Scientific-Research Quarterly Journal of Technical Science Khatam Al-Nabieen University
Volume 2, Issue 2, Spring 2024

Software Project Management Using Machine Learning Technique — A Review

Mohammed Najah Mahdi, Mohd Hazli Mahammed Zabil, Abdul Rahim Ahmad, Roslan Ismail, Yunus Yusoff, Lim Kok Cheng, Muhammad Sufyian bin Mohd Azmi, Hayder Natiq and Hushalini Happala Naidu

Translated by Habibullah Haidari

Abstract
Project management planning and assessment are of great significance in project performance activities. Without a realistic and logical plan, it isn't easy to handle project management efficiently. This paper presents a wide-ranging comprehensive review of papers on the application of Machine Learning in software project management. Besides, this paper presents an extensive literature analysis of (1) machine learning, (2) software project management, and (3) techniques from three main libraries, Web Science, Science Directs, and IEEE Explore. One-hundred and eleven papers are divided into four categories in these three repositories. The first category contains research and survey papers on software project management. The second category includes papers that are based on machine-learning methods and strategies utilized on projects; the third category encompasses studies on the phases and tests that are the parameters used in machine-learning management and the final classes of the results from the study, contribution of studies in the production, and the promotion of machine-learning project prediction. Our contribution also offers a more comprehensive perspective and a context that would be important for potential work in project risk management. In conclusion, we have shown that project risk assessment by machine learning is more successful in minimizing the loss of the project, thereby increasing the likelihood of the project success, providing an alternative way to efficiently reduce the project failure probabilities, and increasing the output ratio for growth, and it also facilitates analysis on software fault prediction based on accuracy.

Keywords: machine learning technique; software project estimation; software estimation; software project management; project risk assessment

[Downloaded from jots.knu.edu.af on 2026-07-04]

فصلنامه علمی - تحقیقی علوم فنی
پوهنتون خاتم‌النبیین(ص)
سال دوم ، شماره دوم، بهار ۱۴۰۳
صفحه ۳ - ۶۹

مروری بر مدیریت پروژۀ نرم‌افزار
با استفاده از تکنیک یادگیری ماشین

محمد نجاح مهدی، محمد هزلی محمد زبیل، عبدالرحیم احمد، رُسلان اسماعیل، یونس یوسف، لیم کوک چِنگ، محمدسفیان بن محمد عظمی، حیدر ناطق و هوشالینی هاپالا نایدو

مترجم: حبیب الله حیدری[1]

چکیده

برنامه‌ریزی و ارزیابی مدیریت پروژه در فعالیت‌های عملکردی از اهمیت بالایی برخوردار است. بدون برنامه‌ای واقعی و منطقی، مدیریت پروژه به‌طور کارآمد، آسان نیست. این مقاله، مروری گسترده و جامع از مقالات مربوط به کاربرد یادگیری ماشین در مدیریت پروژۀ نرم‌افزاری ارائه می‌دهد؛ علاوه‌بر این، مقالۀ حاضر تجزیه‌وتحلیل ادبیات گسترده‌ای از یادگیری ماشین، مدیریت پروژۀ نرم‌افزار و تکنیک‌های سه کتابخانۀ اصلی Web of Science، Science Directs و IEEE Exploreرا ارائه می‌دهد. ۱۱۱ مقاله در این سه مخزن به چهار دسته تقسیم می‌شوند؛ دستۀ اول شامل مقالات تحقیقی و نظرسنجی در مورد مدیریت پروژۀ نرم‌افزاری است؛ دستۀ دوم شامل مقالاتی است که مبتنی‌بر روش‌ها و استراتژی‌های یادگیری ماشین هستند که در پروژه‌ها استفاده می‌شوند؛ دستۀ سوم شامل مطالعات مربوط به مراحل و آزمون‌هایی است که پارامترهای مورد استفاده در مدیریت یادگیری ماشین هستند و کلاس‌های نهایی نتایج حاصل از مطالعه، مشارکت مطالعات در تولید و ارتقای پیش‌بینی پروژه‌های یادگیری ماشین هستند. مشارکت ما همچنین دیدگاه جامع‌تری دارد و زمینه‌ای را ارائه می‌دهد که برای کار بالقوه در مدیریت ریسک پروژه مهم است. در نتیجه، ما نشان داده‌ایم که ارزیابی ریسک پروژه توسط یادگیری ماشین در به حداقل رساندن ضرر پروژه موفق‌تر است؛ بنابراین، احتمال موفقیت پروژه را افزایش می‌دهد و راهی جایگزین برای کاهش مؤثر احتمالات شکست پروژه و افزایش نسبت خروجی برای رشد ارائه می‌کند و همچنین تجزیه‌وتحلیل پیش‌بینی خطای نرم‌افزار را براساس دقت تسهیل می‌کند.

واژگان کلیدی: تکنیک یادگیری ماشین، برآورد پروژۀ نرم‌افزار، برآورد نرم‌افزار، مدیریت پروژۀ نرم‌افزار، ارزیابی ریسک پروژه.

---

[1] لیسانس علوم کامپیوتر، دانشگاه قم، قم، ایران

[Downloaded from jots.knu.edu.af on 2026-07-04]

۱. مقدمه

بهبود کارایی و حفظ پایداری پروژهٔ نرم‌افزاری از موانعی است که مدیران پروژه با آن روبرو هستند. احتمال شکست پروژه عموماً به‌دلیل کمبود دانش، مهارت، منابع و فناوری در حین اجرای پروژه است [۱-۳]. دانشی که از مجموعهٔ داده‌های پروژه‌های گذشته به دست می‌آید، می‌تواند برای توسعهٔ مدل‌های پیش‌بینی با استفاده از روش‌شناسی‌های ریاضی ازجمله رگرسیون خطی و مطالعهٔ روش‌های مرتبط با یادگیری ماشین (ML) مانند شبکهٔ مصنوعی شبکه (ANN)و ماشین‌های بردار پشتیبانی(SVM) روشی را ارائه دهند که برای پیش‌بینی آیندهٔ پروژه بر شواهد فعلی و گذشتهٔ آن متمرکز است. به‌دلیل زیاد بودن الگوریتم‌های مختلف ML، تعدادی از آن‌ها هنوز مورد مطالعه قرار نگرفته‌اند. با توجه به یافته‌های موارد مطالعه‌شده دراین مقاله، دلیل استفاده از پروژه‌های خودکار، مسائل ارزیابی مدیریت پروژه و روش‌شناسی توسعهٔ ML مطرح می‌شود. نتایج تجربی مورد ارزیابی قرار خواهند گرفت.

اگرچه ادبیات پروژه، موفقیت و شکست پروژه را توصیف می‌کند، اما بحث‌های طولانی در مورد چگونگی اندازه‌گیری پیشرفت پروژه را وجود دارد. درک عملکرد پروژه و ارزیابی موفقیت پروژه متفاوت است [٤]. هیوز[1] و سایر اعضا [٥] در مؤسسهٔ مدیریت پروژه (PMI) [٦] بین متغیرهای موفقیت پروژه و عملکرد آن تفاوت قائل می‌شوند.

آستانه‌های پیشرفت پروژه برای اندازه‌گیری موفقیت و شکست آن ارزیابی می‌شوند و بازخوردی برای پیشرفت پروژه در نظر گرفته می‌شود. از لحاظ تاریخی، ارائهٔ نتایج لازم و استفاده از منابع انتخاب‌شده توسط یک پروژه موفق در مدت پروژهٔ مشخص [۷] متمایز می‌شود [٦]. PMI ابتکاراتی را شناسایی می‌کند که به اهداف پروژه، معیارها و جاه‌طلبی‌های ذی‌نفعان با موفقیت دست می‌یابد. محققانی مانند کومو علادوانی[2] [۸]، کِیتس و ملاقاسمی[3] [۹]، پارسونز[4] [۱۰] و روزِنفلد[5] [۱۱] اثرات معیارهای کلاسیک هدف مانند هزینهٔ پروژه (بالاتر و کمتر از بودجه)، زمان پروژه (زود، دیر یا زود) و خروجی نتایج پروژه (با ویژگی‌ها و توابع کمتر یا بهتر) را شرح می‌دهند.

ارزیابی نیازمندی‌های پروژه به هزینه‌ها، هزینه‌های زمانی، اهداف محقق‌نشده یا حتی لغو پروژه‌ها کمک می‌کند و به خطر طبیعی و ناخواستهٔ پروژه و اثرات نامطلوب بر قابلیت اطمینان پروژه‌های نرم‌افزاری تبدیل می‌شود [۱۲]. الزامات اصلاح مشخصات (از نظر گسترش چندگانه، حذف و اصلاح) در طول پروژهٔ توسعهٔ نرم‌افزار ازجمله عوامل اصلی ایجاد مشکلات برای پروژه است [۱۳-۱۶].

[1] Hughes
[2] Como Aladwani
[3] Cates and Mollaghasemi
[4] Parsons
[5] Rosenfeld

[Downloaded from jots.knu.edu.af on 2026-07-04]

بخش دوم این مقاله شامل تجزیه‌وتحلیل توضیحی اصول ارزیابی پروژهٔ نرم‌افزاری و فناوری آموزش کامپیوتر است. ساختار این مقاله به این شرح است؛ بخش ۳ رویکرد و ازجمله منبع مطالب، الزامات واجد شرایط بودن تحقیق، بررسی ادبیات سیستماتیک (SLR) و اثرات نتایج جستجو از انتشارات را تعریف می‌کند. این بخش همچنین سؤالات تحقیق (RQs) را برای این پژوهش شناسایی می‌کند، تهدیدهایی که اعتبار آن‌ها را تهدید می‌کند و به چالش‌های عمده برای اثربخشی SLRها اشاره می‌کند. پرس‌وجوهای هر موضوع از سه وبسایت مقاله به چهار کلاس با طبقه‌بندی ادبی در مدیریت پروژهٔ نرم‌افزاری با استفاده از تکنیک ML تقسیم شده‌اند. بخش ٤ به الهامات، مشکلات و توصیه‌ها در آن حوزهٔ تحقیقاتی و رویکردی مدرن برای مدیریت ریسک پروژه‌های نرم‌افزاری می‌پردازد. در نهایت، بخش ٥ نتیجه‌گیری را ارائه می‌دهد.

۲. مطالعهٔ مقدماتی

در این بخش، برخی از مفاهیم تخمین پروژهٔ نرم‌افزاری و تکنیک یادگیری ماشین را بررسی و تصریح می‌کنیم.

۲-۱. تخمین تلاش نرم‌افزاری

پیش‌بینی تلاش و مدت‌زمان توسعهٔ نرم‌افزار وظیفه‌ای حیاتی برای مدیریت پروژهٔ نرم‌افزاری مؤثر (SPM)[1] است. دقت و قابلیت اطمینان مکانیسم‌های پیش‌بینی نیز ضروری هستند. داشتن ارزیابی دقیق تلاش به‌ویژه در مرحلهٔ اولیهٔ پروژهٔ نرم‌افزاری ممکن است به‌طور قابل توجهی خطرات بالایی را که محصولی نرم‌افزاری در طول توسعه متحمل می‌شود، کاهش دهد. متأسفانه بسیاری از تکنیک‌های برآورد موجود به‌طور اساسی اشتباه هستند و بیشتر پروژه‌ها با بیش از حد تلاش مواجه می‌شوند؛ بااین‌حال، مشخص شد که برآورد پروژهٔ نرم‌افزاری براساس الگوریتم‌های ML می‌تواند تخمین تلاش دقیق‌تری را ارائه دهد.

۲-۲. یادگیری ماشین (ML)

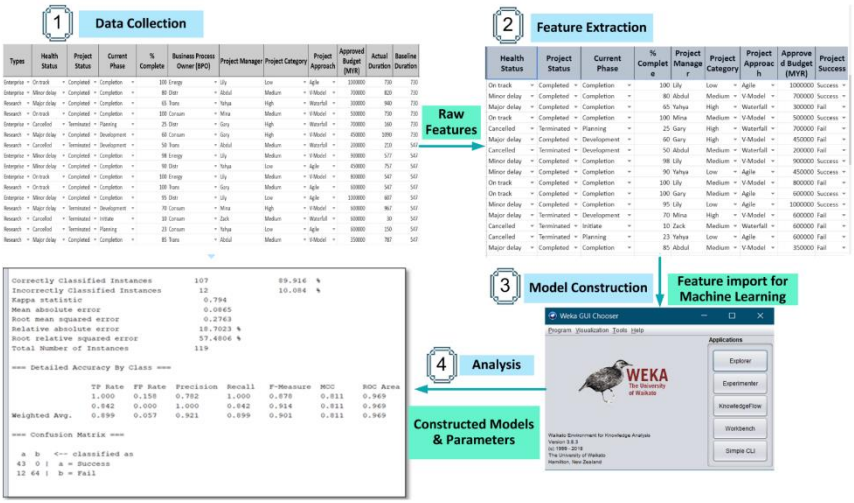
ML، برنامه‌ای کاربردی از هوش مصنوعی است که سیستم‌هایی را برای یادگیری و بهبود از تجربهٔ بدون برنامه‌ریزی صریح به‌طور خودکار فراهم می‌کند. به عبارت دیگر، هدف اصلی MLاین است که به رایانه‌ها اجازه دهد تا به‌طور خودکار و بدون دخالت یا کمک انسان یاد بگیرند و سپس اقدامات مورد نیاز را براساس آن تنظیم کنند. علاوه‌بر این، ML پردازش حجم عظیمی از اطلاعات را امکان پذیر می‌کند.

