فصلنامه سیستم‌های پردازشی و ارتباطی چندرسانه‌ای هوشمند
Intelligent Multimedia Processing and Communication Systems (IMPCS)

**Summer, 2022, 3(2) 11-18**
**DOR: 20.1001.1.27832570.1401.3.2.2.2**

**Received: 07 May 2022**
**Accepted: 25 June 2022**

**مقاله پژوهشی**

# Comparison of the Classification Methods in Software Development Effort Estimation

**Sadegh Ansaripour [1], Taghi Javdani Gandomani [2*]**

1.MSc Student, Department of Computer Engineering, Aghigh University, Shahin shahr, Isfahan, Iran.
2.Assistant Professor, Department of Computer Science, Shahrekord University, Shahrekord, Iran. *(Corresponding Author)*
javdani@sku.ac.ir

**Abstract**

**Introduction:** The main goal of software companies is to provide solutions in various fields to better meet the needs of customers. The process of successful modeling depends on finding the right and accurate requirements. However, the key to successful development for adapting and integrating different developed parts is the importance of selecting and prioritizing the requirements that will advance the workflow and ultimately lead to the creation of a quality product. Validation is the key part of the work, which includes techniques that confirm the accuracy of a set of requirements for building a solution that leads to the project's business objectives. Requirements change during the project, and managing these changes is important to ensure the accuracy of the software built for stakeholders. In this research, we will discuss the process of checking and validating the software requirements.

**Method**: Requirement extraction is conducted by means of discovery, review, documentation, and understanding of user needs and limitations of a system. The results are presented in the form of products such as text requirements descriptions, use cases, processing diagrams, and user interface prototypes.

**Findings:** Data mining and recommender systems can be used to increase the necessary needs, however, another method. of social networks and joint filtering can be used to create requirements for large projects to identify needs.

**Discussion**: In the area of product development, requirements engineering approaches focus exclusively on requirement development. There are challenges in the development process due to the existence of human resources. If the challenges are not seen well at this stage, it will be extremely expensive after the software production. Therefore, in this regard, errors should be minimized and they should be identified and corrected as soon as possible. Now, with the investigations carried out, one of the key issues in the field of requirements is the discussion of validation, which first confirms that the requirements are able to be implemented in a set of characteristics according to the system description, and secondly, a set of essential characteristics. such as complete, consistent, according to standard criteria, non-contradiction of requirements, absence of technical errors, and lack of ambiguity in requirements. In fact, the purpose of validation is to ensure the result that a sustainable and renewable product is created according to the requirements.

**Keywords:** Software effort estimation, Data mining, Machine learning, Classification, Software engineering.

فصلنامه سیستم‌های پردازشی و ارتباطی چندرسانه‌ای هوشمند
Intelligent Multimedia Processing and Communication Systems (IMPCS)
انجمن علمی تجارت الکترونیکی ایران

# مقایسه روش‌های طبقه‌بندی در تخمین تلاش توسعه نرم‌افزار

تاریخ دریافت: ۱۴۰۱/۰۲/۱۷ سال سوم، تابستان ۱۴۰۱
تاریخ پذیرش: ۱۴۰۱/۰۴/۰۴ شماره دوم ، صص: ۱۱-۱۸

**صادق انصاری‌پور[۱]، تقی جاودانی گندمانی[۲*]**

۱. دانشجوی کارشناسی‌ارشد، گروه مهندسی کامپیوتر، دانشگاه غیرانتفاعی عقیق، شاهین‌شهر، ایران. asnsariour565@gmail.com
۲. استادیار، گروه علوم کامپیوتر، دانشگاه شهرکرد، شهرکرد، ایران. ( نویسنده مسئول) javdani@sku.ac.ir

**چکیده:** نادر ست بودن تخمین هزینه نرم‌افزار یکی از دلایل مهم ناامیدی متخ ص صان نرم‌افزار و محققان تخمین هزینه ا ست. علی‌رغم تلاش‌های فراوانی که برای بهبود تخمین تلاش یا هزینه انجام‌شـــده‌اســـت اما هنوز هم دقت تخمین پایین اســـت. عدم تجزیه و تحلیل مناســـب در ابتدای شـــروع به کار پروژه و همچنین عدم به‌روزرســـانی آن در حین انجام پروژه یکی از مهم‌ترین دلایل شـــکســـت پروژه‌ها محســـوب می‌شـــود. اگرچه زمانی که یک پروژه نرم‌افزاری نهایی می‌شـــود، بازخوردهای آن ایجاد می‌شـــود، اما اگر تخمین‌ها و واقعیات ثبت شده با پروژه انجام شده به‌طور کامل مطابقت نداشته‌باشند، آنگاه نمی‌توان تخمین دقیقی انتظار داشت. بنابراین جمع‌آوری داده‌های پروژه بر ا ساس ویژگی‌های م شخص امری ضروری ا ست و در اینجا ست که می‌توان به نقش پررنگ پروژه‌های انجام شده در گذ شته و مجموعه داده‌هایی که می‌توان با اســـتفاده از آن‌ها ایجادنمود، پی‌برد. در این مطالعه ســـعی بر این اســـت که به بررســـی نقش روش‌های مختلف داده‌کاوی در تخمین تلاش نرم‌افزار بپردازیم و از این دیدگاه به بررســـی روند بهبود دقت در این زمینه بپردازیم. نتایج حاصـــل از تحقیق نشان‌می‌دهند که روش‌های طبقه‌بندی می‌توانند نتایج بهتری نسبت به روش‌های رگرسیونی و خوشه‌بندی به‌دست‌آورند و تعداد زیادی از محققین سعی‌دارند که با روش‌های ترکیبی بتوانند بهبود بیشتری در این زمینه به‌دست‌آورند.

