدام به پیشنهاد مجموعه جدیدی از ویژگی‌ها می‌کند. الگوریتم ASO تا زمانی تکرار می‌شود که میزان دقت بدست آمده در این الگوریتم از عدد مشخصی بیشتر شود یا تعداد تکرارهای از پیش تعریف شده این الگوریتم به پایان برسد. خروجی این لایه مجموعه موثرترین ویژگی‌ها روی تلاش مورد نیاز برای توسعه نرم افزار است. این خروجی توسط لایه ۲ فاز آموزش و همچنین در فاز آزمایش مدل پیشنهادی استفاده می‌شود.

$$RE = \frac{Estimate - Actual}{Actual} \quad (۴)$$

$$MRE = \frac{|Estimate - Actual|}{Actual} \quad (۵)$$

$$MdMRE = Median(MRE) \quad (۶)$$

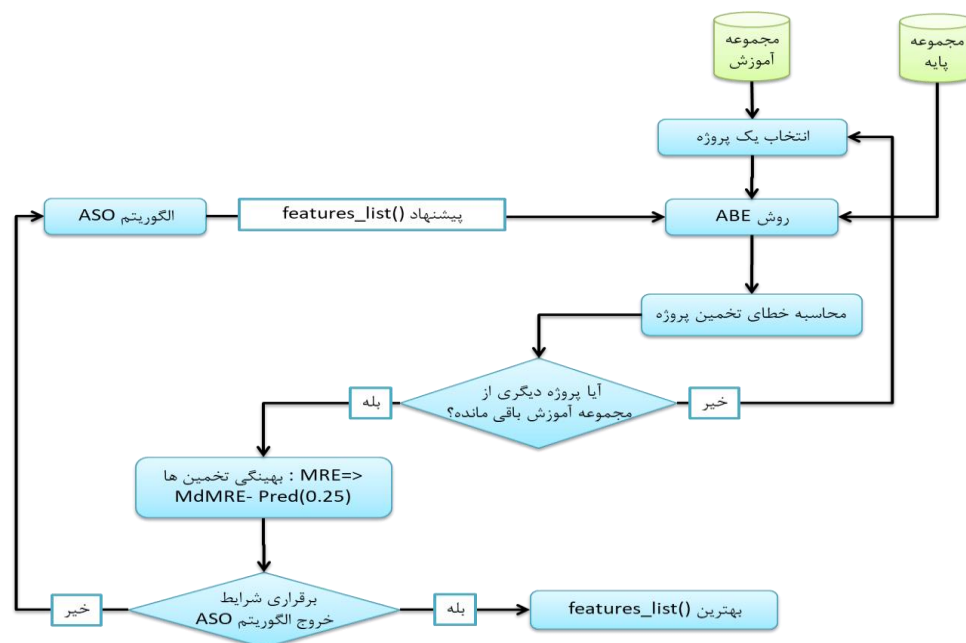
$$PRED(X) = \frac{A}{N} \quad (۷)$$

$$MMRE = Mean(MRE) \quad (۸)$$

#### ۴- مدل پیشنهادی

در این مقاله یک مدل برای تخمین تلاش توسعه نرم افزار ارائه شده است. مدل ارائه شده دارای دو فاز آموزش و آزمایش است. این مدل ابتدا در فاز آموزش با استفاده از مجموعه داده‌های پروژه‌های واقعی آموزش می‌بیند سپس در فاز آزمایش دقت آن محاسبه می‌شود. به این منظور ابتدا مجموعه داده‌های پروژه‌ها به سه قسمت آموزش، پایه و آزمایش تقسیم می‌شود. پروژه‌های قسمت آموزش به مدل فاز آموزش و پروژه‌های قسمت آزمایش به مدل فاز آموزش ارسال می‌شود. پروژه‌های قسمت پایه در هر دو فاز آموزش و آزمایش استفاده می‌شود. فاز آموزش مدل پیشنهادی دارای ۲ مرحله است. هر یک از این مراحل به عنوان یک لایه در نظر گرفته شده است. در ادامه این بخش، فازهای مدل پیشنهادی معرفی شده است.

#### ۴-۱- فاز آموزش



شکل ۱ - معماری لایه ۱ فاز آموزش

می‌شود. الگوریتم ASO در تکرارهای بعدی خود سعی می‌کند تا با پیشنهاد پارامترهای بهتر برای معادله ۹ و روش ABE به تخمین دقیق‌تری برای پروژه های مجموعه آموزش دست یابد. خروجی این لایه بهترین پارامترها برای معادله ۹ و روش ABE است. این خروجی‌ها توسط فاز آزمایش مدل پیشنهادی استفاده می‌شود.

$$Effort = (estimation \times x_1) \pm x_2 \quad (9)$$

#### ۴-۲- فاز آزمایش

معماری فاز آزمایش مدل پیشنهادی در شکل (۳) نمایش داده شده است. این فاز روی مجموعه پروژه‌های قسمت آزمایش و پایه کار می‌کند. ورودی‌های این فاز، بهترین ویژگی‌های بدست آمده از لایه ۱ فاز آموزش، بهترین پارامترهای مربوط به پیکربندی روش ABE و معادله ۹ حاصل از لایه ۲ فاز آموزش، است. در ابتدای فرآیند این فاز، هر یک از پروژه‌های قسمت آزمایش با استفاده از روش ABE تخمین زده می‌شود. پیکربندی روش ABE بر اساس نتایج بدست آمده از لایه ۲ فاز آموزش انجام می‌شود. در ادامه فرآیند این فاز، تخمین حاصل از روش ABE به معادله ۹ ارسال می‌شود. معادله ۹ با استفاده از نتایج لایه ۲ فاز آموزش پیکربندی می‌شود. تخمین نهایی با استفاده از این معادله محاسبه می‌شود. اگر پروژه دیگری از مجموعه پروژه‌های قسمت آزمایش باقی مانده باشد فرآیند یک بار دیگر برای آن طی خواهد شد. پس از تخمین تمام پروژه‌های قسمت آزمایش، دقت نهایی محاسبه می‌شود. دقت تخمین بدست آمده در این فاز، مشخص کننده دقت مدل پیشنهادی است.