[1] Software Project Management

[Downloaded from jots.knu.edu.af on 2026-07-04]

۲-۳. برآورد مدیریت پروژهٔ نرم‌افزار براساس ML

شکل ۱ روش تخمین مدیریت پروژهٔ نرم‌افزاری را نشان می‌دهد که می‌توان آن را به‌صورت زیر خلاصه کرد.



شکل ۱: نمونه‌ای از روش‌شناسی توسعهٔ برآورد مدیریت پروژهٔ نرم‌افزاری

مرحلهٔ ۱. جمع‌آوری داده‌ها: استخراج توکن‌ها (نشانه‌ها)، توکن‌های کلمه، ویژگی‌های ضروری برای محاسبهٔ تلاش در پروژه هستند. بخش کلیدی مواد پروژه توسط توکن‌ها تعریف می‌شود. از مدل‌سازی زبان Unigram به‌عنوان مؤلفهٔ اصلی مدل تخمین در مورد توکن‌ها استفاده شد.

مرحلهٔ ۲. استخراج ویژگی: پس از استخراج توکن‌ها، ویژگی‌های پروژه برای تجزیه‌وتحلیل بیشتر انتخاب شدند: فرکانس مدت-فرکانس معکوس سند (TF-IDF) هر توکن «فرکانس مدت» ابزاری برای محاسبهٔ معنای عبارتی است که فراوانی کلمه و شمارش معکوس رکوردها یا کلمات ازجمله این کلمه را در نظر می‌گیرند که روشی برای اندازه‌گیری معنای کلمه‌ها است.

مرحلهٔ ۳. ساخت مدل: برای الگوریتم‌های طبقه‌بندی مختلف، ویژگی‌های مشتق‌شده به‌عنوان ورودی Weka استفاده می‌شود. در زیر مروری بر نحوهٔ انتخاب الگوریتم‌های یادگیری موجود است: بیز ساده،[1] طبقه‌بندی احتمالی‌ای است که بر قضیهٔ بیز متمرکز است که ویژگی‌های مستقل را از کلاس درس می‌گیرد. بیز ساده، با سادگی خود با ابعاد بالای داده‌ها و با ادعای ذهنی آزادی مبارزه می‌کند تا می‌تواند از رویکردهای طبقه‌بندی پیچیده‌تر نیز فراتر رود. درخت تصمیم J48: منبع باز جاو C4.5 یک الگوریتم مولد درخت تصمیم است که در آن مجموعهٔ تَنینگ[2] به‌صورت خطی قابل تفکیک نیست. درختان تصمیم به‌خوبی با نتایج آموزشی مطابقت دارند. جنگل

[1] Naïve Bayes
[2] tanning collection

مروری بر مدیریت پروژهٔ نرم‌افزار با استفاده از تکنیک یادگیری ماشین
[Downloaded from jots.knu.edu.af on 2026-07-04]

تصادفی: طبقه‌بندی‌کنندهٔ دسته‌بندی است که از چندین درخت تصمیم و خروجی‌های کلاس تشکیل شده و مدل آماری گروه‌های خروجی درختان در آن جداگانه هستند.

مرحلهٔ ٤. تجزیه‌وتحلیل: این مطالعه، بهترین مدل‌های ML را برای عملکرد بهتر پیدا کرده و نشان می‌دهد که محاسبهٔ ریسک پروژه با استفاده از یادگیری ماشین در به حداقل رساندن خطای پروژه برای بهبود بخشیدن به احتمال پاسخ پروژه مؤثرتر است و راه جایگزینی برای کارآمدی جهت کاهش احتمالات و افزایش نسبت خروجی برای رشد ارائه می‌دهد.

۳. روش‌شناسی

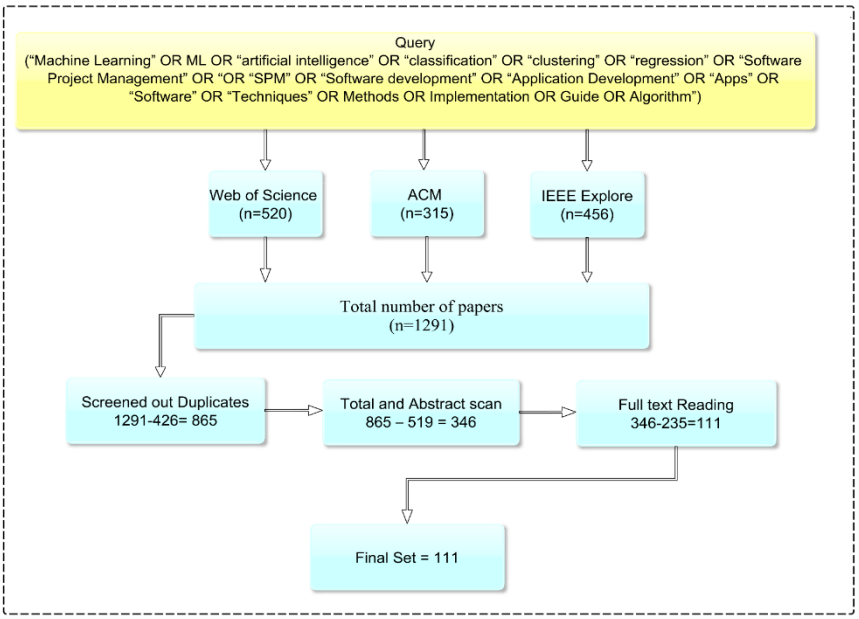
با جستجوی دو عبارت «یادگیری ماشین» و «مدیریت پروژهٔ نرم‌افزار» مقالاتی را به دست آوردیم که با SPM مرتبط بودند. جستجو در سه کتابخانهٔ دیجیتال انجام شد که عبارت‌اند از: ۱. Web of Science (WOS)، به‌دلیل اینکه به مقالات تحقیقاتی چندرشته‌ای در زمینه‌های علم، هنر و... می‌پردازد؛ ۲. IEEE، به دلیل اینکه مقالات تخصصی در زمینهٔ مهندسی برق و الکترونیک را ارائه می‌دهد؛ ۳. کتابخانهٔ دیجیتال ACM که دارای پایگاه دادهٔ جامع و مقالات علمی در زمینهٔ محاسبات و فناوری اطلاعات است.

مقالات و ادبیات مهم ارجاع‌شده توسط خروجی‌های جستجو براساس دو معیار انتخاب و دسته‌بندی شدند: ۱. استفاده از سه تکرار در فراین فیلتر کردن که مقالات اضافی و تکراری را به استثنای مقالات نامربوط با استفاده از عنوان حذف می‌کند؛ ۲. غربالگری اولیه را انجام می‌دهد و مقالات انتخاب‌شده را با مطالعهٔ دقیق نتایج جستجوی محدودشده صفحه با استفاده از SPM دنبال می‌کند.

بسیاری از جستجوها در سه پایگاه دادهٔ ذکرشده در مارس ۲۰۲۰ با استفاده از چندین کلمهٔ کلیدی (عبارات) مانند «یادگیری ماشین» یا «ML» یا «هوش مصنوعی» یا «طبقه‌بندی» یا «خوشه‌بندی» یا «رگرسیون» و «مدیریت پروژهٔ نرم‌افزاری» یا «SPM» یا «توسعهٔ نرم‌افزار» یا «توسعهٔ برنامه‌ها» یا «برنامه‌ها» یا «نرم‌افزار» یا «تکنیک‌ها» یا «روش‌ها» یا «پیاده‌سازی» یا «راهنما» یا «الگوریتم» انجام شد.

شکل ۲ متن پرس‌وجوی معمولی استفاده‌شده را نشان می‌دهد. ما نتایج جستجو را که مکاتبات، نامه‌ها، فصل‌های کتاب و... هستند، با استفاده از گزینه‌های پیشرفتهٔ موتورهای جستجو حذف کرده‌ایم. برای دسترسی جدیدترین مقالات علمی و فقط مقالات، استثنائات اهمیت زیادی دارند که قابلیت SPM را افزایش می‌دهند. تمرکز بر این است که تمامی مقالات و دست‌نوشته‌های علمی که تمامی معیارهای این اثر را برآورده می‌کنند، در بر گیرد. سپس، آن‌ها به کلاس‌های عمومی و درشت‌دانه تقسیم می‌شوند. مورد دوم در چهار بخش بعدیِ به دست آمده از نتایج مطالعه بحث می‌شود که در آن از موتور جستجوی Google Scholar برای تعیین جهت مطالعه استفاده شده است.

[Downloaded from jots.knu.edu.af on 2026-07-04]



شکل ۲: راهنمای روش تحقیق

شکل ۲ نشان می‌دهد که ۱۲۹۱ مقاله پس از انجام پرس‌وجوها جمع‌آوری شده که از بین تمام اسناد، ۵۲۰ مقاله از WOS، ٤٥٦ مقاله از IEEE و ۳۱۵ مقاله از کتابخانه‌های دیجیتال ACM بوده است. همهٔ مقالات منتخب بین سال‌های ۲۰۰۹ و ۲۰۲۰ منتشر شده‌اند؛ این مقالات بعداً به سه گروه تقسیم شدند که عبارت‌اند از ۱. ٤٢٦ مقالهٔ اضافی؛ ۲. ۵۱۹ مقاله که براساس عناوین و چکیده‌ها نامربوط بودند؛ ۳. ۱۱۱ مقاله که در معیارهای SPM قرار دارند.

همانطور که قبلاً مشخص شد، مقاله‌ها درصورتی‌که معیارهای انتخاب را برآورده نکند، حذف می‌شود؛ این معیارها عبارت‌اند از: ۱. مقاله به زبان انگلیسی نوشته نشده باشد؛ ۲. تمرکز مقاله بر تکنیک‌ها و/یا روش‌ها باشد؛ ۳. علاقهٔ پژوهشی در مقاله فقط روی SPMو بدون توسعهٔ نرم‌افزار یا یادگیری ماشین متمرکز باشد.

علاوه‌بر این، اگر SPM پس از تکرار دوم گنجانده نشده باشد، مقالات همچنان حذف می‌شوند یا دو حالت رخ می‌دهد: ۱. سهم مقاله هیچ جنبه‌ای از یادگیری ماشین و مدیریت پروژه را در نظر نمی‌گیرد؛ ۲. بحث مقاله فقط روی SPM متمرکز است و هیچ موضوع دیگری را بحث نمی‌کند. در این کار، مقاله‌ها تحت ML گسترده قرار می‌گیرند و به موجب آن، مقالات باقی‌مانده بعداً در دسته‌هایی دسته‌بندی می‌شوند که به چگونگی افزایش SPM می‌پردازند.

۳-۱. تهدید به اعتبار

مطالعات دیگر به چالش‌های قابل توجهی برای اثربخشی SLRها اشاره کرده‌اند [۱۷،۱۸] و روندهای استفاده از الگوریتم‌های یادگیری ماشین، مجموعهٔ داده‌های معیار، روش‌های

[Downloaded from jots.knu.edu.af on 2026-07-04]

اعتبارسنجی و معیارهای اندازه را برای تخمین تلاش نرم‌افزار برجسته کرده‌اند. چهار استراتژی مختلف برای به حداقل رساندن خطراتی که توسط این TTVها به‌صورت استراتژیک ایجاد می‌شود، استفاده شد:

۱. اعتبار سازه: چهارچوب با اجرای جست‌وجوی دستی و خودکار جملات تأیید شد تا داده‌های SPM محاسبه‌شده از جمع‌آوری داده‌ها به حداقل برسد. علاوه‌بر این، مقالات انتخاب‌شده با تجزیه‌وتحلیل کامل نتایج جستجوی کاهش یافته و SPM را ارزیابی می‌کنند.

۲. اعتبار داخلی: روش‌هایی که توسط [۱۷،۱۸] برای حل اعتبار داخلی استفاده شد. علاوه‌بر این، برای جلوگیری از سوگیری در طول جستجوی جامع برای مقالات مجلات، تکنیکی که دو مرحله از جستجو را ترکیب می‌کند برای رویکرد انتخاب جامع استفاده شد. کلیهٔ مقالات مورد علاقه از پایگاه‌های دادهٔ مورد استفاده برای تحقیقات مرتبط استخراج شده است [۱۷،۱۸،۱۹] و تحت فرایندهای انتخاب کامل قرار گرفتند که در شکل ۲ نشان داده شده است.

۳. اعتبار خارجی: اعتبار خارجی با ادغام ده سال چهارچوب زمانی مطالعات SPM بررسی شد که منجر به نتایج کلی گشت. رابطه‌ای موازی بین مجموعهٔ انباشتهٔ مقالات و مقالات موجود دیده می‌شود که نشان می‌دهد این SLR می‌تواند گزارشی تعمیم‌یافته را حفظ کند که با معیارهای اعتبار خارجی تحقیق مطابقت دارد.

٤. اعتبار نتیجه‌گیری: اعتبار نتیجه‌گیری با استفاده از روش‌ها و دستورالعمل‌های SLR به‌کارگرفته‌شده توسط محققان از نشریات معتبر مانند [۱۹] که نتایج را برای بازتولید گاه‌شماری تحقیقاتی این SLR با نتایج قابل اندازه‌گیری و یکسان ممکن می‌سازد انجام شد.