**واژه‌های کلیدی:** تخمین تلاش نرم‌افزار، داده‌کاوی، یادگیری ماشین، طبقه‌بندی، مهندسی نرم‌افزار.

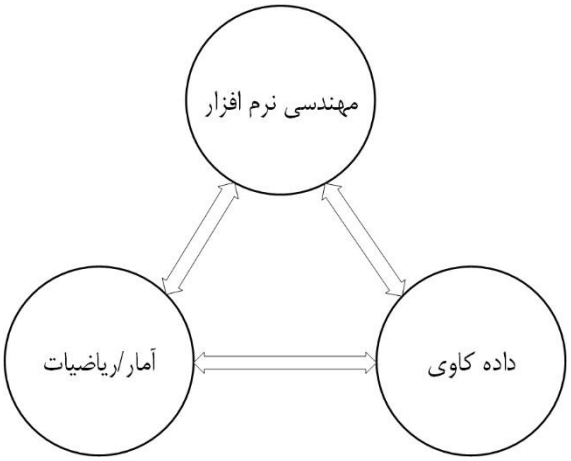
## ۱. مقدمه

تخمین هزینه در نرم‌افزار برای یک سازمان امری ضروری است. از آنجا که تلاش انسانی مهم‌ترین عاملی است که در تخمین هزینه‌های نرم‌افزار می‌توان به آن ا شاره‌نمود، بنابراین در مطالعات مختلف تلاش و هزینه به‌جای یکدیگر مورد استفاده قرارمی‌گیرند. به‌طور کلی بحث تخمین برای تمامی ذی‌نفعان یک پروژه از اهمیت بسیار بالایی برخوردار است زیرا تخمین دقیق از یک سو می‌تواند به مشتری محتمل‌ترین زمان تحویل نرم‌افزار را اطلاع‌دهد و از سوی دیگر باعث می شود که تیم تو سعه برنامه‌ای را بر طبق تخمین صورت‌گرفته برای تو سعه نرم‌افزار ارائه‌دهد. بر ا ساس یک نظر سنجی که تو سط مؤ سسه مدیریت پروژه[۱] (PMI) در سال ۲۰۱۷ انجام‌شد، ۵۷ درصد از پروژه‌ها در بودجه تعیین‌شده، ۵۱ درصد پروژه ها به‌موقع انجام‌یافته و ۱۴ درصد از پروژه‌ها شکست خورده‌بودند [۱].

تخمین هزینه‌ها در نرم‌افزار یکی از ضروری‌ترین پیش‌بینی‌هایی است که برای یک محصول نرم‌افزاری وجوددارد. مدیران پروژه‌های نرم‌افزاری برای انجام فعالیت های خود به تکنیک هایی قابل‌اطمینان نیازدارد زیرا تخمین بیش از حد نیاز باعث ازد ست‌رفتن درآمد سازمان و همچنین ازبین‌رفتن منابع و اتلاف زمان می شود. این در حالی ا ست که تخمین کمتر از حد نیاز مستلزم ساعات کار طولانی اعضای تیم تو سعه برای تکمیل پروژه وعده‌داده شده، ا ست که می‌تواند باعث ایجاد پروژه‌ای با کیفیت پایین و یا تأخیر در تحویل پروژه نرم‌افزاری شود.

با توجه به تبعات جدی که در رابطه با تخمین بیش از حد یا کمتر از حد نیاز برای یک سازمان وجوددارد، تجزیه و تحلیل تخمین کمک می‌کند تا میزان هزینه و همچنین تلاش مورد نیاز برای توسعه یک محصول نرم‌افزاری را بهتر پیش‌بینی‌کنیم.

سازمان‌ها در طول فعالیت‌های توسعه و نگهداری نرم‌افزار[۲]، داده های زیادی را تولیدمی‌نمایند. داده‌کاوی امکان کشف دانش مفید و الگوهای پنهان از داده‌های مهندسی نرم‌افزار را فراهم‌می‌کند. ریاضی توابع ابتدایی را فراهم‌می‌کند و آمار احتمال، روابط و همبستگی را در داده های جمع‌آوری‌شده، تعیین‌می‌کند. در این میان علم داده[۳] که برآیندی از ر شته‌های مختلف مانند داده‌کاوی، مهند سی نرم‌افزار و آمار است، بسیار مورد توجه قرارمی‌گیرد.




**شکل ۱: ارتباط داده‌کاوی، مهندسی نرم‌افزار و آمار**

داده‌کاوی فرایند استخراج مقادیر زیادی داده از پایگاه‌های داده و یافتن الگو های جالب در میان آن ها است. تاکنون از تکنیک های داده‌کاوی زیادی به‌منظور تخمین تلاش استفاده‌شده‌است و می‌توان گفت این تکنیک‌ها نقش مهمی در تخمین تلاش نرم‌افزار ایفامی‌کنند. شکل ۱ مرتبط‌بودن سه حوزه اصلی داده‌کاوی، مهندسی نرم‌افزار و آمار / ریاضیات را نشان‌می‌دهد. از جمله مهم‌ترین روش‌های داده‌کاوی می‌توان رگر سیون، خو شه‌بندی، طبقه‌بندی و کاوش قوانین انجمنی[۴] را نام‌برد.