#### ۵- روش ارزیابی

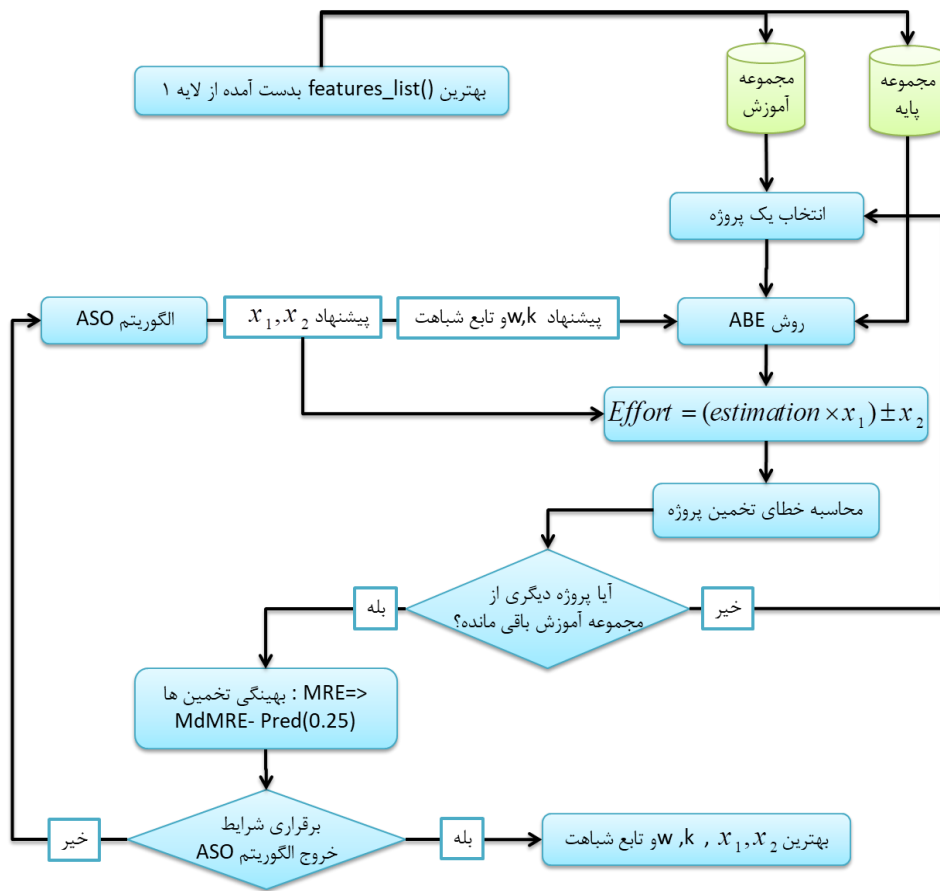
نحوه چینش نمونه‌ها در گروه آزمایش یا آموزش بر روی خطای حاصل و نیز کیفیت آموزش شبکه تأثیر بسزایی دارد [۴۴]. بنابراین برای نشان دادن پایداری نتایج حاصل از مدل پیشنهادی نیازمند روشی هستیم تا استقلال نتایج بدست آمده از محل قرارگیری نمونه‌ها را نشان دهد. برای دستیابی به این هدف، روش‌های ارزیابی مختلفی مانند 3fold، 10fold، LOO<sup>1</sup> و ... وجود دارد که در این پژوهش از روش LOO استفاده شده است. در روش LOO هر بار همه پروژه‌ها به جز یک پروژه به

معماری لایه دوم فاز آموزش در شکل (۲) نمایش داده شده است. در این لایه نیز از مجموعه پروژه‌های قسمت آموزش و پایه استفاده می‌شود. اما تنها از ویژگی‌هایی استفاده می‌کنیم که توسط لایه اول توصیه شده است. در این لایه الگوریتم ASO در تکرارهای خود در جستجوی بهترین پیکربندی برای روش ABE است تا دقت تخمین این روش را افزایش دهد. روش ABE برای تخمین تلاش، شبیه‌ترین پروژه ها با پروژه هدف را جستجو می‌کند. این روش بر مبنای انطباق ویژگی عمل می‌کند. برای افزایش دقت روش ABE از ASO استفاده شده است. الگوریتم ASO سعی می‌کند تا دقیق‌ترین پیکربندی را برای روش ABE پیشنهاد دهد و به آن کمک کند تا تخمین دقیق‌تری ارائه نماید. پیکربندی ارائه شده توسط الگوریتم ASO بر اساس شرایط پروژه و ویژگی‌های آن متفاوت است. از طرف دیگر لایه ۱ با استفاده از پالایشی که روی داده‌های ورودی به این لایه انجام می‌دهد به افزایش دقت لایه ۲ کمک می‌کند. به عبارت ساده‌تر به کمک لایه ۱ داده با کیفیت بهتر در اختیار لایه ۲ قرار می‌گیرد. الگوریتم ASO علاوه بر پارامترهای مربوط به پیکربندی روش ABE دو پارامتر دیگر به نام های  $x_1$  و  $x_2$  نیز پیشنهاد می‌دهد. این دو پارامتر مربوط به معادله ۹ است. معادله ۹ در معماری لایه ۲ آمده است. این معادله مقدار تخمین یک پروژه را از روش ABE دریافت می‌کند. مقدار تخمین نهایی یک پروژه با استفاده از معادله ۹ و با توجه به پارامتر های  $x_1$  و  $x_2$  محاسبه می‌شود. علت پیشنهاد این معادله در این مرحله از این لایه، افزایش انعطاف‌پذیری مدل پیشنهادی در برابر داده‌های ناهمگون است. پارامترهای به کار رفته در این معادله، مدل را قادر می‌سازد تا همپوشانی بیشتری بین مدل پیشنهادی با شرایط پروژه به وجود آورد. الگوریتم ASO در زمانی که شرایط پروژه چندان غیر قابل پیش‌بینی نباشد با مقدار دهی مناسب این دو پارامتر می‌تواند آنها را بی‌اثر نماید. در ادامه فرآیند این لایه و پس از محاسبه مقدار نهایی تخمین تمامی پروژه‌های قسمت آموزش، بررسی می‌شود که آیا پروژه دیگری از مجموعه آموزش باقی مانده است یا خیر؟ در صوتی که پروژه دیگری باقی نمانده باشد مقدار دقت مدل پیشنهادی برای مجموعه پروژه‌های قسمت آموزش محاسبه شده و به عنوان بازخورد به الگوریتم ASO ارسال

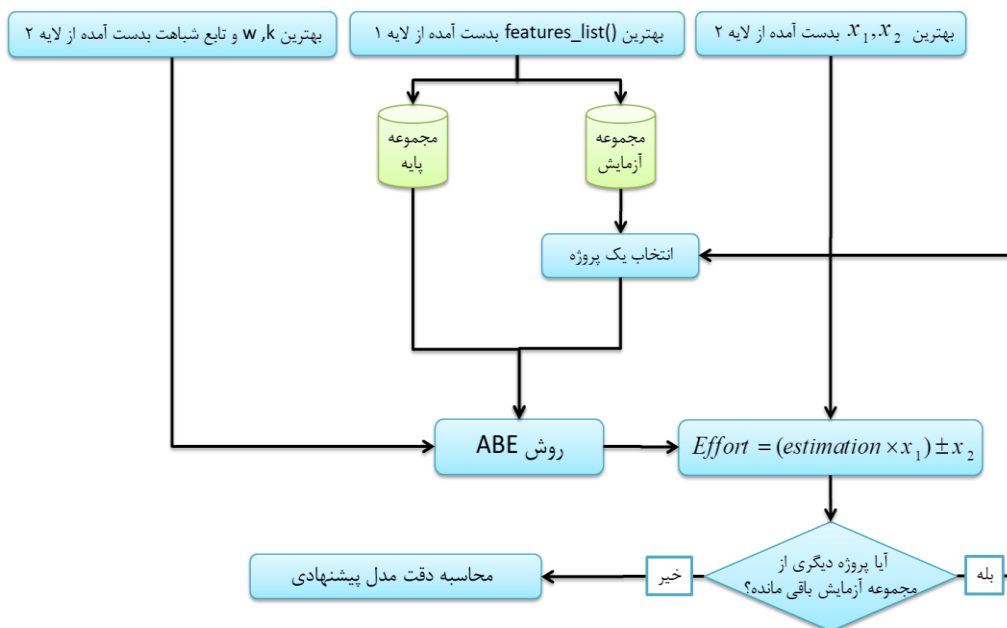
<sup>1</sup> Leave One Out

مرحله آزمایش، دقیق تر بودن و پایدارتر بودن نتایج حاصل از این روش است [۴۴].