۳-۲. سؤالات تحقیق

با توجه به انجام بررسی سیستماتیک ادبیات تحقیق، سؤالات تحقیق نقش برجسته‌ای در تصمیم‌گیری استراتژی جستجو و تجزیه‌وتحلیل دارند. ما سؤالات پژوهشی زیر (RQs) را برای این تحقیق شناسایی کردیم:

۱. ادبیات تحقیق موجود در مورد مدیریت پروژهٔ نرم‌افزاری با استفاده از تکنیک‌های یادگیری ماشین چه چیزی را نشان می‌دهد؟

۲. آیا می‌توانیم مدل‌های مبتنی‌بر یادگیری ماشین بهتری را از نظر پیش‌بینی دقت با اعمال تبدیل و انتخاب ویژگی برای کاهش احتمالات شکست پروژه به‌طور مؤثر بسازیم؟

۳. شکاف‌های موجود برای چشم‌انداز تحقیق در زمینهٔ مدیریت پروژهٔ نرم‌افزاری چیست؟

٤. معیارهای پیش‌بینی و سطح دقت فعلی آن‌ها که توسط تکنیک‌های تخمین مختلف اثبات می‌شود، چیست؟

۵. کدام الگوریتم یادگیری ماشین تمایل به دستِ کم گرفتن و کدام الگوریتم دست کم گرفتن دارد؟

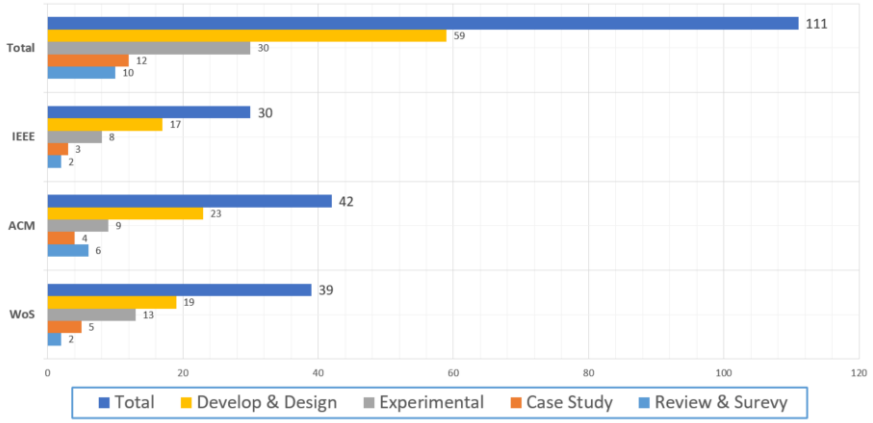
[Downloaded from jots.knu.edu.af on 2026-07-04]

۳-۳. اطلاعات آماری در گردآوری مقالات

نتیجهٔ بررسی در قالب پاسخ به سؤالات تحقیق مطرح می‌شود.

سؤال اول تحقیق: ادبیات تحقیق موجود در مورد مدیریت پروژهٔ نرم‌افزاری با استفاده از تکنیک‌های یادگیری ماشین چه چیزی را نشان می‌دهد؟

شکل ۳ طبقه‌بندی را نشان می‌دهد. رکوردها را می‌توان به چهار کلاس اساسی دسته‌بندی کرد که شامل ۱. مرور و بررسی،[1] ۲. مطالعهٔ موردی،[2] ۳. تجربی[3] و ٤. تحلیل و معماری[4] می‌شوند. دستهٔ اول؛ مواد تحقیق و پرسش‌نامه، رویکردها و راهبردهای ML به کار گرفته شده در SPM را برای دستیابی به اهداف و رفع نگرانی‌ها تشریح می‌کند. دستهٔ دوم، اثرات، محرک‌ها، اقدامات متقابل و شرایط را بحث می‌کند و فناوری‌هایی را برای بهبود کنترل اثربخشی پیشنهاد می‌کند. دسته سوم اثرات روش مورد استفاده برای طبقه‌بندی متغیرهای متعدد را ارائه می‌دهد که می‌تواند جنبه‌های مختلف روش یا محصول را در حین تولید تحت‌تأثیر قرار دهد. دستهٔ چهارم شامل ساختارها، روش‌ها و تخصص برای مأموریت است.



شکل ۳: تعدادی مقاله در دسته‌بندی‌های مختلف براساس مجلات چاپی

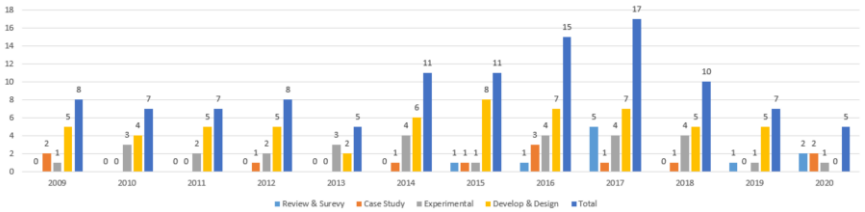
شکل ۳ آمار دسته‌های مختلف بالا را برای مقالات مرتبط با SPM نشان می‌دهد؛ در این شکل، ۱۱۱ مقاله از سه پایگاه داده به بررسی و نظرسنجی (۱۰)، مطالعهٔ موردی (۱۲)، توسعه و طراحی (۵۹) و مقالاتی که مطالعهٔ تجربی را توصیف می‌کنند (۳۰) ، تقسیم شده‌اند.

شکل ٤ تعداد انتشارات را براساس زمینه‌ها و منطقه‌ای که مطالعه و مطالعات در SPM در آن توسعه یافته است، نشان می‌دهد. یافته‌ها به ۱۰ مقاله از ۱۱۱ مقاله تقسیم شدند. ۱۲ مقاله از ۱۱۱ مقاله، مربوط به روش‌های تحلیل موردی ML و استراتژی‌هایSPM و ۳۰ مقاله، مراحل

[1] Review and Survey
[2] Case Study
[3] Experimental
[4] Analytics and Architecture

[Downloaded from jots.knu.edu.af on 2026-07-04]

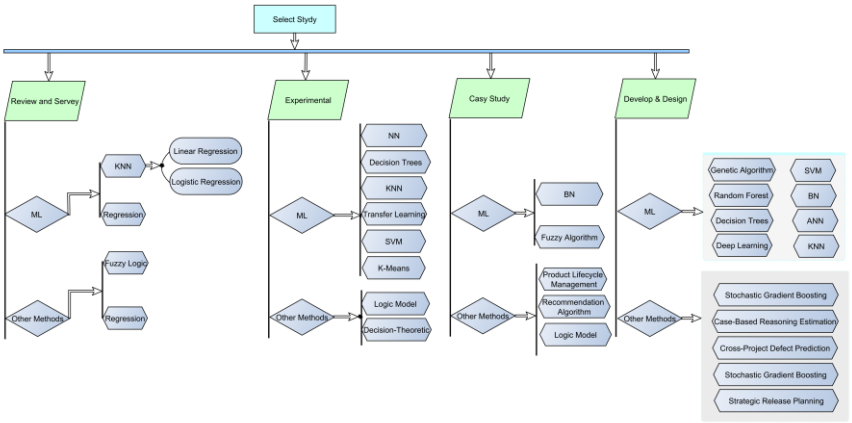
سیستماتیک و معیارهای آزمایشی برای بررسی مدیریت پروژه ML-Software هستند. گروه‌های نهایی، مشارکت‌های علمی و نتایج در طراحی و رشد تحقیق ML-SPM، ۵۹ مقاله از ۱۱۱ مقاله است. این شکل حتی مطالعهٔ ریاضی گروه‌های متعدد را نشان می‌دهد.



شکل ٤: مقالات منتشر شده بین سال‌های ۲۰۰۵ تا ۲۰۲۰

شکل ٤ از طرف دیگر، شامل مقالاتی است که وابسته به سال انتشار هستند و مقالات طبقه‌بندی‌شدهٔ علمی را بین سال‌های ۲۰۰۹ تا ۲۰۲۰ نشان می‌دهد. در سال ۲۰۰۹، تنها ۸ مقاله نوشته شده و ٤۹ مقاله از سال ۲۰۱۰ تا ۲۰۱۵ منتشر شده است. برای سال‌های ۲۰۱٦، ۲۰۱۷، ۲۰۱۸، ۲۰۱۹ و ۲۰۲۰ به‌ترتیب ۱۵، ۱٦، ۱۱، ۷ و ۵ مقاله نوشته شده است. منابع اولیهٔ تجزیه‌وتحلیل براساس تحقیقات ML-SPM گنجانده شده و دستورالعمل‌های کلی آن ارزیابی شد.

همانطور که در شکل ۵ نشان داده شده است، چندین روند پیدا کردیم و یک طبقه‌بندی تولید کردیم. ما همچنین بسیاری از زیرمجموعه‌ها را متمایز کردیم، اما چندین حوزهٔ اصلی مشاهده شده است. همانطور که در شکل ۵ مشاهده می‌شود، ما متوجه برخی مضامین از ادبیات تحقیق شدیم و یک طبقه‌بندی ارائه کردیم؛ درحالی‌که فیلدهای خاصی با یکدیگر همپوشانی داشتند، چندین زیرمجموعه ایجاد کردیم.



شکل ۵: طبقه‌بندی ادبیات ارزیابی ریسک پروژهٔ نرم‌افزاری با استفاده از تکنیک یادگیری ماشین

۳-٤. مرور و بررسی مقالات

اسناد تجزیه‌وتحلیل و تحقیق، آخرین ادراک از فناوری‌های ML را در تهیه و ارزیابی SPM و کاربرد الگوریتم‌های ML نشان می‌دهند.

[Downloaded from jots.knu.edu.af on 2026-07-04]

۳-٤-۱. مطالعات انجام‌شده دربارهٔ یادگیری ماشین و استفاده از آن‌ها در SPM

این بخش، فرایندهای ML را بحث می‌کند و از آن‌ها استفاده می‌کند. مقالات به چندین موضوع و اجرا تقسیم شدند. مطالعات منتخب به گروه‌های بزرگی دسته‌بندی شده‌اند که بر روش‌های ML تکنیک‌های تولید متمرکز شده‌اند. برای شش نشریهٔ این گروه به ترتیب سه زیرمجموعه وجود داشت. این خوشهٔ فرعی روی الگوریتم K-Nearest Neighbor (KNN) انجام شد. در منبع [۲۰] مشاهدات، معیارها، مجموعهٔ داده‌ها، معیارهای محاسباتی، چالش‌های ML، مدل‌های مختلف پیش‌بینی‌ها و مدل‌های مجموعهٔ مورد استفاده در منطقهٔ پیش‌بینی تعمیر و نگهداری ارزیابی و تجزیه‌وتحلیل شدند. مقالهٔ [۲۱] با KNN برای مدیریت مقادیر گمشده در ساختارهای دادهٔ مهندسی اطلاعات، نگرانی فزاینده‌ای را برای فناوری‌های ML نشان داده است.

مطالعات طبقه‌بندی دیگری روی رگرسیون انجام شد. مقالهٔ [۱۹] روش‌ها و پیش‌بینی‌کننده‌های پیش‌بینی نوسانات و معیارهای طبقه‌بندی را شناسایی کرد. ویژگی‌هایی که به‌عنوان شاخص‌های پارامترهای نوسانات ادبیات و تکنیک‌های پیش‌بینی مورد استفاده برای افزایش دقت نوسانات الزامات پیش‌بینی مورد استفاده قرار گرفتند، ایجاد شده‌اند. مشخصات بانوسان برای برنامه‌های نرم‌افزاری حیاتی هستند، زیرا مستقیماً منجر به هزینه‌ها و دورهٔ بیش از حد می‌شوند. در [۲۲] SLR برای کمک به مکانیسم رسمی یافته‌های تکرارپذیر پیشنهاد شد. مطالعه نمی‌تواند کاربرد دقیق را توسط سازماندهی یک مجموعهٔ داده مانند سایر مجموعه‌های داده حل کند. مقالهٔ [۱۸] به استفاده از روش‌های ML برای محاسبهٔ تلاش برنامه پرداخته است. مطالعهٔ سیستمیک نشان داد که رویکردهای ML، مقیاس‌های اندازه، مجموعهٔ داده‌های مقایسه‌ای، روش‌های ارزیابی و... تأثیرگذار بودند.

یک مقاله در مورد مطالعات منطق فازی [۲۳] استفاده از روش‌های ML را برای آزمایش تلاش برنامه بررسی کرد. علاوه‌بر این، او با تعدادی کار نرم‌افزاری، ارزیابی هزینه‌های روش‌های عملکرد سیستم‌ها را تشریح کرد و نتیجه‌گیری اصلی این بود که هیچ روش دیگری نباید براساس فراین و مدل ترجیح داده شود.

۳-٤-۲. سایر روش‌ها

این بخش به بحث و استفاده از سایر رویکردها می‌پردازد. این مقالات در موضوعات مختلف و پیاده‌سازی گروه‌بندی شده‌اند. آثار فهرست‌شده در گروه مدل منطقی خاصی گروه‌بندی می‌شوند.

نویسندهٔ [۲٤] از ارزیابی بازدهٔ مالی سرمایه‌گذاری (ROI) زیرساخت شبکه استفاده کرد. کالایی که منفعت «فروش» ایجاد نمی‌کند هنگام اجرای یک اصل مالی ROI دشوار است، همانطور که در خرید یا فروش موجودی‌ها در بیشتر محیط‌های دانشگاهی دیده می‌شود. مقالهٔ [۱۷] تحقیقات اخیر تعمیر و نگهداری برنامه را به‌طور گسترده تجزیه‌وتحلیل می‌کند. نتایج مطالعه نشان داد که استفاده در پیش‌بینی تعمیر و نگهداری از الگوریتم‌های ML از سال ۲۰۰۵ افزایش یافته است. مشکلات براساس حوزه‌های تخصصی بدنهٔ دانش مدیریت پروژه (PMBOC) طبقه‌بندی و تجزیه‌وتحلیل شدند [۲۵،۲٦]. مشکلات با استفاده از ابزارهای هوش مصنوعی و موانع در چابک PM مورد

[Downloaded from jots.knu.edu.af on 2026-07-04]

بررسی قرار گرفت. این کمک به نیاز مورد انتظار برای توسعه مدل‌های PM مدرن و تکنیک‌های IT مربوط می‌شود که روش‌های مبتنی‌بر ML را با درمان عدم دقت، ابهام یا ابهام به شاخص‌های عملکرد حیاتی که با حوزه‌های تخصصی تخصصی مرتبط هستند، ادغام می‌کند.