در ادامه ، مبانی تخمین تلاش نرم‌افزار و تکنیک‌های طبقه‌بندی به‌طور خلاصه در بخش ۲ بررسی‌می‌شود. در بخش سوم کارهای مرتبط انجام‌شده بررسی‌خواهدشد. در بخش ۴ بحث و بررسی در مورد روش‌های ذکرشده بیان‌می‌شود. بخش ۵ نیز به نتیجه‌گیری می پردازد.

## ۲. مبانی پژوهش

در این بخش به ا صول تخمین تلاش نرم‌افزار و همچنین نحوه کارکرد روش‌های داده‌کاوی مورد مطالعه می‌پردازیم.

### ۲,۱. تخمین تلاش نرم‌افزار

در زمینه تخمین هزینه نرم‌افزار، تلاش مورد نیاز برای توسعه پروژه جدید با درنظرگرفتن عوامل مؤثر بر توسعه پروژه برآورد می‌شود. سپس پروژه جدید با داده‌های تاریخی پروژه‌های گذشته که با موفقیت اجرا شده‌اند، مقایسه‌می شود که شامل اندازه‌گیری معیارها و تلاش تو سعه مرتبط است.

برای مدل‌های تخمین تلاش متغیر وابسته میزان تلاش کلی است، در حالی که طیف و سیعی از متغیرهای م ستقل برای مدل‌های تخمین تلاش وجوددارد. برخی از متغیرهای مستقل در بیشتر مدل‌های تخمین تلاش رایج هستند و در برخی متفاوت می‌باشند. اما به‌طور کلی در طول توسعه یک نرم‌افزار دو عامل اصلی وجوددارد اولین عامل اندازه می‌باشد که بارزترین ویژگی نرم‌افزار است و نیاز به نیروی انسانی را هدا یت‌می‌کند و دیگری بهره‌وری نرم‌افزار است که مستقیم یا غیرمستقیم تأثیرگذار است.

- اندازه یک نرم‌افزار نشان‌دهنده حجم کاری است که باید تحویل داده شود. می‌توان آن را از نظر تعداد خطوط کد[۵] (LOC) یا از نظر عملکرد درنظرگرفت. اندازه عملکرد[۶] یک نرم‌افزار را می‌توان برحسب معیارهای مختلفی مانند نقاط عملکردی[۷] (FP) و نقاط استفاده کاربردی[۸] (UCP) اندازه‌گیری‌کرد.
- بهره‌وری نرم‌افزار را می‌توان با نسبت وا حد های خروجی به واحدهای ورودی تعریف‌کرد که در آن خروجی‌ها کد منبع و اسناد هستند و ورودی‌ها تلاش‌هایی هستند که برای تولید خروجی و توسعه محصول صرف‌می‌شود[۲]. از آنجا که طیف عظیمی از عوامل وجوددارد که ممکن است بر بهره‌وری تأثیر بگذارد، بسیار ضروری ا ست که مرتبط‌ترین عوامل را از بین همه عوامل در نظر

بگیریم. این عوامل ممکن اســت مختص مدل، حوزه یا ســازمان خاص باشــند. در این راســتا، ترندویز و همکاران یک نمای کلی جامع و بســیار خوب از عوامل مرتبط با بهره‌وری ارائه‌کرده‌اند که به‌وفور توسط متخصصان نرم‌افزار استفاده‌می‌شود[۳].

اگر S را اندازه نرم‌افزار و FI را عواملی که بر بهره‌وری توســعه نرم‌افزار تأثیردارند، در نظر بگیریم، آنگاه تلاش به‌صــورت زیر محاســبه می‌شود.

$$Effort = f(S.FI) \qquad (۱)$$

FI مجموعه‌ای از همه$FI_i$ها که به‌صورت مجموعه $FI = \{FI_i : i = 1 \cdots n\}$ تعریف‌می شود و مسئول تعریف محیط بهره‌وری یک ســازمان، مدل یا دامنه می‌باشــد. به‌طور تجربی مشــخص شده‌است که بین بهره‌وری پروژه‌های جدید و ارتقاء داده شده، تفاوت معناداری وجوددارد[۴, ۵]. بنابراین، پیشنهاد شده‌است که پروژه‌های جدید باید به‌طور متفاوتی نســبت به پروژه‌های نوع ارتقایافته برای اهداف تخمین نرم‌افزاری، تحلیل‌شوند [۶].