عنوان آموزش و تنها پروژه باقی مانده به عنوان آزمایش در نظر گرفته می شود. علت استفاده از روش LOO برای



شکل ۲ - معماری لایه ۲ فاز آموزش

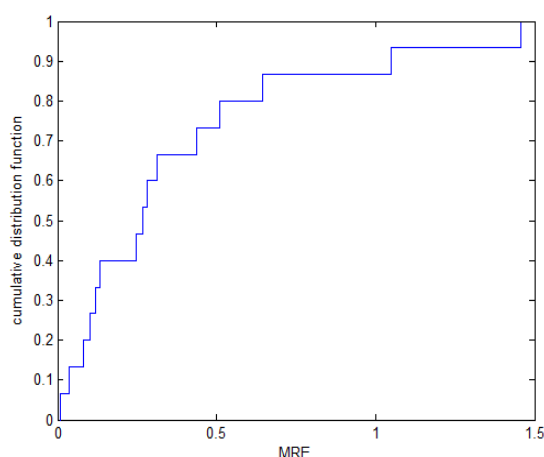


شکل ۳ - معماری فاز آزمایش



گرفته است. در پیکرندی الگوریتم ASO، تعداد اتم‌ها ۵۰، تعداد حداکثر تکرارها ۱۰۰۰ بار، وزن عمقی ۵۰ و ضریب وزنی ۰.۲ در نظر گرفته شده است. این تنظیمات در نتیجه انجام آزمایشات متعدد انتخاب شده است. محاسبه و نمایش دقت مدل در آزمایشات انجام شده به کمک معیارها و معادلات معرفی شده در بخش ۳ انجام شده است.

به عنوان اولین آزمایش مجموعه داده kemerer انتخاب شده است. مشخصات این مجموعه داده در بخش ۶ معرفی شد. نمودار توزیع فراوانی MRE حاصل از اجرای مدل پیشنهادی، به صورت LOO در شکل (۴) آمده است. در این نمودار فراوانی پروژه‌ها با خطای مشخص، قابل مشاهده است. این نمودار بیان‌کننده درصد توزیع مقادیر مختلف خطای MRE است. بررسی نمودار شکل (۴) نشان می‌دهد که بیش از ۷۰٪ پروژه‌ها با خطای کمتر از ۰.۵ تخمین زده شده‌اند. درصد بسیار بالایی از خطاهای بدست آمده مقدار کوچکی دارند. این مسئله به شکل بسیار مناسبی در شکل (۴) و در قسمت پرسیب آن قابل مشاهده است. شیب بالای این نمودار در یک بخش، بیان‌کننده درصد بالای توزیع خطا در آن بازه است. از سوی دیگر هرچه نمودار به سمت خطاهای بزرگ پیش رفته است سطح شیب آن ملایم شده و حتی به صفر رسیده است. این مسئله بیان‌کننده اندک بودن تعداد پروژه‌های با تخمین بد است. در نتیجه انجام آزمایش این مجموعه داده، مقدار MdmRE برای این آزمایش ۰.۲۶ و مقدار PRED برابر با ۰.۴۶ و مقدار MMRE برابر با ۰.۳۷ بدست آمده است.



شکل ۴ - نمودار توزیع فراوانی خطای MRE برای پایگاه داده kemerer

آزمایش دیگر بر روی مجموعه داده maxwell انجام شده است. مشخصات این مجموعه داده در بخش ۶ معرفی شد.

## ۶- معرفی مجموعه داده‌ها

برای آزمایش مدل پیشنهادی از ۳ مجموعه داده kemerer، desharnais و maxwell استفاده شده است. داده‌های این مجموعه داده‌ها بر اساس پروژه‌های واقعی جمع‌آوری شده است. این مجموعه داده‌ها در پژوهش‌های زیادی مورد استفاده محققان بوده است. در جدول ۲ مجموعه داده‌های استفاده شده معرفی شده است. مجموعه داده desharnais شامل اطلاعات ۸۱ پروژه نرم افزار است. این مجموعه داده از «خانه نرم افزار کانادایی» جمع‌آوری شده است. در این مجموعه داده، پروژه‌های نرم افزار، توسط ۱۱ ویژگی شرح داده شده است. ویژگی وابسته cost بر مبنای ۱۰۰۰ نفر ساعت است. ۱۰ ویژگی‌های مستقل این مجموعه داده عبارتند از 'YearEnd'، 'ManagerExp'، 'TeamExp'، 'Duration'، 'Transactions'، 'Entities'، 'AdjFP'، 'RawFP'، 'AdjFactor' و 'Dev.Env'. با توجه به اینکه اطلاعات ۴ پروژه از ۸۱ پروژه این مجموعه داده در دسترس نیست و گم شده است در نتیجه آزمایشات با استفاده از ۷۷ پروژه باقی مانده انجام شده است. مجموعه داده kemerer شامل اطلاعات ۱۵ پروژه نرم افزار است. پروژه‌های نرم افزار در این مجموعه داده بوسیله ۵ ویژگی مستقل و یک ویژگی وابسته شرح داده شده‌اند. ویژگی‌های مستقل عبارتند از 'hardware'، 'Language'، 'KSLOC'، 'Duration'، 'RawFp' و 'AdjFP'. ویژگی وابسته cost است. ویژگی cost بر مبنای نفر ماه است. مجموعه داده maxwell شامل داده‌هایی در مورد ۶۲ پروژه نرم افزار واقعی است. یک ویژگی وابسته به نام Effort و ۲۵ ویژگی مستقل از ۱ تا ۲۵ در این مجموعه داده نمایه شده است.