۳-۵. مطالعات تجربی

این بخش فناوری‌هایی را طبقه‌بندی می‌کند که تست‌های اندازه‌گیری استاندارد و پارامترهای تجربی را که در تجزیه‌وتحلیل ML-SPM استفاده می‌شوند، انجام می‌دهند؛ این مقالات به موضوعات و اجراهای مختلفی تقسیم می‌شوند.

سؤال دوم تحقیق: آیا می‌توانیم مدل‌های مبتنی‌بر یادگیری ماشین بهتری را از نظر پیش‌بینی دقت با اِعمال تبدیل و انتخاب ویژگی برای کاهش احتمالات شکست پروژه به‌طور مؤثر بسازیم؟

۳-۵-۱. مطالعات انجام‌شده روی روش‌های یادگیری ماشین

بر اساس رویکردهای روش‌شناسی تولید ML، مقالات انتخاب‌شده در گروه‌های وسیعی دسته‌بندی شدند؛ ۲۷ نشریه در ۷ زیرمجموعه در گروه زیر سازماندهی شدند که شامل ۱۵ مقاله است که برای SPM با الگوریتم‌های مختلف ML استفاده می‌شود. جدول ۱- که مربوط به ML است- تعریف، دامنه و دیگر جنبه‌های اصلی تأسیس تجربی را در تضاد قرار می‌دهد.

جدول ۱: مطالعات انجام‌شده روی روش‌های یادگیری ماشین

مرجع	نوع ML	شرح	دامنه	استخراج ویژگی	محدودیت سیستم قدیمی	محدودیت سیستم جدید
[27]	SVM	دو رویکرد ML را برای تقویت هماهنگی بین کدهای نظارتی و مشخصات در سطح کالا ارزیابی کرد.	امنیت و حریم خصوصی در حوزهٔ مراقبت‌های بهداشتی	-	موفقیت محدود برای ردیابی کدهای نظارتی به‌دلیل تفاوت در اصطلاحات که می‌تواند بین کدها و الزامات سطح محصول وجود داشته باشد.	داده‌کاوی را روی مدل دقیق‌تری از کدهای نظارتی HIPAA اعمال کرد که حقوق خاصی را نشان می‌دهد.

[Downloaded from jots.knu.edu.af on 2026-07-04]

[Downloaded from jots.knu.edu.af on 2026-07-04]

[28]	[29]	[30]	[31]
چندین نوع ML	چندین نوع ML	چندین نوع ML	چندین نوع ML
استدلال کرد که تجزیه‌وتحلیل اطلاعات از فناوری‌های محاسباتی استفاده می‌کند.	ارزیابی ماشین را توسعه دهید، استفاده از سرمایه را به حداکثر برسانید.	راه‌حل جدیدی را برای رسیدگی به این معضل فراگیر از طریق ترکیب مدرن دیجیتالی کردن و ML نشان	تکمیل پوکر برنامه‌ریزی دستی Agile
طیف گسترده‌ای از تجربه و آگاهی میدانی	برآورد تلاش و مدت	ارزیابی پروژه، سرعت تیم و برآورد زمان	تخمین تلاش توسعۀ نرم‌افزار
-	-	-	استخراج توکن
تجزیه‌وتحلیل ماشین کامل، تجزیه‌وتحلیل نرم‌افزار، ML، پردازش داده‌ها و تجسم دانش	برنامه‌ریزی و شاخص‌های تاریخی کالا بسته به روش یادگیری	ایجاد مفهوم آبشار حدود یک دهه پیش	هیچ چهارچوبی برای رشد چابک وجود ندارد که مناسب‌ترین باشد.
تخصص در طراحی و پیاده‌سازی ابزارهای پردازش دادۀ مقیاس‌پذیر و ابزارهای یادگیری	در دسترس بودن داده های دانه‌ای در مورد ویژگی‌های پروژه و محصول	توسعه‌داده‌شده برای تولید داده‌ها دربارۀ مشارکت فردی و تیمی که می‌تواند برای مدیریت مفید باشد.	مجموعۀ داده‌ها و توابع بزرگتر در این آزمایش شامل نمی‌شود.

[Downloaded from jots.knu.edu.af on 2026-07-04]

[32]	[33]
چندین نوع ML	چندین نوع ML
بسیاری از استراتژی‌های انفرادی برای پیش‌بینی تلاش توسعهٔ نرم‌افزار سیستم پیشنهاد شد.	هدف دستیابی به راه‌حلی با پیاده‌سازی دستگاه هوشمندی بود که اعضای تیم را خلاقانه به ماموریت خاصی اختصاص می‌دهد.
تخمین تلاش نرم‌افزاری	مدیریت پروژهٔ نرم‌افزاری
ارقام مجموعهٔ داده شامل تعداد سرمایه‌گذاری‌ها و تعداد ویژگی‌ها است.	CollabCrew ETL
در هر صورت کافی دیده شده است.	در اصل برای مقابله با مشکل نرم‌افزار ساخته شده است.
هدف ارزیابی تأثیر تعداد شرکت‌کنندگان گروه بود.	نتایج این تحقیق به نفع چهارچوب بلادرنگ است و بینشی دربارهٔ کارایی، دقت و سطح قابلیت اطمینان ارائه می‌کند.

[Downloaded from jots.knu.edu.af on 2026-07-04]

[34]	[35]
NB و SVM	چندین ML دیگر
مقایسهٔ گسترده‌ای از داده‌های شناخته‌شده ارائه شده است.	راه‌حل مدل سازگار آنلاین فعال به ACONA ارائه دهید که مجموعه‌ای از دسته‌ها را به‌صورت پویا با پروژه‌های مختلف تطبیق می‌دهد.
پیش‌بینی نقص بین پروژه‌ها	مدیریت فراین توسعهٔ نرم‌افزار؛ مدیریت ریسک
رویکردهای مبتنی‌بر ویژگی	-
استراتژی داده Iter به‌طور قابل توجهی کارایی پیش‌بینی نقص بین پروژه را بهبود می‌بخشد و روش انتخابی سلسله‌مراتبی پیشنهادشده به‌طور قابل توجهی عملکرد را بهبود می‌بخشد.	استفاده از طبقه‌بندی‌های آموزش‌دیده برای ارائهٔ پیش‌بینی‌های خوب برای پروژهٔ فعلی با جریان داده‌ها دربارهٔ داده‌های تاریخی گسترده از پروژه‌های دیگر
طبقه‌بندی‌کنندهٔ دیگری برای ساختمان مدل به غیر از NB یا SVM پیدا کنید.	به نتایج بهبودیافته با نگرانی کمتر دربارهٔ طرح CI واقعی می‌رسد که نشان می‌دهد ACONA می‌تواند به‌طور چشمگیری هزینه‌های CI را بیش از روش‌های فعلی به حداقل برساند.

[Downloaded from jots.knu.edu.af on 2026-07-04]

[36]	[37]	[38]
RF، پرسپترون چندلایه و SVM	DT، FL	SVM
هدف از پیش‌بینی تلاش	در موارد خاص، ارقام قابل اعتمادی ارائه می‌دهد.	پروژهٔ توسعهٔ خارجی از رویکردهای کلیدی برای ساخت نرم‌افزاری است که میزان شکست بالایی دارد. مدل هوشمند پیش‌بینی ریسک می‌تواند به زمان‌بندی پروژه‌های پرخطر کمک کند.
تلاش پروژهٔ نرم‌افزاری	برآورد هزینهٔ نرم‌افزار	پروژهٔ نرم‌افزاری
ویژگی‌های غیرخطی	ویژگی‌های زیرمجموعه از ISBSG	۲۵ عامل خطر را انتخاب کرد.
برآورد دقیق تلاش پروژهٔ نرم‌افزاری	مدل‌های دقیق و مفید ساخته‌شده درواقع محدود هستند؛ حتی اگر به ذی‌نفعان فناوری مزایای مالی	مدل‌های موجود عمدتاً بر این فرض متمرکز شده‌اند که تمام هزینه‌های طبقه‌بندی اشتباه معادل هستند که با این واقعیت که پیش‌بینی ریسک در منطقهٔ پروژهٔ نرم‌افزاری وجود دارد، ارتباطی ندارد.
ترکیب سایر مدل های ML مانند Treeboost، XBoost و... و اعتبارسنجی با مجموعهٔ داده‌های متنوع دیگر	مدل‌هایی در حوزهٔ رشد واقعی	از طبقه‌بندی‌کننده‌های قوی‌تری برای بهبود دقت پیش‌بینی ریسک پروژهٔ نرم‌افزاری برون‌سپاری‌شده استفاده می‌کند.

[Downloaded from jots.knu.edu.af on 2026-07-04]

[39]	[40]	[41]
SVM	K-Means	DT
تأثیر دامنه‌های پر سروصدا بر هشت دقت ML و الگوریتم‌های تشخیص روندهای آماری را بررسی می‌کند.	از استراتژی طراحی مهندسی اطلاعات خاصی برای شناسایی نرم‌افزار معیوب استفاده کرد	تخمین تلاش نرم‌افزار حیاتی‌ترین کار در مهندسی نرم‌افزار و PM است.
پیش‌بینی تلاش نرم‌افزاری	توسعهٔ جهانی نرم‌افزار	تخمین تلاش نرم‌افزاری
ویژگی به‌طور تصادفی انتخاب شده است.	انتخاب زیرمجموعهٔ ویژگی	-
راه‌حل‌هایی برای مشکل حوزه‌های پر سروصدا در پیش‌بینی تلاش نرم‌افزار از دیدگاه احتمالی	برای ارتقای تصمیمات نرم‌افزار PM با داده‌کاوی و تولید نتایج عملی	با توجه به مقایسهٔ الگوریتم‌های ML برای تخمین تلاش در نرم‌افزار با اندازه‌های مختلف
با در نظر گرفتن مطالعهٔ شبیه‌سازی دقیق‌تر با استفاده از انواع بسیار متعادل‌تر مجموعهٔ داده‌های مورد نیاز برای درک شایستگی STOCHS، به‌ویژه مجموعهٔ داده‌های بزرگ‌تر، گسترش یافته است.	بررسی و مقایسه با سایر روش‌های داده‌کاوی	با استفاده از سایر الگوریتم‌های ML و اعتبارسنجی با سایر مجموعه‌های دادهٔ متنوع تقویت می‌شود.

[Downloaded from jots.knu.edu.af on 2026-07-04]

[42]	[43]	[44]
kNN، DT، و LDA	شبکهٔ عصبی	ANN، SVM
رویکرد هوشمند برای پیش‌بینی خطای نرم‌افزار براساس بهینه‌سازی شعلهٔ پروانهٔ دودویی با نمونه‌گیری مصنوعی تطبیقی معرفی شد.	MLبه‌عنوان رگرسیون عمومی شبکهٔ عصبی برای پیش‌بینی کارایی در عملکرد برنامه‌ها نام‌گذاری شد.	چندین الگوریتم ML برای پیش‌بینی مدت‌زمان نرم‌افزار
پیش‌بینی خطای نرم‌افزار (SFP)	متخصصان نرم‌افزار	SPM
فرکانس انتخاب هر ویژگی از همهٔ مجموعهٔ داده‌ها با استفاده از EBMFOV3	-	-
بهبود عملکرد همهٔ طبقه‌بندی‌کننده‌ها پس از حل مشکلات نامتعادل	توسعه‌دهندگان و مدیران به خروجی متخصصان فناوری اشاره می‌کنند که معمولاً به‌عنوان نسبت اندازه به زمان محاسبه می‌شود.	الگوریتم‌ها را با توجه به ضریب همبستگی آن‌ها ارزیابی کرد.
اهمیت ویژگی‌ها را برای افزایش عملکرد طبقه‌بندی‌کننده‌ها و دقت مدل SFP مطالعه کرد.	استفاده از شبکه‌ای عصبی با ویژگی پایهٔ شعاعی برای پیش‌بینی کارایی متخصصان و تیم‌های توسعه‌دهنده	پیش‌بینی با توجه به جزئیات پروژهٔ فعلی/گذشته عمل می‌کند و کار بالقوه و طول پروژه را تخمین می‌زند.