### ۲,۲. نحوه کار روش‌های طبقه‌بندی

به‌طور کلی به داده کاوی فرآیند اســتخراج داده از منابع مختلف گفتــه‌می‌شــود. برای این منظور از تکنیــک‌هــای داده‌کــاوی زیــادی استفاده‌می شود. در بسیاری از موارد داده‌های جمع‌آوری شده ماهیت خام دارد و ممکن است مقیاس‌های مختلفی داشته‌باشد. برای دستیابی به کارایی بالا، باید مجموعه داده خام را پیش‌پردازش کنیم تا به یک مقیاس مشترک برسیم. در حیطه تخمین تلاش از تکنیک پیش‌پردازش نرمال‌سازی حداقل-حداکثر استفاده‌می شود. این تکنیک مقادیر داده را در بازه [۰،۱] قرارمی‌دهد به گونه‌ای که به کمترین مقدار ۰ و بالاترین مقدار ۱ اختصــاص داده‌می‌شــود. در مرحله بعد، تکنیک‌های داده‌کاوی مانند طبقه‌بندی، خوشه‌بندی، و کاوش قوانین انجمنی اعمال‌می شوند تا الگوهای مفید و روابط در داده‌های مهندسی نرم‌افزار کشف شوند. در نهایت، تکنیک های اعتبارســنجی برای ارزیابی نتایج داده کاوی اعمال‌می‌شــود که در این حوزه از روش اعتبارســنجی k-fold cross validation بیشتر استفاده‌می‌شود.

## ۳. تخمین تلاش با استفاده از داده‌کاوی

تحقیقات و مطالعات مروری زیادی در زمینه تخمین تلاش نرم‌افزار انجام شده‌است. علی و گراوینو[۹] طی یک سیستماتیک مطالعات انجام شده در سال‌های ۱۹۹۱ تا ۲۰۱۷ را که از تکنیک‌های یادگیری ماشین در تخمین تلاش نرم‌افزاری نرم‌افزاری اســتفاده‌نمودند مورد بررســی قراردادند. بر اســاس این مقاله شــبکه عصــبی مصــنوعی[۱۰] (ANN) و ماشــین بردار پشــتیبان[۱۱] (SVM) دو تکنیکی هســتند که در مطالعات بهتر از ســایر تکنیک‌های یادگیری عمل‌کرده‌اند [۷]. ون و همکاران[۱۲] [۸] در یک نظرســنجی مطالعات تخمین تلاش را که بر تکنیک‌های یادگیری ماشین متمرکز بود، تجزیه و تحلیل کرد و آن‌ها را با مطالعات متمرکز بر تکنیک‌های غیریادگیری ماشین مقایسه‌کردند. بر اساس این نظرســنجی، اســتدلال مبتنی بر مورد[۱۳](CBR)[۹] و شــبکه عصــبی مصنوعی (ANN) پرکاربردترین تکنیک‌ها بودند. به‌دلیل استفاده زیاد از شــبکه‌های عصــبی در این حیطه دیو و دوتا[۱۴]در یک مقاله مروری مطــالعــاتی را کــه فقط بر روی شــبکــه عصــبی تمرکز دارنــد، بررسی‌کردند[۱۰].

طیف گســترده‌ای از روش های تخمین تلاش توســعه نرم‌افزار وجوددارد که قبلاً پیشنهاد شده‌اند و تقریباً از نیم قرن گذشته دائماً در حال تحقیق هســتند. در طول این مطالعه، روش‌های مختلف را جهت به‌کارگیری تکنیک‌های داده‌کاوی و یادگیری ماشین، بررسی و مقایسه خواهیم‌کرد. در ادامه در جدول ۱ مطالعاتی که در آن از تکنیک های داده‌کاوی و یادگیری ماشین استفاده‌شده، نشان‌داده‌شده‌است.

**جدول ۱: مطالعات انجام‌شده با استفاده از داده‌کاوی و یادگیری ماشین**

| مرجع | سال | وظیفه داده کاوی | هدف | الگوریتمها |
|---|---|---|---|---|
| [۱۱] | ۲۰۰۴ | رگرسیون | ارزیابی عملکرد CF نسبت به روش رگرسیون سنتی | SWR, ANN |
| [۱۲] | ۲۰۰۶ | رگرسیون | مقایسه SVR، رگرسیون خطی و شبکه های عصبی RBF | SVR,LR, RBF |
| [۱۳] | ۲۰۰۸ | رگرسیون | مجموعه شبکه های عصبی با حافظه انجمنی(ENNA) | NN, MLP, KNN |
| [۱۴] | ۲۰۰۹ | رگرسیون | مجموعه ای از شبکه های عصبی با حافظه انجمنی (ENNA) را پیشنهاد شده است. | NN, MLP, KNN |
| [۱۵] | ۲۰۱۰ | رگرسیون | کاربرد ARR و تشخیص چند خطی برای بهبود دقت مدل رگرسیون | OLS,RR |
| [۱۶] | ۲۰۱۱ | رگرسیون | برای نشان دادن اثربخشی SVR برایSEE | SVR, RBF |
| [۱۷] | ۲۰۱۱ | رگرسیون | برای ارزیابی اینکه آیا روش‌های گروهی در دسترس،SEE را افزایش می‌دهند یا خیر | MLP, RBF, RT |
| [۱۸] | ۲۰۱۲ | رگرسیون | استفاده از مدل‌های بین شرکتی برای ایجاد مجموعه‌های متنوعی که بتوانند به صورت پویا با تغییرات سازگار شوند | WC RTs, CC-DWM |
| [۱۹] | ۲۰۱۲ | رگرسیون | برای تولید تخمین در یادگیری جمعی از روش های پیش بینی چندگانه داده کاوی استفاده نموده است | CART, NN, LR, PCR, PLSR, SWR, ABE0-1NN, ABE0-5NN |
| [۲۰] | ۲۰۱۲ | طبقه بندی رگرسیون | استفاده از تکنیک های داده کاوی برای تخمین تلاش نرم افزار | M5, CART, LR, MARS, MLPNN, RBFNN, SVM |
| [۲۱] | ۲۰۱۳ | رگرسیون | برای نشان دادن رفتار اقدامات در SEE و ایجاد گروه های خوب | MLP, RBF, REPTree, |
| [۲۲] | ۲۰۱۳ | طبقه بندی | ارائه الگوریتم جدید QUICK برای یافتن محتوای ضروری از داده‌های تخمین | RR, ABE |