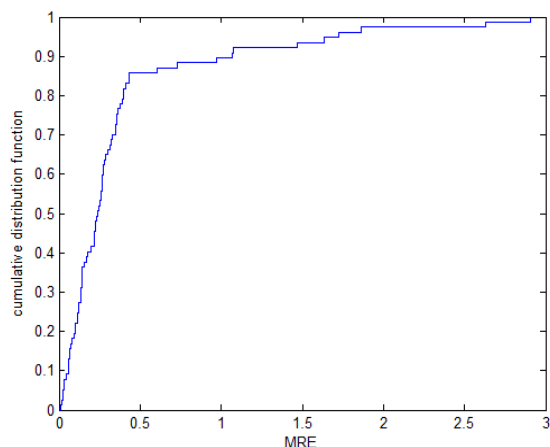
جدول ۲ - مجموعه داده‌ها

نام	تعداد نمونه	تعداد ویژگی	میان‌تلاش
kemerer	۱۵	۷	۲۱۹
desharnais	۷۷	۱۰	۴۷۹۵
maxwell	۶۲	۲۶	۸۲۲۳

## ۷- آزمایش مجموعه داده‌ها

در این قسمت مدل پیشنهادی مورد آزمایش قرار گرفته است. هدف از آزمایش این مدل، بررسی میزان دقت آن بوده است. آزمایش‌ها بر روی مجموعه داده‌های معرفی شده انجام شده است. نتایج این آزمایشات به تفکیک نوع مجموعه داده نمایش داده شده و مورد ارزیابی و تحلیل قرار

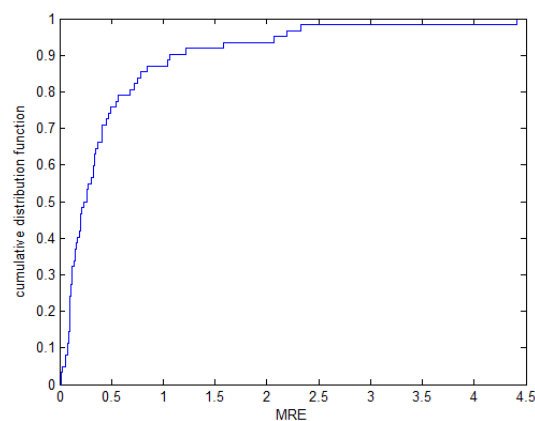
مورد آزمایش قرار گرفته است. نتایج بدست آمده نشان دهنده دقت بسیار بالای مدل پیشنهادی است. مقدار Pred مدل پیشنهادی، نشان دهنده نرخ بالای تخمین‌های دقیق این روش است. همچنین نتایج این آزمایشات به صورت مجزا بر روی هر مجموعه داده به صورت نمودار در شکل (۷)، (۸) و (۹) مقایسه شده است.



شکل ۶ - نمودار توزیع فراوانی خطای MRE برای پایگاه داده desharnaise

علت دقیق بودن نتایج مدل پیشنهادی، تفکیک کردن فیلترهای پالایش داده از پیکربندی ابزار است. در پژوهش‌های انجام شده در حوزه تخمین، اغلب در صورتی که از شیوه‌های پالایش داده استفاده شده است این شیوه‌ها بصورت ترکیب شده با فرآیند تخمین انجام می‌شود اما این مسئله مشکلات زیادی به همراه دارد. ترکیب کردن فرآیند پالایش و تخمین موجب می‌شود تا مدل با پیچیدگی فراوانی در مسئله مواجه باشد که این پیچیدگی موجب کاهش دقت نتایج خواهد شد. در پژوهش انجام شده در این مقاله نیز هنگام آزمایشات اولیه به شکل مرسوم اقدام به پالایش و تخمین همزمان انجام شد اما نتایج حاصل دقت بسیار پایینی داشت. اما هنگامی که مراحل به صورت لایه‌های مجزا طراحی و آزمایش شد، دقت نتایج به شکل چشمگیری بهبود پیدا کرد. از دلایل دیگر دقت بالای نتایج می‌توان به نقش پیش‌بینی‌کننده خطای تخمین توسط معادله ۹ اشاره کرد. استفاده از این معادله نقش بسیار مهمی در کاهش میزان MRE برای پروژه‌های با درصد خطای تخمین بالا داشته است. به عبارت ساده‌تر استفاده از معادله ۹ موجب شد تا حد بالای مقادیر MRE به شکل چشمگیری روند نزولی پیدا کرده و در محدوده قابل قبولی قرار گیرند.

مقدار MRE حاصل از اجرای مدل پیشنهادی، به صورت LOO در شکل (۵) آمده است. شکل (۵) نمودار توزیع فراوانی خطای MRE برای مجموعه داده maxwell است. همانطور که در شکل (۵) نمایش داده شده است درصد بسیار بالایی از خطاهای بدست آمده در بازه کمتر از ۰.۵ قرار دارد. این نکته در شکل (۵) کاملاً مشخص و قابل مشاهده است. بخش‌های پرشیب این نمودار مربوط به خطاهای کوچک است. ازسوی دیگر هر چه نمودار به سمت خطاهای بزرگ پیش رفته است سطح شیب آن ملایم شده و حتی به صفر رسیده است. این مسئله بیان‌کننده اندک بودن تعداد پروژه‌های با تخمین بد است. با آزمایش این مجموعه داده مقدار MdmRE برای این آزمایش ۰.۲۴ مقدار PRED برابر با ۰.۵۰ و مقدار MMRE برابر با ۰.۴۸ بدست آمده است.