[Downloaded from jots.knu.edu.af on 2026-07-04]

[45]	[46]
درخت تصمیم	درخت تصمیم
تصمیمات درحال تکامل پیشنهادی ازطریق الگوریتمی تکاملی و درخت مربوطه برای پیش‌بینی تلاش تعمیر و نگهداری دستگاه	ابزاری برای تقویت عملکرد پیش‌بینی تلاش برنامه پیشنهاد شده است.
پیش‌بینی تلاش نرم‌افزاری	پیش‌بینی نرم‌افزار
-	ویژگی چهاربعدی
استفاده از HEAD-DT برای ایجاد الگوریتمی مبتنی‌بر قراردادهای قضاوت که با برنامۀ نگهداری داده‌ها سازگار است.	آغاز درک بهتر و استفاده از نهادهای تصمیم‌گیرنده به‌عنوان طبقه‌بندی بخشی از روش‌های انتساب مجموعه
اثربخشی فراابتکاری در ارزیابی سایر شاخص‌های نرم‌افزار اولیه، ایجاد داده در نرم‌افزارهای خصوصی و عمومی	داده های ناقص و برآورد ماشینی تحلیل نظری و مشاهده‌ای

[Downloaded from jots.knu.edu.af on 2026-07-04]

[47]	[48]
k-NN	درخشان و رگرسیون
برای کشف اینکه چگونه پارامترها با پارامترهای خود تطبیق بیشتری دارند و هر چندوقت یک‌بار ممکن است خروجی MLs در SEE تحت‌تأثیر قرار گیرد.	هدف، جزئیات محاسبهٔ تلاش ماشینی بین شرکتی (CC) استفاده صریح از دانش یا مدل‌های CC برای پیش‌بینی داده‌های CC یا داده‌های مدل در موقعیت‌های WC است.
تخمین تلاش نرم‌افزاری	تخمین تلاش نرم‌افزاری
-	تعداد سرمایه‌گذاری با هر مشخصه
آزمایش‌های سیستمیک روی سه مجموعهٔ داده با پنج ML در تنظیمات پارامترهای چندگانه انجام شد.	این سیستم نه‌تنها از دانش WC بسیار کمتری نسبت به مدل WC قابل مقایسه استفاده می‌کند، بلکه خروجی معادل/بهتر تولید می‌کند.
بررسی ML و مجموعهٔ داده‌های اضافی؛ سایر اَشکال اندازهٔ عمل ازجمله موارد غیرپارامتریک و اندازهٔ پنجره‌های اضافی برای ارزیابی یادگیری آنلاین	حساسیت Dycom به مقادیر پارامتر، مردمک ساده، ورودی‌ها و جدا کردن CC به بخش‌های جداگانه

[Downloaded from jots.knu.edu.af on 2026-07-04]

[49]	[50]	[51]
SVM	SVM	ANN
مطالعات سیستماتیک نشان می‌دهند RVM در مقایسه با رویکردهای پیشرفته یSEE بسیار موفق است.	محاسبهٔ درست تلاش کمک می‌کند تا مشخص شود کدام چالش‌ها باید در دور بعدی اصلاح یا حل شوند.	روش‌های کالیبراسیون به فرم‌های تنظیم خطی به جز تنظیم غیرخطی مبتنی‌بر ANN بستگی دارد.
تخمین تلاش نرم‌افزاری	برآورد تلاش	تخمین تلاش توسعهٔ نرم‌افزار
ویژگی‌های خاص حساب SEE	ویژگی‌های محاسبه‌شده روی معیارهای وظیفهٔ طبقه‌بندی وابسته به ویژگی‌های اولیه است.	غیرعادی بودن و ویژگی‌های طبقه‌بندی مجموعهٔ داده‌های مختلف
نشان داده است که RVM شاخص برجستهٔ SEE است و نیاز به تجزیه‌وتحلیل و استفادهٔ بیشتری دارد.	ویژگی‌های توسعه برای ساخت مدل‌های آماری که نقاط داستان را برای پروژه‌های منبع باز تحلیل می‌کنند، استفاده شده‌اند.	به‌عنوان روشی پایه برای تخمین تلاش توسعهٔ نرم‌افزار در نظر گرفته شده است.
با استفاده از ارزیابی اعتبار خودکار RVM، سه مورد منحصربه‌فرد ایجاد شد و توصیه‌هایی در اینکه آیا تلاش مورد نیاز پیشنهاد شده است یا خیر.	پیش‌بینی‌ها را می‌توان با در نظر گرفتن ویژگی‌های جدید مرتبط با ویژگی‌های توسعهٔ انسانی افزایش داد.	در بسط این مطالعه، گزینه‌های دیگری برای تابع هسته در LS-SVM به غیر از تابع پایهٔ شعاعی وجود دارد.

فرایندهای خوشه‌بندی، یادگیرندگان ساده، ویژگی های پروژهٔ ورودی، توابع پروژهٔ خوشه‌بندی، مقادیر پارامترها	خوشه‌بندی Dycom با K-Means به جداسازی برنامه‌های CC کمک می کند و کارایی پیش‌بینی خوب یا بهتری نسبت به Dycom ایجاد می کند.	برای توصیف پروژه‌های آموزشی برای خوشه‌بندی می توان از ویژگی های متفاوتی استفاده کرد.	تخمین تلاش نرم‌افزاری	رویکردهای خوشه‌بندی تعمیم داده می‌شوند تا برای ساخت زیر مجموعه‌های CC مورد استفاده قرار گیرند. سه روش مجزا برای خوشه‌بندی بررسی شده است.	K-Means	[52]

۳-۵-۲. مطالعات انجام‌شده روی سایر روش‌ها

این بخش بحث می‌کند و سیستم ML را گسترش نمی‌دهد. آثار انتخاب‌شده به گروه‌هایی ازجمله مدل‌ها یا تکنیک‌ها تقسیم می‌شوند؛ این مقالات در گروه‌ها دستورالعمل‌های مختلفی را با پارامترهای موجود در ارزیابی با توجه به تجزیه‌وتحلیل خود شناسایی می‌کنند.

مدل اول. حالت منطقی: تأکید [۵۳] بر افزایش ویژگی‌های کیفیت، مانند خطاها، ماه‌ها و تنش بود. طرفداران مدل پارامتری، ادعا می‌کنند که مدل‌های مستقل از دامنه ممکن است با داده‌های محلی تطبیق داده شوند. نویسندگان [۵٤] ادغام مرجع و تجسم را در پیشرفت‌های پروژه توصیه کردند.

مدل دوم. مدل پارامتریک: نویسندگان [۵۵] مزایای روش‌های محاسبهٔ نادرست هزینه را هنگام توسعهٔ الگوهایی برای خرابی‌های نرم‌افزار پیش‌بینی‌کننده که از اطلاعات پروژهٔ مخزن متقابل استفاده می‌کنند، بررسی کردند؛ در این شرایط، متوجه می‌شوید که آموزش‌های حساس به هزینه، دارای نکاتی نیستند که بیشتر از طبقه‌بندی‌کننده‌های حساس به هزینه باشند.

مدل سوم. تکنیک‌های بهینه‌سازی نظری تصمیم: در [٥٦] تکنیک‌های بهینه‌سازی نظری تصمیم ارائه شدند که می‌توانند بهترین پارامترها را برای طیف وسیعی از گردش‌های کاری انتخاب کنند. آزمایش‌های اولیه نشان می‌دهد که گردش‌های کاری بهینه‌شده به‌طور قابل توجهی مقرون‌به‌صرفه‌تر از پارامترهای تنظیم‌شدهٔ دستی هستند. آن‌ها استدلال می‌کنند که روش‌های هوش مصنوعی(AI) مانند ML، تئوری تصمیم‌گیری و بهینه‌سازی می‌توانند این مشکلات را حل کنند و ساخت سریع جریان‌های کاری مؤثر با منبع جمعی را آسان کنند.

[Downloaded from jots.knu.edu.af on 2026-07-04]

۳-۶. مطالعهٔ موردی

این بخش، پروژه، کمپین یا شرکتی را که موقعیتی، راه‌حل‌های پیشنهادی یا اقدامات اجرایی را شناسایی می‌کند، تجزیه‌وتحلیل می‌کند و عواملی را که با استفاده از تکنیک ML به شکست یا موفقیت در تکنیک‌های توسعهٔ SPM کمک کرده‌اند، شناسایی می‌کند.

مقالات این بخش روی روش‌های ML تمرکز دارند و آثار منتخب وابسته به روش‌های ML در تکنیک‌های توسعهٔ SPM در دسته‌های گسترده طبقه‌بندی شدند. هفت مقالهٔ این دسته به سه زیرمجموعه تقسیم شدند. این بخش شامل پنج مقالهٔ ML است که از بسیاری از الگوریتم‌های SPM استفاده می‌کنند. براساس رویکردهای ML روش‌شناسی توسعهٔ نرم‌افزار، کار به گروه‌های گسترده تقسیم می‌شود. مقالات [۵۷،۵۸] بر بهبود قابلیت پیش‌بینی برآورد و تخصیص تلاش مورد نیاز برای تطبیق با مشتری، مدیریت پروژه و مسائل مختلف توسعه تمرکز داشتند. ازطریق راه‌حل، نیاز به پرداختن به مسائل مربوط به پروتکل‌های گزارش‌دهی و تخصص و اطمینان از روتین بودن تحلیل کور بحث خواهد شد. دیگران [۵۹،۶۰] روشی را برای ارزیابی دیدگاه‌های ذی‌نفعان، جداسازی موضوعات بخش و ایجاد پروفایل‌هایی پیشنهاد کردند که ترجیحات ذی‌نفعان را در همهٔ موضوعات منعکس می‌کند؛ علاوه‌بر این، تکنیک‌های محاسباتی و رگرسیون پیش‌بینی‌کنندهٔ نرم‌افزار نیز در مقابل هم قرار گرفتند.

مقالات این دسته در مورد الگوریتم شبکه‌های بیزی مطالعه می‌کنند. در مقالهٔ [۶۱] راه‌حلی برای تخمین ارزش با استفاده از ترکیبی از راه‌حل‌های کیفی و ML ارائه شده است که در آن مدلی احتمالی شامل دانش سهام‌داران مختلف ارائه می‌شود که برای پیش‌بینی ارزش کلی تصمیم معینی مربوط به مدیریت و توسعهٔ محصول استفاده می‌شود. نویسندگان [۶۲] مدلی را پیاده‌سازی کردند که به‌طور خودکار رابطهٔ بین عوامل خطر و کاهش را ازطریق سیستم پشتیبانی تصمیم‌گیری هوشمند (DSS) شناسایی می‌کند؛ روش پیشنهادی آن‌ها محدودیت‌های مدیریت ریسک فعلی مانند فقدان DSS یکنواخت و ارتباط بین ریسک‌های نرم‌افزار و کاهش را که به‌طور گسترده به آن اشاره شده است، پوشش می‌دهد.

مقالات این دسته براساس الگوریتم فازی انجام شده است. مقالات [۶۳،۶۴] روش ریاضی فازی را به مدل پارامتری تأثیر ریسک برای حل مشکل جدی معرفی کردند که احتمال رویدادهای مهم به‌راحتی به دست نمی‌آید. این کار با ایجاد ساختار توپولوژی عوامل خطر، رابطهٔ عوامل تأثیرگذار مختلف را در فراین مدیریت ریسک پروژه‌های فناوری اطلاعات توصیف می‌کند؛ یافته‌ها با پارامترهای ارزیابی مختلف در تضاد هستند.

مطالعات انجام‌شده روی سایر روش‌ها

این بخش به بررسی سایر روش‌های مورد استفاده می‌پردازد. این مقالات به موضوعات و کاربردهای مختلفی تقسیم شدند؛ آن‌ها سه مقاله در این دسته هستند عبارت‌اند از:

[Downloaded from jots.knu.edu.af on 2026-07-04]

مدل اول که در مدیریت چرخهٔ عمر محصول (PLM) انجام شد. نویسندگان مقالهٔ [٦٥] PLM را ارائه کردند؛ این روش لایه‌ای از عملکرد ایجاد می‌کند تا امکان تکرار بعدی PLM در اطراف شبکهٔ PLM ایجادشده را فراهم کند، سپس این PLM جدید با استفاده از مطالعهٔ موردی Ford Powertrain در اکوسیستم اتوماسیون کارخانهٔ دیجیتال ادغام خواهد شد.

مدل دوم براساس یک الگوریتم توصیه انجام شده است. الگوریتم نرم‌افزاری جدیدی توسط [٦٦] پیشنهاد شده است که ابتدا یک ویژگی مبتنی‌بر باگ و یک مکانیسم غربالگری خاص برای تأیید اعتبار رفع‌کنندهٔ متقاضی اضافه می‌شود، شبکه‌ای از تعهدات چند برنامه‌نویس را با گرفتن طیفی از نظرات و قول‌ها ایجاد می‌کند، آن‌ها را در رتبه‌بندی قرار می‌دهد و سپس مناسب‌ترین رفع‌کنندهٔ اشکال را تعیین می‌کند. نتیجه نشان می‌دهد که این راه‌حل با موفقیت، تابع تریاژ خطا را اجرا می‌کند.

مدل سوم مدل منطقی است که دو مقالهٔ [٦٧،٦٨] از مطالعهٔ موردی واقعی در دامنهٔ توزیع‌شده استفاده کردند، آزمایش چابک را برای تیم منتخب اِعمال کردند و نتیجهٔ آن‌ها را با سه گروه دیگر مقایسه کردند تا تأثیر مشارکت مشتری در فراین آزمایش برای غلبه بر چالش‌های توسعهٔ توزیع‌شده را تعیین کنند. بااین‌حال، گروهی که از تست چابک استفاده می‌کنند بیش از ۹۹٪ از تمام درخواست‌های واردشده به فراین آزمایش را تأیید کردند که تفاوت قابل توجهی از بهره‌وری هر پروژهٔ توسعهٔ پشتیبانی دارد.

۳-۷. توسعه و طراحی

۵۹ مقاله، مطالعهٔ یک طرح، نوع ساختار یا مدل معماری برای برآوردن نیازهای مرحله است؛ جایی که نتایج تحقیق در مورد PMS برای پرداختن به آن و روش مورد استفاده توسط ML تولید می‌شود.

۳-۷-۱. مطالعات انجام‌شده روی روش‌های یادگیری ماشین

آثار برگزیده براساس روش‌های ML در SPM به دسته‌های گسترده طبقه‌بندی شدند که ٤٣ مقاله در این دسته به ۹ زیرشاخه تقسیم شدند.

این بخش شامل ۱۵ مقالهٔ فراین ML است که از الگوریتم‌های مختلف SPM استفاده می‌کنند. براساس روش ML در تولید نرم‌افزار، آثار منتخب در دسته‌های بزرگ دسته‌بندی می‌شوند.