| | | | | |
|---|---|---|---|---|
| [۲۳] | ۲۰۱۳ | خوشه بندی طبقه بندی | تخمین تلاش برای توسعه نرم افزار | NN, ABE, C-means |
| [۲۴] | ۲۰۱۴ | رگرسیون | بهبود مدل COCOMO با استفاده از شبکه های عصبی مصنوعی مورد بررسی قرار می گیرند | MLP, RBFNN, SVM, PSO-SVM Extreme learning Machines |
| [۲۵] | ۲۰۱۵ | رگرسیون | برای نمایش تأثیر تکنیک های پیش پردازش داده ها بر روش های ML درSEE | CBR, ANN, CART Preprocessing rech: MDT, LD, MI, FS, CS, FSS, BSS BR |
| [۲۶] | ۲۰۱۵ | رگرسیون | سفارشی سازی رگرسیون بیزی و الگوریتم EM برای SDEE | ABE |
| [۲۷] | ۲۰۱۵ | طبقه بندی | طبقه بندی انتخابی با در نظر گرفتن ماهیت پروژه ها برای بهبود دقت ABE | ANN, CART, SWR, MLR |
| [۲۸] | ۲۰۱۵ | طبقه بندی | تعیین وزن هر ویژگی پروژه | MLP, RBFNN, GRNN, CCNN |
| [۲۹] | ۲۰۱۶ | رگرسیون | چهار مدل شبکه عصبی با یکدیگر مقایسه می شوند. | GA and PSO |
| [۳۰] | ۲۰۱۶ | رگرسیون | ارائه مدلی مبتنی بر شبکه بیزی | SVM, RBNN |
| [۳۱] | ۲۰۱۶ | طبقه بندی رگرسیون | یک مدل ترکیبی با استفاده از SVM و RBNN در مقایسه با مدل‌های قبلی | SVM, KNN |
| [۳۲] | ۲۰۱۷ | طبقه بندی | برای تخمین تلاش نرم افزار با استفاده از تکنیک هایML | constant output, Sugeno with linear output |
| [۳۳] | ۲۰۱۹ | رگرسیون | طراحی و مقایسه سه مدل منطق فازی مختلف برای پیش‌بینی تلاش نرم‌افزاری | Random Forest ،REPTree ، SMOReg ،M5Rule وMLP |
| [۳۴] | ۲۰۲۰ | طبقه بندی رگرسیون | بهینه سازی ازدحام ذرات به عنوان یک الگوریتم انتخاب ویژگی برای روش های مختلف یادگیری ماشین | firefly algorithm, black hole optimization, and genetic algorithm |
| [۳۵] | ۲۰۲۲ | طبقه بندی | تخمین تلاش نرم‌افزاری با مجموعه وزنی الگوریتم‌های مبتنی بر جستجوی ترکیبی | |

## ۴. بحث و بررسی

چندین مطالعه روش های مختلف طبقه‌بندی را از طریق روش های مختلف طبقه‌بندی و رگرسیونی مقایسه و بررسی کرده‌اند. زیرا این دو دسته روش در میان روش‌های داده‌کاوی موجود بیشترین طرفدار را دارند و دلیل آن وجود مجموعه داده هایی است که در این حوزه وجوددارد.در این مجموعه داده‌ها به‌دلیل ساختاری که دارند، به‌راحتی می‌توان روش های طبقه‌بندی و رگرسیونی را پیاده‌کرد؛ همچنین دسترسی آسان و رایگان به مجموعه داده‌ها از دیگر دلایل رایج‌بودن استفاده از این روش‌هاست.

روش‌های طبقه‌بندی تاکنون به طرق مختلف بررسی شده‌اند. مثلاً برخی استفاده از روش‌ها را به دو دسته یادگیری منفرد و جمعی تقسیم‌بندی و از روش‌های bagging و boosting استفاده‌کردند[۱۷, ۲۱]. برخی مجموعه داده‌ها را تغییرداده و برای تخمین از داده‌های بین شرکتی یا درون شرکتی استفاده‌کردند[۱۸].

در این مقاله سعی‌شد تا تحقیقات انجام‌شده از طریق روش‌های رگرسیونی و طبقه‌بندی مورد بررسی قرارگیرد. لازم به ذکر است که در زیرمجموعه روش‌های داده‌کاوی، روش‌های خوشه‌بندی نیز وجوددارد اما به‌دلیل اینکه در تخمین تلاش توسعه نرم‌افزار، کمتر مورد توجه بوده، در این مقاله از بررسی آن‌ها صرف‌نظر شده‌است.