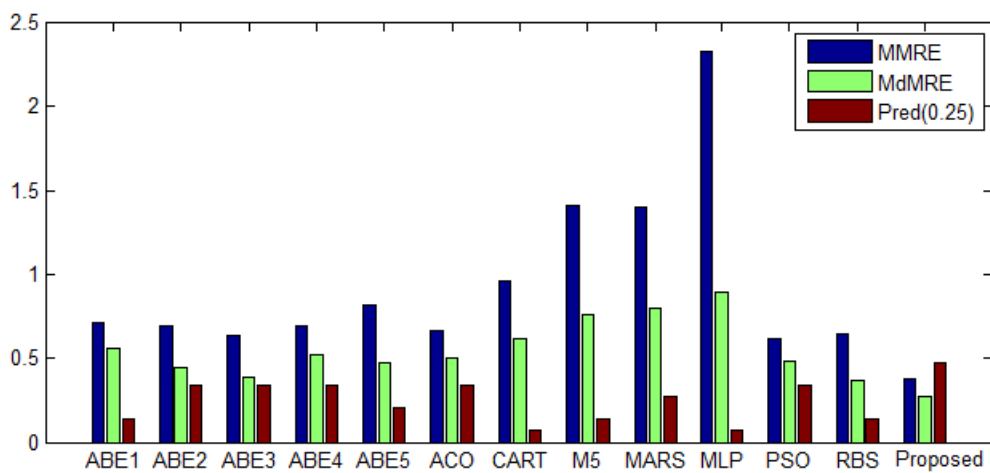


شکل ۵ - نمودار توزیع فراوانی خطای MRE برای پایگاه داده maxwell

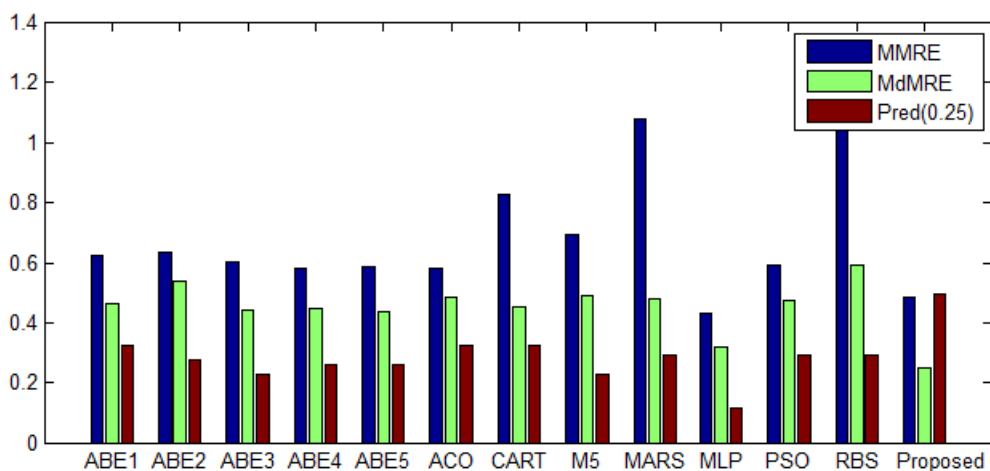
آزمایش بعدی بر روی مجموعه داده desharnaise انجام شده است. مشخصات این مجموعه داده در بخش ۶ معرفی شد. در شکل (۶) توزیع فراوانی خطای MRE برای مجموعه داده desharnaise نمایش داده شده است. همانطور که در این شکل می‌بینید حدود ۹۰٪ پروژه‌ها با دقت خطای کمتر از ۰.۵ تخمین زده شده‌اند و عملاً خطاهای بالا تعداد بسیار کم و به صورت موردی دارند. این نمودار نیز به شکل بسیار مناسبی دقت تخمین مدل پیشنهادی را نمایش می‌دهد. با آزمایش این مجموعه داده، مقدار MdmRE برای این آزمایش ۰.۲۳ مقدار PRED برابر با ۰.۵۲ و مقدار MMRE برابر با ۰.۳۹ بدست آمده است. نتایج بدست آمده توسط مدل پیشنهادی در جدول ۲ با سایر روش‌ها مقایسه شده است. نتایج روش‌های مختلف که در جدول ۳ آمده است به روش ارزیابی LOO

جدول ۳ - مقایسه نتایج آزمایشات بر اساس معیارهای MMRE، MdMRE و Pred(0.25) در مجموعه داده ها

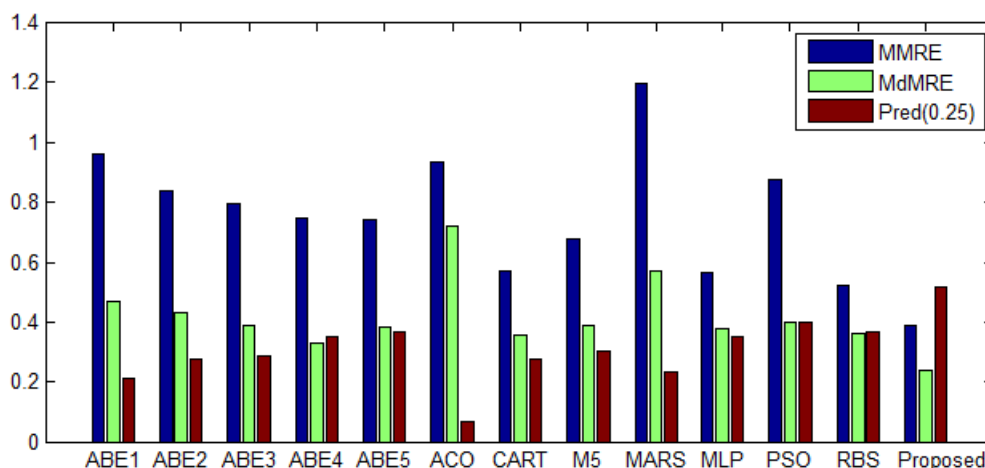
	مجموعه داده kemerer			مجموعه داده maxwell			مجموعه داده desharnaise		
	MMRE	MdMRE	Pred(0.25)	MMRE	MdMRE	Pred(0.25)	MMRE	MdMRE	Pred (0.25)
ABE (1)	۰.۷۱	۰.۵۵	۰.۱۳	۰.۶۲	۰.۴۶	۰.۳۲	۰.۹۵	۰.۴۷	۰.۲
ABE (2)	۰.۷	۰.۴۴	۰.۳۳	۰.۶۳	۰.۵۴	۰.۲۷	۰.۸۳	۰.۴۳	۰.۲۷
ABE (3)	۰.۶۲	۰.۳۸	۰.۳۳	۰.۶	۰.۴۴	۰.۲۲	۰.۷۹	۰.۳۸	۰.۲۸
ABE (4)	۰.۷	۰.۵۱	۰.۳۳	۰.۵۸	۰.۴۴	۰.۲۵	۰.۷۴	۰.۳۲	۰.۳۵
ABE (5)	۰.۸۲	۰.۴۶	۰.۲	۰.۵۸	۰.۴۳	۰.۲۵	۰.۷۴	۰.۳۸	۰.۳۶
M5	۱.۴	۰.۷۶	۰.۱۳	۰.۶۹	۰.۴۹	۰.۲۲	۰.۶۷	۰.۳۹	۰.۲۹
MLP	۲.۳۲	۰.۸۹	۰.۰۶	۰.۴۳	۰.۳۱	۰.۱۱	۰.۵۶	۰.۳۷	۰.۳۵
RBS	۰.۶۳	۰.۳۶	۰.۱۳	۱.۲۱	۰.۵۹	۰.۲۹	۰.۵۲	۰.۳۵	۰.۳۶
MARS	۱.۴	۰.۸	۰.۲۶	۱.۰۷	۰.۴۸	۰.۲۹	۱.۱۹	۰.۵۷	۰.۲۳
CART	۰.۹۶	۰.۶۱	۰.۰۶	۰.۸۲	۰.۴۵	۰.۳۲	۰.۵۷	۰.۳۵	۰.۲۷
ACO[24]	۰.۶۵	۰.۴۹	۰.۳۳	۰.۵۸	۰.۴۸	۰.۳۲	۰.۹۳	۰.۷۲	۰.۰۶
PSO[36]	۰.۶۱	۰.۴۷	۰.۳۳	۰.۵۹	۰.۴۷	۰.۲۹	۰.۸۷	۰.۴	۰.۴
Proposed model	۰.۳۷	۰.۲۶	۰.۴۶	۰.۴۸	۰.۲۴	۰.۵	۰.۳۹	۰.۲۳	۰.۵۲



شکل ۷ - مقایسه مدل پیشنهادی با سایر مدل‌ها در مجموعه داده kemerer



شکل ۸ - مقایسه مدل پیشنهادی با سایر مدل‌ها در مجموعه داده maxwell



شکل ۹ - مقایسه مدل پیشنهادی با سایر مدل‌ها در مجموعه داده desharnaise

حاصل از آزمایش wilcoxon روی MRE حاصل از هر روش در مقایسه با مدل پیشنهادی در مجموعه داده‌های مختلف نشان داده شده است. نتایج حاصل، نشان دهنده اثربخشی بالای مدل پیشنهادی در افزایش دقت تخمین است.