کاربردهای حوزهٔ SPM توسط مقالات دستهٔ اول [٦٩،٧٠] ارزیابی شدند که رفتارها به‌عنوان کار و به‌طور مناسبی طبقه‌بندی می‌شوند و یک مشکل یادگیری چند هدفی را در طراحی مدل برای تخمین تلاش سیستم نشان داد. همچنین، به درک مصالحهٔ بین معیارهای مختلف عملکرد با ایجاد مدل‌های SEE که به‌طور هم‌زمان توسط چندین الگوریتم تکاملی عینی، خودکار می‌شوند، کمک می‌کند. بیز ساده، رگرسیون لجستیک و جنگل‌های تصادفی استراتژی‌هایی هستند که در این تحلیل استفاده می‌شوند.

[Downloaded from jots.knu.edu.af on 2026-07-04]

دو مقالهٔ [۷۴،۷۵] روش خودکار مبتنی‌بر ML را برای تخمین تلاش نرم‌افزار براساس متن کار معرفی کردند. یک ANN برای ساده کردن توابع تخمین تلاش استفاده می‌شود. ارزیابی نرم‌افزار SPM از یک شرکت نرم‌افزاری، نتایجی را به دست می‌آورد که فراتر از ادبیات مربوطه است و سیستمی که قول می‌دهد ادغام آن با هر ابزار نرم‌افزاری SPM که شرح وظایف متنی را ذخیره می‌کند، بسیار آسان‌تر باشد، اساساً بر توصیف متنی وظایف متکی است که برخلاف روش‌های مختلف دیگر تقریباً همیشه در دسترس هستند.

در مقالهٔ [۷۶]، نویسندگان نتیجهٔ بازخورد معیارهای اجتماعی کار داده‌کاوی کاربردی، تخمین تلاش، تولید مورد آزمایشی و... را نشان دادند؛ سپس نتایج آن تحلیل غیررسمی در ۷ اصل و ده‌ها نکتهٔ دیگر رسمیت یافت و نظام‌مند شد. هدف، تشریح رویکردهایی برای نتایج موفقیت آمیز داده‌کاوی صنعتی است، اما اضافه کنیم که برخی از این اصول برای داده‌کاوی دانشگاهی صادق هستند.

تحقیق در مورد [۷۷-۷۹] یک مدل هیبریدی جدید را به‌درستی ایجاد کردند؛ این مدل حتی برای طیف وسیع‌تری از فعالیت‌ها ایده‌آل است، زیرا در یک پایگاه داده قابل استفاده است. دو الگوریتم ML، یعنی ANN و SVM عملکرد مدل ما را بررسی می‌کنند. آزمایش‌ها، نسخهٔ قوی‌تری نسبت به مدل پیش‌بینی خطر SVM ما را نشان می‌دهند.

بقیهٔ مقالات [۸۰-۸۲] اغلب، شامل توضیحات مبهم برای بحث در مورد ابهام موقعیت و ملاحظات زبانی برای بهبود واکنش فنی به روش‌های مدیریت ریسک پروژه است. سیاست فعلی برای کمک به کاهش و سرمایه‌گذاری برق اِعمال می‌شود. اختصاص پروژه و عملکرد روش برنامه‌ریزی‌شده براساس پارامترهایی مانند عدد صحیح، میانگین اشتباه مطلق، منبع و اشتباه مطلق نسبی محاسبه و تجزیه‌وتحلیل می‌شود.

نویسندگان مقاله‌های [۸۳-۸۵] چهارچوبی آزمایشی از معیارهای کد منبع ارائه کردند و بهترین مجموعهٔ متریک را برای عملکرد مدل انتخاب کردند. روش برآورد هزینه برای آزمایش مدل‌های شکست پیش‌بینی‌کننده استفاده می‌شود. هدف، همچنین رفع این محدودیت‌ها با بستن فاصلهٔ بین نتایج آزمایش تجدیدنظرشده و اجرای بالقوه در فعالیت MLهای کارآمد در چرخهٔ عمر طرح اولیهٔ رشد پروژه بود.

سایر مقالات [۸۶-۸۸] پیشنهاد کردند که سیستم انتخاب مدل‌های نرم‌افزاری توسط پروژه استفاده می‌شود و با استفاده از اثبات مهندسی نرم‌افزار تاریخی، مدل فراین نرم‌افزاری را انتخاب می‌کند که برای پروژهٔ فعلی در مراحل اولیهٔ توسعه مناسب است. نویسندگان مقالهٔ [۸۹] معماری‌ای ارائه می‌کنند که مدل‌های تجزیه‌وتحلیل خودکار شکست را با استفاده از الگوریتم‌های طبقه‌بندی ML برای آزمایش نتایج حاصل از تکنیک‌های مختلف برای فایرفاکس و Netbeans توسعه می‌دهد. آن‌ها نشان می‌دهند که چگونه مدل‌های پیش‌بینی خودکار در

[Downloaded from jots.knu.edu.af on 2026-07-04]

تقریب این دو پارامتر به‌طور واقعی‌تر از انواع خطوط پایه برای حالت‌ها و پروژه‌های تلفات خاص کارآمدتر هستند.

این بخش مقالاتی را توصیف می‌کند که از شبکه‌های عصبی مصنوعی (ANN) استفاده کرده‌اند. دو مقالهٔ [۹۰،۹۱] در مورد ایجاد محرک خطر سیستم‌های ML، مناسب‌ترین مشوق‌های ریسک را در مورد الزامات، سناریوها و برچسب‌های طبقه‌بندی برای ایجاد پروژهٔ نرم‌افزاری ارائه می‌کنند. مطالعه باید مستقل از هریک از این طبقه‌بندی‌ها مشاهده شود، زیرا طبقه‌بندی‌ها مستقل از علل خطر هستند. نویسندگان مقالات [۹۲،۹۳]، در مورد کاربرد برنامه‌های ANN محور و بهینه‌شدهٔ تیم توسعهٔ نرم‌افزار برای تشخیص شکاف‌های ظرفیت و آماده‌سازی برنامه‌ریزی و زمان‌بندی مهارت‌های SPM، تسهیل پیش‌بینی و پیش‌بینی استعدادها براساس فناوری‌های هوشمند کاربردی، بیان منابع و تکنیک‌های توسعهٔ کارکنان صحبت کردند. یکی از اهداف کلیدی مقالهٔ [۹٤] کمک به پیش‌بینی SCE با استفاده از فراین یادگیری ANN فعلی است. اثرات میانگین ریشه و بزرگی متناسب متوسط خطا هستند.

دسته‌بندی دیگری از مقاله‌ها، ماشین بردار پشتیبان (SVM) را مطالعه می‌کند. دو مقاله، کار سیستم‌های Gating پروژه و استفاده از آن‌ها برای تمییز دادن خط ساختمان را توضیح می‌دهند. دستهٔ اولیه برای سیستم‌های CI هستند [۹۵،۹۶]. سه اکتشاف برای رسیدگی به ارسال‌ها برای سیستم Gating پیشنهاد و بررسی می‌شوند. نشان داده می‌شود که این نتیجه از استفاده از غربالگری با نرخ موفقیت بالا و نظارت مستمر در سطح پایین است که نتیجه‌ای قوی است. سومین و آخرین ارزیابی اکتشافی اهرم ML را برای بهینه‌سازی انتخاب آزمایش می‌کند. مقالات دیگر [۹۷،۹۸،۹۹،۱۰۰]، مدل ارزیابی ریسک پروژهٔ مربع کمتر (LS-SVM) را توسعه داده‌اند. شبیه‌سازی نشان می‌دهد که نتیجهٔ پیش‌بینی‌شده SVM موفقیت‌آمیز است. برای تحلیل مدل ارزیابی ریسک پروژه از رویکرد LS-SVM استفاده شد. داده‌های ارزیابی ریسک متخصص برای آموزش مدل رگرسیون LS-SVM برای ارتباط نقشه‌برداری بین خطر و ویژگی‌ها استفاده می‌شود. همچنین یافته‌ها دقت و تعمیم قوی مدل LS-SVM را نشان می‌دهند. آخرین مقاله [۱۰۱] پیشنهاد کرد که مراحل باید با روش‌های یادگیری SVM در زمان اجرا تجزیه‌وتحلیل و درجه‌بندی شوند. مکانیسمی را تعریف کرده است که با انتخاب معیارها از برنامهٔ موجود یا تغییر جهت نرم‌افزار اندازه‌گیری‌ها، تقویت برنامه‌های محاسباتی با معیارهای تطبیق‌پذیری مشخص می‌شود.

مقالات این بخش، جنگل تصادفی را توصیف می‌کنند. دو مقالهٔ [۱۰۲،۱۰۳] مدل پیش‌بینی بسیار قابل اعتمادی ایجاد کرده‌اند. در مرحلهٔ ایجاد نرم‌افزار در حال انجام و ابتکار تحقیق، رویکرد ذکرشده در مجلهٔ عملی باید شامل پیش‌بینی نقص باشد. نتایج تدریس را با دقیق‌ترین پیش‌بینی‌کنندهٔ خطا در بخش در طبقه‌بندی‌های متعدد ازجمله NB، DT، یا RF مقایسه و دوباره بررسی کنید. در مقالهٔ [۱۰٤] نتایج ارزیابی فعالیت تیمی کامل که شامل بیش از ٤۰

[Downloaded from jots.knu.edu.af on 2026-07-04]

مرحلهٔ عینی و قابل مشاهده است که توسط گروه‌های دانش‌آموزی که در طرح‌های کلاسی همکاری می‌کنند، انجام شده است. همچنین، چهارچوب ML از الگوریتم RF برای پیش‌بینی رفتار هم‌تیمی‌ها و نتایج تیم استفاده می‌کند.

فقط یک مقاله در شبکه بیزی انجام شد. مقالهٔ [۱۰۵] از پایگاه‌های اطلاعاتی متعددی برای جمع‌آوری معیارهایی استفاده کرد که از مشخصات طراحی برای سه برنامهٔ جداگانهٔ ناسا که برای ابزار فضاپیما، چهارچوب پیش‌بینی زمین در زمان واقعی و برنامه‌های کاربردی ماهواره‌های پرواز ساخته شده‌اند، گرفته شده‌اند. کاربرد BN در مهندسی نیازمندی‌ها را بررسی کرده و به‌طور خاص بر شناسایی و ارزیابی الزامات پرخطر تمرکز می‌کند.

دو مقاله در مورد K-Nearest Neighbor (k-NN) انجام شده است. مدل پیشنهادی نشان داده‌شده در مقالهٔ [۱۰۶] مکان‌های مختلفی به مدیران پروژه را ارائه می‌کند تا بهترین سایت‌های تولید جهانی را برای وظایف فردی انتخاب کنند. مدل تخصیص شغل پیشنهادی نیز برای سایر رویکردها ارزیابی و بررسی می‌شود. دومین الگوریتم ترکیبی در مقالهٔ [۱۰۷] وجود دارد که ترکیبی از بهینه‌سازی الگوریتم‌های COA-Cuckoo و KNN است. یافته‌ها نشان می‌دهند که هزینهٔ پیش‌بینی‌شده قابل اعتمادتر است.

دو مقالهٔ دیگر طبقه‌بندی در مورد درختان تصمیم انجام شده است. در مقالهٔ [۱۰۸]، متغیر گسسته‌ای پیشنهاد شده است و الگوریتم مدل طبقه‌بندی معرفی شد. یافته‌ها نشان می‌دهند که درخت‌های سیاست دقیق آماری از رگرسیون لجستیکی شرطی تکاملی و استاندارد پیشی می‌گیرند. دومین مقاله [۱۰۹]، همگنیِ داده‌های هزینه را در حوزه‌های دستگاه، تجزیه‌وتحلیل می‌کند و بر حس جاسازی تمرکز می‌کند. مدل‌های داده بین‌دامنه‌ای Equatinf با مدل دادهٔ دامنه، سه نصب آزمایشی ایجاد می‌کنند.

مقاله‌ای که روی الگوریتم ژنتیک انجام شد [۱۱۰] پیشنهاد کرده است که از NN برای ایجاد فهرستی از متخصصان سازماندهی‌شده برای هر معیار استفاده شود. ترکیبی از رویکردهای برنامه‌ریزی تخلیهٔ نیمه‌خودکار و تخصیص موقعیت نیمه‌خودکار نیز اجرا شده است. نتیجهٔ انباشتهٔ برنامه‌ای تکراری برای جزئیاتی است که سازنده روی آن کار می‌کند.

مقاله‌ای در مورد یادگیری عمیق انجام شد. در مقالهٔ [۱۱۱] دو معماری یادگیری عمیق قابل اعتماد وجود داشت؛ شبکهٔ بزرگراه[1] و حافظهٔ دوربرد[2]. چهارچوب پیش‌بینی شامل آموزش سرتاسری از داده‌های ورودی ابتدایی تا اثرات پیش‌بینی بدون مهندسی عملکرد دستی است. بررسی تحلیلی نشان می‌دهد که میانگین مطلق بودن، مطلق بودن میانه و دقت یکنواخت سه خط پایهٔ معیار و شش گزینه به‌طور قابل اعتمادی بهتر عمل می‌کنند.

[1] highway network
[2] long-range memory

[Downloaded from jots.knu.edu.af on 2026-07-04]

۳-۷-۲. مطالعات انجام‌شده روی سایر روش‌ها

در این بخش، روش‌های دیگر و نحوهٔ استفاده از آن‌ها بررسی می‌شود؛ این مقالات به موضوعات و کاربردهای مختلفی تقسیم شدند.