در این راستا دیجایگر و همکاران[۱۵] به بررسی هر دو روش با یکدیگر پرداختند. ایشان در یک مطالعه با مقیاس بزرگ انواع مختلفی از تکنیک‌ها از جمله تکنیک‌های القای مدل‌های مبتنی بر درخت/قانون مانند M5 و CART، مدل‌های خطی مانند انواع مختلف رگرسیون خطی، مدل‌های غیرخطی (MARS)، شبکه‌های عصبی پرسپترون چندلایه، شبکه‌های تابع پایه شعاعی، و حداقل مربعات). ماشین‌های بردار پشتیبان)، و تکنیک های تخمینی که به‌صراحت مدلی را القانمی‌کنند (به عنوان مثال، رویکرد استدلال مبتنی بر مورد) را بررسی کردند. ایشان در نهایت دریافتند که رگرسیون حداقل مربعات معمولی در ترکیب با تبدیل لگاریتمی بهترین عملکرد را خواهدداشت[۲۰]. در مطالعه دیگری در سال ۲۰۲۰، محققین به بررسی ویژگی‌ها در سه مجموعه داده آلبرشت، دشارنایس، کوکومو و ناسا پرداختند زیرا اعتقاد داشتند مجموعه داده‌ها اغلب شامل ویژگی‌هایی هستند که همه آن‌ها برای طبقه‌بندی مفید نیست و سعی‌کردند با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات، ویژگی‌های تکراری و همچنین ویژگی‌هایی که باعث افزایش سوءاستفاده از عملکرد برخی ویژگی‌ها می‌شد را کاهش‌دهند. در نهایت آن ها از الگوریتم های Random Forest، REPTree، SMOReg، Linear Regression، M5Rule و MLP استفاده‌کردند و در کنار آن روش bagging نیز پیاده‌سازی کردند. نتایج حاصل از تحقیق آن‌ها نشان‌داد که روش M5Rule با کمترین مقدار MMRE توانسته‌است بهترین نتایج را تولیدنماید[۳۴].

در را ستای بررسی روش‌های رگرسیونی نا صیف و همکاران[۱۶] به مقایسه شبکه‌های عصبی مختلف پرداخته و دریافتند که پیش‌پردازش داده‌ها از مهمترین مسائلی است که قبل از هر کاری باید به ان پرداخت زیرا داده های ازدست‌رفته[۱۷] و داده های پرت که اغلب در مجموعه داده‌های آموزشی وجوددارند تأثیر منفی روی دارند و این مسأله تخمین نادرست را درپی‌خواهدداشت[۲۹].

یکی دیگر از تحقیقاتی که در سال‌های اخیر انجام‌شده‌است استفاده از منطق فازی رگرسیونی است که در آن برای پیش‌بینی تخمین تلاش نرم‌افزار از سه مدل منطق فازی مختلف ا ستفاده شد که عبارتند از ممدانی، سوگنو با خروجی ثابت و سوگنو با خروجی خطی؛ در واقع ایشان برای کمک به طراحی مدل های منطق فازی از تحلیل

رگرسیونی بهره‌بردند. ایشان دریافتند که ناهمگن‌بودن داده ها می‌تواند عملکرد مدل را تا حد زیادی تحت‌تأثیر قراردهد زیرا اکثر مدل‌های فازی به داده‌های پرت حساسند[۳۳].

اخیراً توجه محققان به استفاده از الگوریتم‌های بهینه‌سازی در هوش ازدحامی جلب‌شده و تحقیقات زیادی در این زمینه منتشر شده‌است که عموماً از مدل هزینه ساختاری (COCOMO) مبتنی‌بر تشابه (ABE) و روش‌های یادگیری ماشین استفاده‌کرده‌اند[۳۶-۳۸].

با نگاهی به جدول فوق می‌توان دریافت که دقت به‌دست‌آمده از روش‌های مختلف بر اساس اهداف و یا ساختار پروژه تعریف شده متغیر است. به همین دلیل در مطالعات مختلف گروه‌بندی‌های متفاوتی برای مطالعات ارائه شده‌است. با توجه به مطالعات انجام‌شده در سال‌های اخیر روش جدیدی که منجر به تغییرات اساسی در این حوزه تحقیقاتی شود به‌وجود نیامده‌است و تمامی مقالات سعی در بهبود روش‌های پایه داشته‌اند. علاوه بر این در بسیاری از مقالات از روش‌های ترکیبی استفاده شده‌است؛ درواقع در این مقالات سعی شده‌است که بتوانند با ترکیبی مناسب از الگوریتم‌ها، دقت را بهبود بخشند[۳۹]. همچنین در این مطالعه می‌توان دریافت که شبکه‌های عصبی مختلف معمولاً به‌عنوان طبقه‌بندی اصلی در اکثر تحقیقات در تخمین تلاش نرم‌افزار مورد استفاده قرارگرفته‌است.