جدول ۳- P-value حاصل از آزمایش wilcoxon

	kemerer	maxwell	desharnaise
ABE (1)	۰.۰۰۵۶	۰.۰۱۷۲۵۴	۲.۲۸e-۰۳
ABE (2)	۰.۰۵۲۲	۰.۰۲۲۲۳۶	۰.۰۰۰۱۷۳۷
ABE (3)	۰.۰۵۲۲	۰.۰۰۲۱۷۰۱	۰.۰۰۰۳۶۶۷
ABE (4)	۰.۰۴۰۱	۰.۰۱۰۲۸۳	۰.۰۰۳۶۲۶۹
ABE (5)	۰.۰۱۰۱	۰.۰۱۰۸۹۱	۰.۰۰۶۷۹۸۷
M5	۰.۰۰۰۸	۰.۰۰۴۷۱۲۹	۰.۰۰۰۹۹۴۹۷
MLP	۹.۴۱۵۷e-۰۵	۰.۰۱۵۰۴۷	۰.۰۰۳۷۱۱۷
RBS	۰.۰۰۵۰	۰.۰۰۰۵۱۹۱۲	۰.۰۰۵۸۳۱
MARS	۰.۰۰۴۱	۰.۰۰۲۱۳۴۱	۷.۸۹e-۰۴
CART	۰.۰۰۲۰	۰.۰۳۴۷۳۸	۰.۰۰۱۲۳۵۷
ACO[24]	۰.۰۱۵۸	۰.۰۰۵۷۶۱۴	۵.۹e-۰۱۲
PSO[36]	۰.۰۲۰۴	۰.۰۱۵۶۸۲	۰.۰۰۹۲۷۴۸

برای بررسی دقیق‌تر نتایج مدل پیشنهادی از نمودار boxplot استفاده شده است. این نمودار محدوده MRE حاصل از آزمایش مدل پیشنهادی روی یک پایگاه داده را نشان می‌دهد. در این نمودار به آسانی محدوده و دقت MRE حاصل از مدل پیشنهادی با سایر مدل‌ها قابل مقایسه است. در شکل (۱۰)، (۱۱) و (۱۲) نمودار boxplot معیار MRE حاصل از آزمایش روش‌ها روی به ترتیب مجموعه داده‌های kemerer، maxwell و desharnaise نمایش داده شده است. بررسی این نمودارها نشان می‌دهد کمترین مقدار میانه و پایین‌ترین محدوده چهارک مربوط

در مراحل انجام آزمایشات بر روی مدل پیشنهادی یکی از مهمترین مواردی که مورد آزمایش قرار گرفت، بررسی ترتیب مناسب قرارگیری لایه‌ها بود. ترتیب پیشنهادی برای لایه‌ها در نتیجه انجام آزمایشات مختلف و جابجایی لایه‌ها بدست آمده است. البته نتایج آزمایشات و ترتیب بدست آمده کاملاً قابل توجه و پیش‌بینی بود. اگر ترتیب لایه‌ها را بررسی کنیم متوجه می‌شویم که لایه‌ای که اقدام به انتخاب ویژگی کرده است باید حتماً در ابتدا قرار گیرد زیرا در صورتی که ویژگی نامناسبی در اطلاعات مجموعه پروژه‌ها وجود داشته باشد، تأثیر بسیار بدی روی روش ABE و به خصوص تابع تعیین شباهت خواهد گذاشت. نتایج آزمایشات مختلف نیز تاییدکننده ترتیب قرارگیری لایه‌ها بوده است. برای مقایسه بهتر دقت مدل پیشنهادی با سایر روش‌ها، نتایج آزمایشات بر اساس معیارهای سه گانه به صورت نمودار نمایش داده شده‌اند. این نمودارها در شکل (۷)، (۸) و (۹) نمایش داده شده‌اند.

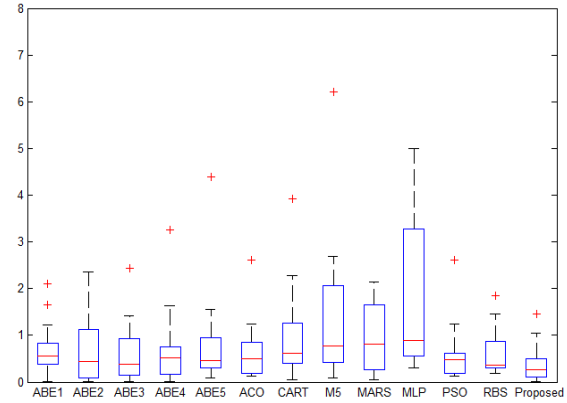
## ۸- تحلیل نتایج

برای بررسی میزان اثربخشی مدل پیشنهادی یک آزمایش آماری به نام wilcoxon انجام شد [۴۵]. آزمایش wilcoxon دو نمونه آماری را به عنوان ورودی دریافت می‌کند. خروجی این آزمایش یک عدد به نام P-value است که میزان تفاوت دو نمونه ورودی را نشان می‌دهد. مقادیر کوچک P-value نشان‌دهنده تفاوت بیشتر دو نمونه و مقادیر بزرگ آن نشان‌دهنده تشابه دو نمونه است. چنانچه مقدار P-value کمتر از ۰.۰۵ باشد به معنی تفاوت بالای دو نمونه است. در جدول ۳، نتایج آزمایش wilcoxon روی معیار MRE آمده است. در این جدول مقدار P-value