مدل اول، مدل‌های فراین نرم‌افزار است. دو مقالهٔ [۱۱۲،۱۱۳] تحقیقات همکاری پژوهشی در برنامه‌های چابک (agile) را بررسی کردند. نمونه‌ها به شناسایی تضادهای بالا ازجمله جایگزین‌های کوتاه‌مدت برای تعدیل‌های کوتاه‌مدت و حل‌وفصل اختلافات منافع با استفاده از تکنیک‌های چابک کمک می‌کنند. دو مقالهٔ [۱۱۴،۱۱۵] گردآوری پیشنهادها را تصریح کرده‌اند. ویژگی‌های انتخاب‌شده و پیاده‌سازی کیت ML برای مفهوم آماری R را ارزیابی کردند؛ به‌علاوه، توضیح داده شده است که چگونه تکنیک‌های داده‌کاوی را می‌توان برای ساخت مدل طبقه‌بندی پیش‌بینیِ تلفات استفاده کرد. مدل پیشنهادی، داده‌ها را از چند پارامتر پروژه جمع‌آوری می‌کند و پروژه‌ای را به یکی از سه کلاس، طبقه‌بندی می‌کند. مقالهٔ [۱۱۶]، رویکردی برای ارزیابی ریسک پروژهٔ نرم‌افزاری پیشنهاد کرده است که از اعتبارات برای اندازه‌گیری اثرات عوامل خطر جهت رفع نظرات نادقیق و تناقضات بین کارشناسان استفاده می‌کند. استنتاج شبکهٔ اعتباری به پیش‌بینی و تشخیص خطر کمک کرد.

در مقالهٔ [۱۱۷]، به اهمیت روش‌های چابک و سیستم‌های پیشرفته مانند اینترنت اشیا، مه[1] و ابر[2] پرداخته شده است؛ بنابراین، نرم‌افزاری از ادغام ساختار طراحی و کاهش ریسک برای پیگیری بهتر این هدف پیشنهاد شد. نوآوری‌های ML که ثابت می‌کند برای گام‌های فعلی در ارزیابی ریسک کسب‌وکار به حالت مستمر مطلوب‌تر هستند و با اینترنت اشیا پیاده‌سازی می‌شوند نیز در چهارچوب پیشنهادی گنجانده شده‌اند. نویسندگان مقالهٔ [۱۱۸] به‌طور سیستماتیک سیستم کمیتی قوی یکپارچهٔ مدیریت ریسک را ارائه می‌دهند که بر مشارکت بین اندازهٔ پروژه و مشارکت تصمیمات مخاطره‌آمیز تمرکز دارد. سازندگان این رویکرد، تحلیلی در پایگاه دادهٔ پروژه‌ای واقعی ازطریق برنامه‌ای کاربردی نشان دادند.

مدل دوم، کنترل‌کنندهٔ منطق فازی است. در دو تا شش مرحلهٔ برآورد تلاش نرم‌افزار، دو مقالهٔ [۱۱۹،۱۲۰] افزایش راندمان کنترل‌کننده‌های روان و منطقی را با بهبود کنترل‌های سوخت مبتنی‌بر مورد با حداقل اندازه براساس کنترل پیشنهاد کرد؛ نتیجه، آبشار کنترل‌کنندهٔ منطق سیال است. قوانین مدل‌هایی که توسط خوشه‌بندی تفریقی ایجاد می‌شوند، امکان کاهش بیشتر را فراهم می‌کنند.

مدل سوم، برنامه‌ریزی انتشار استراتژیک (SRP) است. نویسندگان مقالهٔ [۱۲۱] دریافته‌اند که نرم‌افزار مورد استفاده به‌عنوان افزونه‌ای برای چهارچوب‌های تولیدی، که اغلب استفاده می‌شوند، به بهبود عملکرد فراین کمک می‌کند. SRP، مرحله‌ای مهم در ایجاد برنامه‌های

[1] Fog
[2] Cloud

[Downloaded from jots.knu.edu.af on 2026-07-04]

کاربردی تکراری است. SRP شامل تخصیص ویژگی‌ها یا شرایط برای انتشار در قالب محدودیت‌های سخت و نرم مانند زمان، تعهد، ثبات و پول است.

مدل چهارم، مدل بلوغ است. در مقالهٔ [۱۲۲]، داده‌های بدون ساختار عظیم تحت چهارچوب مدل بلوغ شایستگی دیجیتال (DCMM) ازطریق تجزیه‌وتحلیل دقیق هدف، فرایندهای مدیریتی یا عوامل تأثیرگذار این پروژه توسعه یافته‌اند. پیش‌بینی می‌شود که مقایسهٔ مهارت‌های ذخیره‌سازی داده‌های مرتبط، تأثیر مطلوبی داشته باشند. روشی تثبیت‌شده جهت بهبود عملکرد و استانداردی طلایی برای ایجاد برنامه و دستگاه برای بیش از ۲۰ سال ساخته شده است.

مدل پنجم، وزن دهی متوالی رو به جلو است. نویسندگان مقالهٔ [۱۲۳]، الگوریتم‌های کارآمدی را برای تعمیم انتخاب زیرمجموعهٔ ویژگی‌ها با رویکرد وزن‌دهی ویژگی عملی، توسعه و ارزیابی کردند. این الگوریتم، دقت را بیشتر بهبود می‌بخشد، زیرا همهٔ ویژگی‌ها به‌طور یکسان در حل مسئله برای تعیین وزن عناصر جهت بهبود دقت تخمین کمک نمی‌کنند. پس از آن، آزمایش‌هایی را انجام داده‌اند که براساس طراحی اندازه‌گیری‌های مکرر مجموعهٔ داده‌های دنیای واقعی برای ارزیابی این الگوریتم‌ها است.

مدل ششم، مدل تخمین استدلال مبتنی‌بر مورد (CBR) است. نویسندگان مقالهٔ [۱۲٤] پیشنهاد کردند که سیستم‌های خوشه‌بندی CBR (CBR-Cs) را می‌توان برای ارائهٔ برآورد هزینه‌ای دقیق ایجاد کرد. هدف رویکرد CBR-C، تعیین کمیت خطاها، زمان و اجازه به مدیران برای درک آسان فرایندهای ارزیابی است. این مطالعه نشان می‌دهد که رویکرد CBR-C پیشنهادی، ساختار برآورد هزینهٔ پروژهٔ طراحی نرم‌افزار جامعی را ارائه می‌دهد.

مدل هفتم، مدل زمین-ماه[1] است. نویسندگان مقالهٔ [۱۲۵] مدل مدیریت ریسکی برای پروژه‌های ماه-زمین ارائه کردند که ویژگی‌های ایجاد نرم‌افزار را مطابق با تئوری پیاده‌سازی‌شدهٔ پروژهٔ چرخهٔ عمر نرم‌افزار در نظر گرفت. تحولات اصلی، که در پارادایم کنونی مورد استفاده قرار می‌گیرد، اغلب تجزیه‌وتحلیل و حل می‌شوند.

مدل هشتم، پیش‌بینی نقص پروژه است. در مقالهٔ [۱۲٦]، اثرات ساده‌سازی داده‌ها توصیف و کمی‌سازی شد. آزمایش‌ها انجام شدند و با و بدون توانایی پیش‌بینی CPDP مقایسه شده‌اند. روشی برای ساده‌سازی داده‌ها با استفاده از روش یادگیری تطبیقی برای محاسبات تعامل کاربر معرفی شد.

مدل نهم، Treeboost (تقویت گرادیان تصادفی) است. ورودی‌های مدل، طبق گزارشی در مقالهٔ [۱۲۷]، مقیاس، کارایی و پیچیدگی Treeboost هستند. مدل Treeboost با استفاده از چهار معیار خروجی MMRE، PRED، MdMRE و MSE در برابر مدل رگرسیون

[1] Earth-Moon

[Downloaded from jots.knu.edu.af on 2026-07-04]

چندخطی و مدل موردی آزمایش شد و به‌عنوان مدل رگرسیون چندخطی ایجاد شد. مدل Treeboost را می‌توان برای کمی کردن تعهد خروجی نرم‌افزار استفاده کرد.

٤. بحث

این تحقیق به بررسی مطالعات ضروری در مدیریت پروژهٔ پیشرفته با استفاده از فناوری ML می‌پردازد. هدف این تحلیل تأکید بر الگوهای پژوهشی در این زمینه است.

این تحقیق فعلی نیست و اجرا را پوشش نمی‌دهد، بلکه خود ادبیات را پوشش می‌دهد. این مطالعه با برآوردهای قبلی متفاوت است؛ ادبیات همراه به‌عنوان طبقه‌بندی پیشنهاد شده است. در زمینه‌ای تحقیقاتی، ایجاد طبقه‌بندی ادبیات ممکن است چندین مزیت داشته باشد.

سؤال سوم تحقیق: شکاف‌های موجود برای چشم‌انداز تحقیق در زمینهٔ مدیریت پروژهٔ نرم‌افزاری چیست؟

محقق جدیدی، که در حال تحقیق ارزیابی پروژهٔ نرم‌افزاری است، ممکن است تحت‌تأثیر طیف گستردهٔ اسناد در این بخش، فقدان نوع خاصی از چهارچوب و بررسی این زمینه قرار گیرد؛ مقالات متعددی در مورد این موضوع به روندهای نوظهور در مدیریت پروژه می‌پردازند. امروزه، مطالعات خاصی، مدل‌ها و پیاده‌سازی‌های ML را ایجاد کرده‌اند. طبقه‌بندی ادبیات به‌طور قابل توجهی به سازماندهی این آثار و حوادث متنوع کمک می‌کند و قابل استفاده و سازگار است. از سوی دیگر، روش‌شناسی طبقه‌بندی، بینش مفیدی را در مورد موضوع به محققان می‌دهد. در مرحلهٔ بعد، موضوعات آینده برای مطالعه را شرح می‌دهد و طبقه‌بندی در تجزیه‌وتحلیل فعلی ارزیابی نرم‌افزار نشان می‌دهد؛ برای مثال، به نظر می‌رسد محققان، مسیری را در این بخش برای درگیر کردن مکانیسم‌هایی برای توسعه و عملیات برنامه توصیه می‌کنند. استفاده و پیاده‌سازی فناوری‌های ML ازجمله آخرین ارزیابی پروژه نیز پوشش داده شده است.

همچنین، طبقه‌بندی می‌تواند کمبودهای مطالعه را طبقه‌بندی کند و نگاشت ادبیات، پوشش مطالعاتی ضعیف و قوی را در پیشنهادهای بررسی پروژه در دسته‌های مختلف نشان می‌دهد؛ به‌عنوان مثال، طبقه‌بندی، ارزش بررسی و ارزیابی گروه‌های ادعاهای فردی را به هزینهٔ روش‌ها، ساختارهای تلفیقی و فعالیت‌های رشد نشان می‌دهد که در فراوانی دسته‌های آن‌ها بیان می‌شود. طبقه‌بندی، عدم تحقیق دربارهٔ توسعهٔ بازبینی پروژه را پس از تحقیقی کافی نشان داد. ادبیات، برای تحقیق ضروری است. مطالعات این بخش با هدف بهبود و تبادل ML انجام می‌شوند.

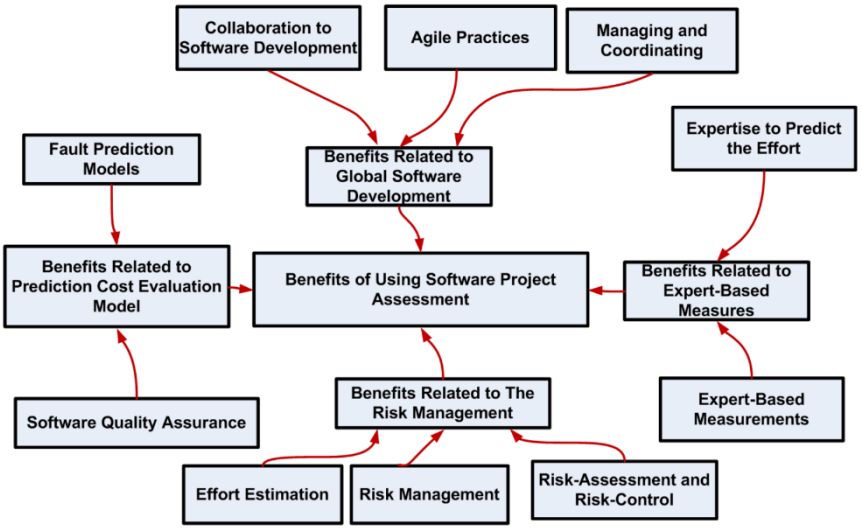
تجزیه‌وتحلیل آماری در بخش‌های جداگانهٔ طبقه‌بندی، صنایع شرکت‌کننده در روش ML را برای مقابله با تحولات در حال ظهور و بهبود زمینه‌های غیرفعال، طبقه‌بندی می‌کند. این مطالعه، طبقه‌بندی‌ای را ارائه می‌دهد که محققان ممکن است در آن با همکاری و تجزیه‌وتحلیل فناوری‌های جدید مانند پیشرفت‌ها، مطالعات مقایسه‌ای و ارزیابی پروژه با استفاده از فناوری ML، که مشابه طبقه‌بندی در زمینه‌های دیگر است، همکاری کنند. همچنین، تجزیه‌وتحلیل

[Downloaded from jots.knu.edu.af on 2026-07-04]

سه بخش از ادبیات را نشان می‌دهد که عبارت‌اند از عوامل پشت ظهور مدیریت پروژهٔ خودکار با استفاده از فناوری‌های ML، چالش‌های استفادهٔ موفق از آن روش‌ها و توصیه‌هایی برای غلبه بر این مشکلات.