## ۵. نتیجه‌گیری

با توجه به رشد صنعت نرم‌افزار، نیاز است که روش‌های مختلف تخمین تلاش توسعه نرم‌افزار نیز خود را با این مسأله تطبیق‌دهند. زیرا در حوزه تخمین تلاش نرم‌افزار باید پیش‌بینی‌کنیم که در آینده به چه تعداد نیروی کار و زمان نیاز داریم. اگر تخمین‌ها و واقعیات ثبت‌شده با پروژه انجام‌شده مطابقت کامل نداشته‌باشند، آنگاه نمی‌توان تخمین دقیقی انتظار داشت و در اینجاست که می‌توان به نقش پررنگ پروژه‌های انجام-شده در گذشته و مجموعه داده‌هایی که می‌توان با استفاده از آن‌ها ایجاد نمود پی‌برد. درواقع هرقدر با دقت بیشتری تخمین را انجام‌دهیم احتمال موفقیت پروژه بالاتر می‌رود. یکی از دیدگاه‌هایی که می‌توان روش‌ها و مطالعات مختلف این حیطه را با آن مورد بررسی و ارزیابی قرارداد، روش-های داده‌کاوی است. در این مقاله دریافتیم که مطالعات زیادی با استفاده از روش‌های داده‌کاوی و به‌خصوص طبقه‌بندی و رگرسیونی انجام‌شده است. همچنین درطی تحقیقات دریافتیم که از روش‌های خوشه‌بندی به‌ندرت استفاده‌شده‌است؛ اما با توجه به این که روش‌های خوشه‌بندی توانسته‌اند در بسیاری از دیگر حوزه‌های مهندسی نرم‌افزار عملکرد خوبی داشته‌باشند، می توان استفاده از خوشه‌بندی در این زمینه را جزء مسائل باز دانست که می‌توان در آینده تمرکز بیشتری بر آن داشت. همچنین به‌نظرمی‌رسد که در سال‌های اخیر استفاده از الگوریتم‌های فراابتکاری بیشتر موردبررسی‌قرارگرفته و نتایج خوبی نیز به‌دست‌آورده‌اند. همچنین بهره‌گیری از ساختارهای ترکیبی راه‌حل دیگری است که اخیراً زیاد به آن پرداخته‌شده‌است و در اکثر آن‌ها یک یا دو الگوریتم فراابتکاری با روش‌های طبقه‌بندی ترکیب‌شده و دقت بهتری را تولیدکرده‌اند.

## مراجع


[۱] P. M. Institute, *Success rates rise: Transforming the high cost of low performance*, 2017.

[۲] I. S. -. "IEEE Standard for Software Productivity Metrics.," *1993:1045*, 1992.

[۳] A. Trendowicz, and J. Münch, "Factors influencing software development productivity—state-of-the-art and industrial experiences," *Advances in computers*, vol. 77, pp. 185-241, 2009.

[۴] D. Rodríguez, M. Sicilia, E. García, and R. Harrison, "Empirical findings on team size and productivity in software development," *Journal of Systems and Software*, vol. 85, no. 3, pp. 562-570, 2012.

[۵] F. González-Ladrón-de-Guevara, M. Fernández-Diego, and C. Lokan, "The usage of ISBSG data fields in software effort estimation: A systematic mapping study," *Journal of Systems and Software*, vol. 113 ,pp. 188-215, 2016.

[۶] A. García-Floriano, C. López-Martín, C. Yáñez-Márquez, and A. Abran, "Support vector regression for predicting software enhancement effort," *Information and Software Technology*, vol. 97, pp. 99-109, 2018.

[۷] A. Ali, and C. Gravino, "A systematic literature review of software effort prediction using machine learning methods," *Journal of software: evolution and process*, vol. 31, no. 10, pp. e2211, 2019.

[۸] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, "Systematic literature review of machine learning based software development effort estimation models," *Information and Software Technology*, vol. 54, no. 1, pp. 41-59, 2012.

[۹] M. Halkidi, D. Spinellis, G. Tsatsaronis, and M. Vazirgiannis, "Data mining in software engineering," *Intelligent Data Analysis*, vol. 15, no. 3, pp. 413-441, 2011.

[۱۰] V. S. Dave, and K. Dutta, "Neural network based models for software effort estimation: a review," *Artificial Intelligence Review*, vol. 42, no. 2, pp. 295-307, 2014.

[۱۱] N. Ohsugi, M. Tsunoda, A .Monden, and K.-i. Matsumoto, "Effort estimation based on collaborative filtering." pp. 274-286.

[۱۲] A. L. Oliveira, "Estimation of software project effort with support vector regression," *Neurocomputing*, vol. 69, no. 13-15, pp. 1749-1753, 2006.

[۱۳] Y .Kultur, B. Turhan, and A. B. Bener, "ENNA: software effort estimation using ensemble of neural networks with associative memory." pp. 330-338.

[۱۴] Y. Kultur, B. Turhan, and A. Bener, "Ensemble of neural networks with associative memory (ENNA) for estimating software development costs," *Knowledge-Based Systems*, vol. 22, no. 6, pp. 395-402, 2009.

[۱۵] Y.-F. Li, M. Xie, and T.-N. Goh, "Adaptive ridge regression system for software cost estimating on multi-collinear datasets," *Journal of Systems and Software*, vol. 83, no. 11, pp. 2332-2343, 2010.

[۱۶] A. Corazza, S. Di Martino, F. Ferrucci, C. Gravino, and E. Mendes, "Investigating the use of support vector

regression for web effort estimation," *Empirical Software Engineering,* vol. 16, no. 2, pp. 211-243, 2.۰۱۱

[۱۷] L. L. Minku, and X. Yao, "A principled evaluation of ensembles of learning machines for software effort estimation." pp. 1-10.

[۱۸] L. L. Minku, and X. Yao, "Can cross-company data improve performance in software effort estimation?." pp. 69-78.