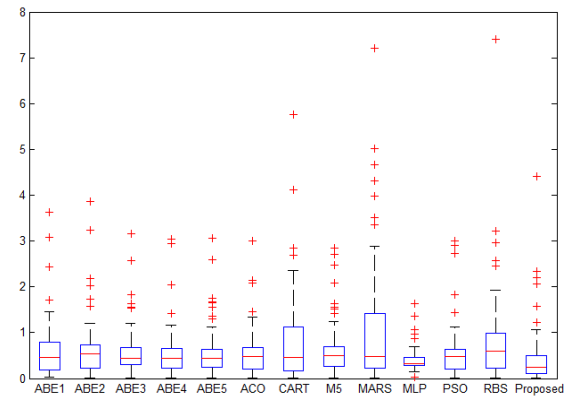
### ۹- نتیجه‌گیری

بر اساس ادبیات تحقیق، استفاده از شیوه‌های فراوری داده، شیوه‌های شناسایی ویژگی‌های موثر، شناسایی نوع ارتباط ویژگی‌ها پروژه بر تلاش مورد نیاز برای توسعه پروژه نرم‌افزار یا مدلسازی داده باعث افزایش دقت تخمین می‌شود. همچنین بکارگیری صحیح الگوریتم‌های هیوریستیک برای پیکربندی روش‌ها و ابزارها، نقش موثری در افزایش کارایی این روش‌ها و ابزارها داشته است. یکی از نوآوری‌های این مقاله، ارائه زیر مدل‌هایی با مقاصد ذکر شده، برای شناسایی ویژگی‌ها و ارتباطات موثر بین آنها، پیکربندی دقیق شیوه‌های مدلسازی داده و تخمین مبتنی بر تشابه ویژگی با کمک الگوریتم ASO است. نوآوری دیگر این مقاله پیشنهاد یک مدل متشکل از ۲ لایه است که در آن زیر مدل‌ها را به روشی در لایه‌های خود سازماندهی کرده است که باعث تقویت دقت یکدیگر شدند. لایه اول این مدل روی ویژگی‌های پروژه کار میکند. در لایه دوم، روش ABE که یک روش تخمین مبتنی بر تشابه ویژگی است با استفاده از ASO پیکربندی شد. بدلیل استفاده لایه دوم از نتایج تحلیل لایه اول روی ویژگی‌های پروژه، دقت این لایه افزایش یافت. همچنین در لایه دوم با استفاده از معادله ۹، خطای تخمین مدل مبتنی بر روش ABE و الگوریتم ASO کاهش پیدا کرد. معادله ۹ با کمک پارامترهای که بوسیله الگوریتم ASO پیکربندی می‌شود امکان همپوشانی مدل پیشنهادی با شرایط پروژه را افزایش داده است. آزمایش هر یک از این لایه‌ها به تنهایی موجب افزایش اندک دقت تخمین شد اما سازماندهی مناسب آنها موجب افزایش چشمگیر دقت نهایی شد. همچنین استفاده از الگوریتم هیوریستیک در این مدل، موجب افزایش انعطاف پذیری لایه‌ها و انطباق پذیری بیشتر آن با شرایط پروژه شد. در ادامه این مقاله، مدل ارائه شده مورد آزمایش قرار گرفت و نتایج بسیار دقیق آن نشان داده شد. دقت نتایج بدست آمده نشان می‌دهد که می‌توان بسیاری از معماری‌هایی را که تا کنون توسط محققان ارائه شده است را با باز طراحی بر مبنای دیدگاه مطرح شده در این پژوهش دقیق‌تر کرد. ایده ارائه شده در این پژوهش میتواند در تقویت دقت و اثر بخشی روش‌های آتی تاثیر بسزایی داشته باشد.

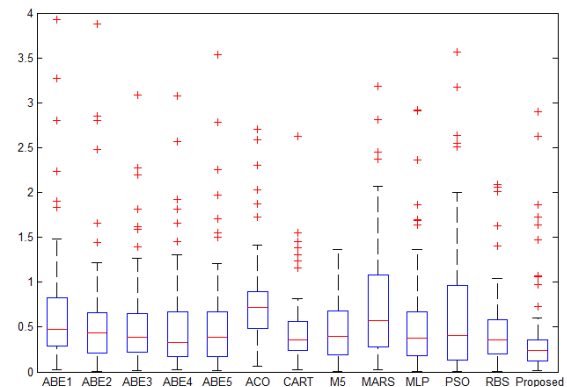
به روش پیشنهادی است. برخی از روش‌های مقایسه شده که توانسته‌اند محدوده‌های خوبی را به خود اختصاص دهند، در سایر مجموعه داده‌ها قادر به تکرار نتایج خوب خود نبوده‌اند. این نتایج پایداری دقت بدست آمده توسط مدل پیشنهادی را تأیید می‌کند.



شکل ۱۰ - box plot معیار MRE حاصل از آزمایش مدل پیشنهادی روی پایگاه داده kemerer



شکل ۱۱ - box plot معیار MRE حاصل از آزمایش مدل پیشنهادی روی پایگاه داده maxwell



شکل ۱۲ - box plot معیار MRE حاصل از آزمایش مدل پیشنهادی روی پایگاه داده desharnaise

## مراجع

- [1] Dominguez, J., "The CHAOS report 2009 on IT project failure," *Project Management Hut*, 2009.
- [2] Nelson, E. A., "Management handbook for the estimation of computer programming costs," SYSTEM DEVELOPMENT CORP SANTA MONICA CA1967.
- [3] Boehm, B. W., "Software engineering economics," in *Software pioneers*, ed: Springer, 2002, pp. 641-686.
- [4] Boehm, B. W., Abts, C., Brown, A. W., Chulani, S., Clark, B. K., Horowitz, E., et al., *Software cost estimation with COCOMO II*: Prentice Hall Press, 2009.
- [5] Albrecht, A. J. and Gaffney, J. E., "Software function, source lines of code, and development effort prediction: a software science validation," *IEEE transactions on software engineering*, 1983, pp. 639-648.
- [6] Abbas, S. A., Liao, X., Rehman, A. U., Azam, A., and Abdullah, M., "Cost estimation: A survey of well-known historic cost estimation techniques," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, 2012, pp. 612-636.
- [7] Finnie, G. R., Wittig, G. E., and Desharnais, J.-M., "A comparison of software effort estimation techniques: Using function points with neural networks, case-based reasoning and regression models," *Journal of systems and software*, vol. 39, 1997, pp. 281-289.
- [8] Sentas, P., Angelis, L., Stamelos, I., and Bleris, G., "Software productivity and effort prediction with ordinal regression," *Information and software technology*, vol. 47, 2005, pp. 17-29.
- [9] Briand, L. C., El Emam, K., Surmann, D., Wiczorek, I., and Maxwell, K. D., "An assessment and comparison of common software cost estimation modeling techniques," in *Proceedings of the 1999 International Conference on Software Engineering (IEEE Cat. No. 99CB37002)*, 1999, pp. 313-323.
- [10] Briand, L. C., Langley, T., and Wiczorek, I., "A replicated assessment and comparison of common software cost modeling techniques," in *Proceedings of the 22nd international conference on Software engineering*, 2000, pp. 377-386.
- [11] Kad, S. and Chopra, V., "Software development effort estimation using soft computing," *International Journal of Machine Learning and Computing*, vol. 2, 2012, p. 548.
- [12] Bardsiri, V. K., Jawawi, D. N. A., Bardsiri, A. K., and Khatibi, E., "LMES: A localized multi-estimator model to estimate software development effort," *Engineering Applications of Artificial Intelligence*, vol. 26, 2013, pp. 2624-2640.
- [13] Attarzadeh, I. and Ow, S. H., "A novel algorithmic cost estimation model based on soft computing technique," *Journal of computer science*, vol. 6, 2010, p. 117.
- [14] Idri, A., Hosni, M., and Abran, A., "Improved estimation of software development effort using classical and fuzzy analogy ensembles," *Applied Soft Computing*, vol. 49, 2016, pp. 990-1019.
- [15] Zare, F., Zare, H. K., and Fallahnezhad, M. S., "Software effort estimation based on the optimal Bayesian belief network," *Applied Soft Computing*, vol. 49, 2016, pp. 968-980.
- [16] Moradbeiky, A. and Bardsiri, A. K., "A New Architecture Based on Artificial Neural Network and PSO Algorithm for Estimating Software Development Effort," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 9, 2017, pp. 13-17.
- [17] Karimi, A. and Gandomani, T. J., "Software development effort estimation modeling using a combination of fuzzy-neural network and differential evolution algorithm," *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 11, 2021.
- [18] Chhabra, S. and Singh, H., "Optimizing design of fuzzy model for software cost estimation using particle swarm optimization algorithm," *International Journal of Computational Intelligence and Applications*, vol. 19, 2020, pp. 2050005.
- [19] Kumar, P. S., Behera, H. S., Kumari, A., Nayak, J., and Naik, B., "Advancement from neural networks to deep learning in software effort estimation: Perspective of two decades," *Computer Science Review*, vol. 38, 2020, pp. 100288.