[۱۹] E. Kocaguneli, T. Menzies, and J. W. Keung, "On the value of ensemble effort estimation," *IEEE Transactions on Software Engineering,* vol. 38, no. 6, pp. 1403-1416, 2011.

[۲۰] K. Dejaeger, W. Verbeke, D. Martens, and B. Baesens, "Data mining techniques for software effort estimation: a comparative study," *IEEE transactions on software engineering,* vol. 38, no. 2, pp. 375-397, 2011.

[۲۱] L. L. Minku, and X. Yao, "Software effort estimation as a multiobjective learning problem," *ACM Transactions on Software Engineering and Methodology (TOSEM),* vol. 22, no. 4, pp. 1-32, 2013.

[۲۲] E. Kocaguneli, T. Menzies, J. Keung, D. Cok, and R. Madachy, "Active learning and effort estimation: Finding the essential content of software effort estimation data," *IEEE Transactions on software engineering,* vol. 39, no. 8, pp. 1040-1053, 2012.

[۲۳] V. Khatibi, D. N. Jawawi, and E. Khatibi, "Increasing the accuracy of analogy based software development effort estimation using neural networks," *International Journal of Computer and Communication Engineering,* vol. 2, no. 1, pp. 78, 2013.

[۲۴] P. Subitsha, and J. K. Rajan, "Artificial neural network models for software effort estimation," *International journal of technology enhancements and emerging engineering research,* vol. 2 ,no. 4, pp. 76-80, 2014.

[۲۵] J. Huang, Y.-F. Li, and M. Xie, "An empirical analysis of data preprocessing for machine learning-based software cost estimation," *Information and software Technology,* vol. 67, pp. 108-127, 2015.

[۲۶] W. Zhang, Y. Yang, and Q. Wang, "Using Bayesian regression and EM algorithm with missing handling for software effort prediction," *Information and software technology,* vol. 58, pp. 58-70, 2015.

[۲۷] V. Khatibi Bardsiri, and E. Khatibi, "Insightful analogy-based software development effort estimation through selective classification and localization," *Innovations in Systems and Software Engineering,* vol. 11, no. 1, pp. 25-38, 2015.

[۲۸] E. Khatibi, and V. Khatibi Bardsiri, "Model to estimate the software development effort based on in-depth analysis of project attributes," *IET software,* vol. 9, no. 4, pp. 109-118, 2015.

[۲۹] A. B. Nassif, M. Azzeh, L. F. Capretz, and D. Ho, "Neural network models for software development effort estimation: a comparative study," *Neural Computing and Applications,* vol. 27, no. 8, pp. 2369-2381, 2016.

[۳۰] F. Zare, H. K. Zare, and M. S. Fallahnezhad, "Software effort estimation based on the optimal Bayesian belief network," *Applied Soft Computing,* vol. 49, pp. 968-980, 2016.

[۳۱] M. Azzeh, and A. B .Nassif, "A hybrid model for estimating software project effort from Use Case Points," *Applied Soft Computing,* vol. 49, pp. 981-989, 2016.

[۳۲] V. Gopinath, and R. Menon, "Software Effort Prediction-A Datamining Approach," *Journal of Network and Information Security Volume,* vol. 5, no. 01, 2017.

[۳۳] A. B. Nassif, M. Azzeh, A. Idri, and A. Abran, "Software development effort estimation using regression fuzzy models," *Computational intelligence and neuroscience,* vol. 2019, 2019.

[۳۴] M. Z. Khan, "Particle swarm optimisation based feature selection for software effort prediction using supervised machine learning and ensemble methods: A comparative study," *Invertis Journal of Science & Technology,* vol. 13, no. 1, pp. 33-50, 2020.

[۳۵] W. Rhmann, B. Pandey ,and G. A. Ansari, "Software effort estimation using ensemble of hybrid search-based algorithms based on metaheuristic algorithms," *Innovations in Systems and Software Engineering,* vol. 18, no. 2, pp. 309-319, 2022.

[۳۶] V. Resmi, S. Vijayalakshmi, and R .S. Chandrabose, "An effective software project effort estimation system using optimal firefly algorithm," *Cluster Computing,* vol. 22, no. 5, pp. 11329-11338, 2019.

[۳۷] A. A. Fadhil, R. G. Alsarraj, and A. M. Altaie, "Software cost estimation based on Dolphin algorithm," *IEEE Access,* vol. 8, pp. 75279-75287, 2020.

[۳۸] M. Dashti, T. J. Gandomani, D. H. Adeh, H. Zulzalil, and A. B. M. Sultan, "LEMABE: a novel framework to improve analogy-based software cost estimation using learnable evolution model," *PeerJ Computer Science,* vol. 7, pp. e800, 2022.

[۳۹] M. Dashti, and T. J. Gandomani, "A Taxonomy of Approaches and Methods for Software Effort Estimation," *Innovations in Computer Science and Engineering*, pp. 97-105: Springer, 2022

.

## پی‌نوشت

[۱] Project Management Institute
[۲] development and maintenance activities
[۳] Data science
[۴] association rule mining
[۵] lines of code
[۶] functional size
[۷] function points
[۸] use case points
[۹] Ali and Gravino
[۱۰] artificial neural network
[۱۱] support vector machine
[۱۲] Wen et al.
[۱۳] case-based reasoning
[۱۴] Dave and Dutta
[۱۵] Dejaeger
[۱۶] Nassif et al.
[۱۷] missing values