- [۲۰] عمید خطیبی بردسیری، سیدمحسن هاشمی و محمدرضا رزازی، "ارائه یک مدل جدید جهت تخمین تلاش لازم برای توسعه سرویس های نرم افزاری"، مدل سازی در مهندسی، دوره ۱۵، شماره ۴۹، تابستان ۱۳۹۶، صفحه ۲۴۵-۲۶۱.
- [۲۱] صبا بیرانوند و محمدعلی زارع چاهوکی، "مروری بر روش های تخمین هزینه نرم افزار مبتنی بر یادگیری ماشین"، محاسبات نرم، دوره ۵، شماره ۱، تابستان ۱۳۹۵، صفحه ۳۶-۶۵.
- [22] Sharma, A. and Chaudhary, N., "Analysis of Software Effort Estimation Based on Story Point and Lines of Code using Machine Learning," *International Journal Of Computing and Digital System*, 2021.
- [23] Setiadi, A., Hidayat, W. F., Sinnun, A., Setiawan, A., Faisal, M., and Alamsyah, D. P., "Analyze the Datasets of Software Effort Estimation With Particle Swarm Optimization," in *2021 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2021, pp. 197-201.
- [24] Ranichandra, S., "Optimizing non-orthogonal space distance using ACO in software cost estimation," *Mukt Shabd J*, vol. 9, 2020, pp. 1592-1604.
- [25] Khan, B., Naseem, R., Binsawad, M., Khan, M., and Ahmad, A., "Software Cost Estimation Using Flower Pollination Algorithm," *Journal of Internet Technology*, vol. 21, 2020, pp. 1243-1251.
- [۲۶] مزده صباغ نژاد و عمید خطیبی بردسیری، "ارائه یک مدل ترکیبی جهت افزایش دقت روش استدلال مبتنی بر رویداد در برآورد تلاش توسعه نرم افزار"، مدل سازی در مهندسی، دوره ۱۶، شماره ۵۴، پائیز ۱۳۹۷، صفحه ۳۹-۵۲.
- [27] Bilgaiyan, S., Mishra, S., and Das, M., "Effort estimation in agile software development using experimental validation of neural network models," *International Journal of Information Technology*, vol. 11, 2019, pp. 569-573.
- [28] Rankovic, D., Rankovic, N., Ivanovic, M., and Lazic, L., "Convergence rate of Artificial Neural Networks for estimation in software development projects," *Information and Software Technology*, 2021, pp. 106627.
- [29] Araujo, R. d. A., Oliveira, A. L., and Meira, S., "A class of hybrid multilayer perceptrons for software development effort estimation problems," *Expert Systems with Applications*, vol. 90, 2017, pp. 1-12.
- [30] Moosavi, S. H. S. and Bardsiri, V. K., "Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation," *Engineering Applications of Artificial Intelligence*, vol. 60, 2017, pp. 1-15.
- [۳۱] سمیه شعبانی زاده رابری، وحید خطیبی بردسیری، و عمید خطیبی بردسیری، "ارائه یک روش نوین جهت پیش بینی نقص نرم افزار با استفاده از ترکیب شبکه عصبی و الگوریتم ملخ"، مدل سازی در مهندسی، دوره ۱۷، شماره ۵۷، تابستان ۱۳۹۸، صفحه ۲۰۱-۲۱۴.
- [32] Ali, A. and Gravino, C., "Improving software effort estimation using bio-inspired algorithms to select relevant features: An empirical study," *Science of Computer Programming*, vol. 205, 2021, pp. 102621.
- [33] Goyal, S. and Bhatia, P. K., "Feature selection technique for effective software effort estimation using multi-layer perceptrons," in *Proceedings of ICETIT 2019*, ed: Springer, 2020, pp. 183-194.
- [34] Phannachitta, P., "On an optimal analogy-based software effort estimation," *Information and Software Technology*, vol. 125, 2020, pp. 106330.
- [35] Chiu, N.-H. and Huang, S.-J., "The adjusted analogy-based software effort estimation based on similarity distances," *Journal of Systems and Software*, vol. 80, 2007, pp. 628-640.
- [36] Bardsiri, V. K., Jawawi, D. N. A., Hashim, S. Z. M., and Khatibi, E., "A PSO-based model to increase the accuracy of software development effort estimation," *Software Quality Journal*, vol. 21, 2013, pp. 501-526.
- [37] Resmi, V., Vijayalakshmi, S., and Chandrabose, R. S., "An effective software project effort estimation system using optimal firefly algorithm," *Cluster Computing*, vol. 22, 2019, pp. 11329-11338.
- [38] Vijay, J. F., "Enrichment of accurate software effort estimation using fuzzy-based function point analysis in business data analytics," *Neural Computing and Applications*, vol. 31, 2019, pp. 1633-1639.
- [39] Kaur, I., Narula, G. S., Wason, R., Jain, V., and Baliyan, A., "Neuro fuzzy—COCOMO II model for software cost estimation," *International Journal of Information Technology*, vol. 10, 2018, pp. 181-187.
- [40] García-Florian, A., López-Martín, C., Yáñez-Márquez, C., and Abran, A., "Support vector regression for predicting software enhancement effort," *Information and Software Technology*, vol. 97, 2018, pp. 99-109.

- [41] Puspaningrum, A. and Sarno, R., "A hybrid cuckoo optimization and harmony search algorithm for software cost estimation," *Procedia Computer Science*, vol. 124, 2017, pp. 461-469.
- [42] Singh, S. P., Singh, V. P., and Mehta, A. K., "Differential evolution using homeostasis adaption based mutation operator and its application for software cost estimation," *Journal of King Saud University-Computer and Information Sciences*, 2018.
- [43] Idri, A., Abran, A., and Khoshgoftaar, T. M., "Estimating software project effort by analogy based on linguistic values," in *Proceedings Eighth IEEE Symposium on Software Metrics*, 2002, pp. 21-30.
- [44] Kocaguneli, E. and Menzies, T., "Software effort models should be assessed via leave-one-out validation," *Journal of Systems and Software*, vol. 86, 2013, pp. 1879-1890.
- [45] Wilcoxon, F., "Individual comparisons by ranking methods," in *Breakthroughs in statistics*, ed: Springer, 1992, pp. 196-202.