٤-١. انگیزه‌ها

مزایای استفاده از پلتفرم مدیریت پروژهٔ ML، شفاف و قانع‌کننده است. این بخش، برخی از مزایای ادبیات بحث می‌کند که براساس مزایای منحصر به فرد طبقه‌بندی می‌شوند. منابع مناسب برای بحث بیشتر ذکر شده است (شکل ٦).



شکل ٦: مزایای استفاده از ارزیابی پروژهٔ نرم‌افزاری

٤-١-١. مزایای مربوط به مدل ارزیابی هزینه پیش‌بینی

این روش برای آزمایش معیارهای کد منبع و معیارهای مناسب برای بهبود عملکرد مدل پیش‌بینی خطا استفاده شد. از روش برآورد هزینه برای آزمایش مدل خطای پیش‌بینی‌کننده استفاده می‌شود. نتایج اولیه عبارت‌اند از [١٢٨]: ١. چندین رویکرد برای رأی‌گیری تحت‌الشعاع روش‌های دیگر؛ ٢. معیارهای مکانیزم کد منبع انتخاب‌شده برای استفاده از معیارهای منبع دستگاه پیشنهادی که در پروژه‌های نرم‌افزاری با درصد کلاس‌های خطای کمتر از مقدار آستانه توصیه شده، در مقایسه با سایر روش‌ها مفید هستند؛ ٣. رویکرد پیش‌بینی شکست.

٤-١-٢. مزایای مدیریت ریسک

فعالیت‌های مختلف برنامه‌ریزی نرم‌افزار را می‌توان به دو روش مؤثر تقسیم کرد؛ ارزیابی مشارکت و کاهش ریسک [١٢٩]. برآورد هزینهٔ تلاش نرم‌افزار به بسیاری از ملاحظات هزینه بستگی دارد و کنترل ریسک به شناسایی، درمان و حذف ریسک نرم‌افزار قبل از نتایج غیرمنتظره نیاز دارد [١٣٠]. تعهد پیش‌بینی‌شده به رشد نرم‌افزار، تلاش‌ها را تعیین می‌کند. ارزیابی ریسک،

[Downloaded from jots.knu.edu.af on 2026-07-04]

فعالیت اصلی در مرحلهٔ برنامه‌ریزی پروژه [۱۳۱]، عنصری حیاتی در ارزیابی موفقیت پروژهٔ توسعهٔ نرم‌افزار است؛ بااین‌حال، قضاوت و تجربهٔ انسانی در موفقیت فعالیت‌های مدیریت ریسک در روش‌های سنتی کاهش ریسک، ضروری است و ارزیابی ریسک برای پروژهٔ نرم‌افزاری، غیرضروری و پرهزینه در نظر گرفته می‌شود [۱۳۲].

٤-١-٣. مزایای توسعهٔ جهانی نرم‌افزار (GSD)

افزایش استفاده از GSD برای به حداقل رساندن هزینه‌های تولید و باز بودن بستهٔ گسترده‌ای از تجزیه‌وتحلیل حرفه‌ای، یکی دیگر از پیشرفت‌های حیاتی در بازار است. سرمایه‌گذاری‌های GSD اغلب چالش‌های مهمی ایجاد می‌کنند، اما محبوب‌تر می‌شوند؛ این چالش‌ها شامل نگرانی‌های ارتباطی بین شرکت‌کنندگان پروژه، مشکلات در ایجاد ارتباطات قابل قبول جامعه، مشکلات فرهنگی و موانع در مدیریت و سازماندهی کار در پروژه‌های اجرایی هستند. به‌طور خلاصه، کار تیمی فشرده در تولید نرم‌افزار در محیط توزیع‌شده، دشوار است [۱۳۱]. با تمرکز بر ارتباطات چهره‌به‌چهره، که برای مدیریت زمینه‌های GSD سخت و پیچیده است، ادغام برخی از رویکردهای چابک از همان اولین تلاش، چالش‌برانگیز به نظر می‌رسد.

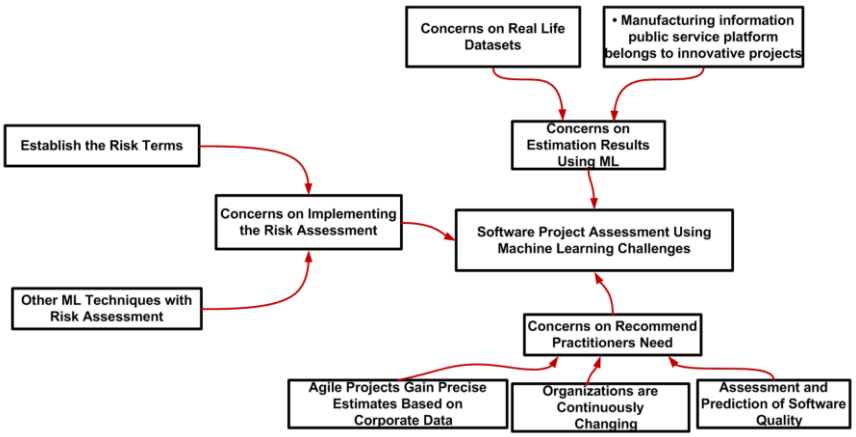
٤-١-٤. مزایای اقدامات مبتنی‌بر خُبره

پیش‌بینی‌کننده از تجربهٔ خود برای پیش‌بینی کار سرمایه‌گذاری‌هایی مانند کارشناسان استفاده می‌کند. مهارت برآوردگر به موضوع و تجربهٔ او از اقدامات مشابه و متعارف بستگی دارد. اگر تعداد محدودی از مراحل، نیاز به حذف اندازه‌گیری‌های متخصص داشته باشد مدل متمرکز، سود قابل‌توجهی خواهد داشت. نباید از جایگاه رفتار متخصص غافل شد. مفهوم ارزیابی ساخته‌شده بدون معیارهای حرفه‌ای حتی بدتر از مدل سازگاری و رویکرد تضمین کیفیت مورد استفاده برای جلوگیری از ارزیابی‌هایی است که نیاز به تحقیقات تخصصی دارند؛ پژوهش مدل ارزیابی، محدودیت‌های زیادی را در جمع‌آوری معیارهای پیش‌بینی برای کار بالقوه ارائه می‌دهد. اشکالات بالقوه ممکن است تعداد ابزارهای خاص موجود برای ارزیابی رفتار، سازگاری و پایبندی به کاربرد یک سیستم یا دقت اندازه‌گیری باشد [۱۳۲].

٤-٢. چالش‌ها

اگرچه تکنیک‌های آموزشی کامپیوتری، که در ارزیابی SPM استفاده می‌شوند، چندین مزیت را ارائه می‌دهند، اما چنین فناوری‌هایی راه‌حل ایده‌آل برای پروژه‌های در حال تکامل در نظر گرفته نمی‌شوند [۱۳۳]. بررسی‌ها نشان می‌دهند که محققان به ارزیابی پروژه‌ها و استفادهٔ آن‌ها از استراتژی‌های ML علاقه‌مند هستند. موانع اصلی در اجرای تکنیک‌های ML، علاوه‌بر موضوعات اضافی، در زیر ذکر شده‌اند و مشکلات تعریف شده‌اند (شکل ۷).

[Downloaded from jots.knu.edu.af on 2026-07-04]



شکل ۷: ارزیابی پروژهٔ نرم‌افزاری با استفاده از چالش‌های یادگیری ماشین

٤-۲-۱. نگرانی از نتایج تخمین با استفاده از ML

تجزیه‌وتحلیل، سؤالات بیشتری را دربارهٔ مجموعهٔ داده‌های واقعی که فاقد رویکردهای توسعهٔ نرم‌افزار اساسی هستند و به معیارهای دقیق دیگری نیاز دارند، که می‌تواند برای محاسبهٔ تلاش استفاده شوند، آشکار کرد. رویکردهای ارزیابی مختلفی برای تأیید یافته‌های پیش‌بینی تلاش برنامه در تحقیقات بیشتر مورد نیاز است. اعتبارسنجی متقاطع، رویکرد Jacknife و روش تکراری عمدتاً اعتبارسنجی می‌شوند. علاوه‌بر این، روندهای مطالعاتی نشان داده‌اند که روش‌های محاسبه به بررسی و تقویت نیاز دارند. همچنین، مجموعهٔ داده‌های واقعی ممکن است معیارهای اندازه و سایر روش‌های ML ازجمله درخت‌های رگرسیون را نیز بررسی کنند. پورتال خدمات عمومی ازجملهٔ ابتکارات خلاقانه در زمینهٔ تولید دانش است. رویه‌های توسعه باید بیشتر توسط تیم پروژهٔ ما به‌روز شوند و مهندسان نمی‌توانند پلتفرم توسعه و زیرساخت‌هایی را که تأثیر بیشتری بر کارایی محصول دارد نظارت کنند.

٤-۲-۲. نگرانی از اجرای ارزیابی ریسک

زمانی که پروژه شروع می‌شود، ابتدا باید تهدیدات و عوامل خطر پروژه را تعریف و شرایط ریسک اصلی را تعیین کنیم؛ یعنی تغییر تقاضای ریسک، خطر زیرساخت، هماهنگی افراد و رویکردهای حفاظت از دستگاه و... . شبکهٔ ارزیابی ریسک، همراه با تخصص کارشناسان، روی موارد فعلی برای ایجاد فراین یادگیری مورد ریسک متمرکز شده است. در پرتوی دو سیستمی که بالاترین نتایج موفقیت را به دست آورده‌اند، ارزیابی درجهٔ ریسک، شرطی ضروری بود که هزینه‌های تجدیدنظر در سطح ریسک آن به مناسب‌ترین چهارچوب برای طبقه‌بندی بیزی تبدیل شد. در این زمینه توصیه می‌شود، زیرا برای سایر تکنیک‌ها مانند ML، نتایج تخمین بهبودیافته را ارائه می‌دهد [۱۲]. دلیل نسبتاً قطعی‌تری برای دقت بالاتر وجود دارد، زیرا داده‌ها

[Downloaded from jots.knu.edu.af on 2026-07-04]

از مجموعهٔ داده‌ای وجود واحد به دسته‌های مختلف همگن، بسته به شرکت یا بخش، استخراج می‌شوند. دقت برآورد تلاش ازطریق طبقه‌بندی مقالات [۱۳۴،۱۳۵،۱۳۶] افزایش یافت.

٤-۲-۳. نگرانی از نیاز به توصیهٔ پزشکان

در مورد نوع روش‌های محاسبه‌ای و نوع مجموعهٔ داده‌هایی که برای پروژه‌ها استفاده می‌شوند، مراقب باشید. پروژه‌های چابک برآوردهای دقیق‌تری را براساس داده‌های شرکت به دست می‌آورند. در مقایسه با داده‌های ملی، کسب و کارهای خصوصی از مقدار محدودی از داده‌های پروژهٔ داخلی سود می‌برند [۱۳۲]. همانطور که شرکت‌ها روش‌های رشد تدریجی مولدتر و چابک‌تری مانند XP را پیاده‌سازی می‌کنند، گزارش‌های بیشتری در مورد پروژه‌هایی که این تکنیک را به کار می‌گیرند، پیشنهاد می‌شود. مطالعات فعلی مدیریت تلاش نسبتاً بهتری را در پروژه‌های XP نشان می‌دهد، همانطور که علی‌رغم حفظ همان بهره‌وری با MMRE کم نشان داده شده است [۱۳۷].

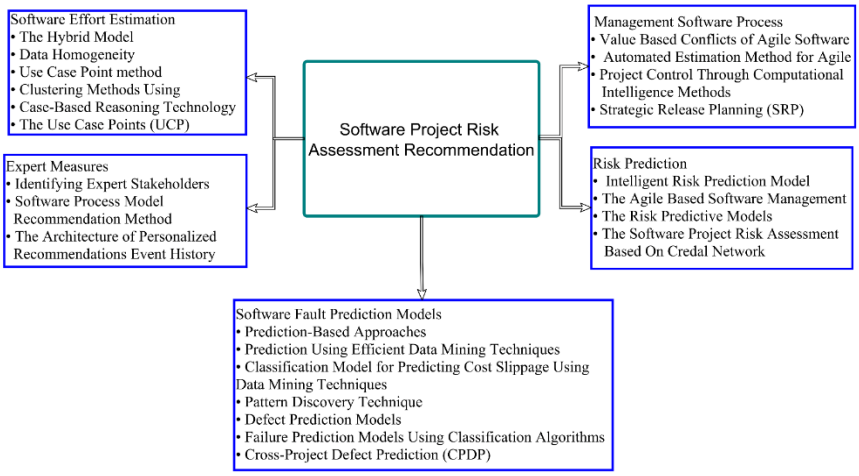
از آنجایی که سازمان‌ها به‌طور مداوم در حال تغییر هستند، سیستم‌های کنترل پروژهٔ تطبیقی نیاز خواهند بود. KPIهای جدید برای استفاده یا حذف و تغییر قوانین ارزیابی متخصص، سناریوهای رایج بین مدیران پروژه هستند. PMIS باید چنین شرایطی را مدیریت کند. ایجاد فرایندهای خودکار و کاملاً تعریف‌شده بین سطوح بالای مدیریت سازمان و تنظیمات داخلی PMIS، نویدبخش ویژگی‌های رابطه‌ای جدید در روشی هستند که این دو نهاد بین آن‌ها ارتباط برقرار می‌کنند [۱۳۸].

با توجه به ارزیابی و پیش‌بینی کیفیت نرم‌افزار در سازمان‌های بزرگ نرم‌افزاری، پیش‌زمینهٔ کلی استفاده از روش‌های ML در شرکت‌های بزرگ محاسباتی، ارزیابی و برآورد کیفیت محصول است. شاخص‌های مختلف محصول را می‌توان در ایجاد مدل کیفیت نرم‌افزار با استفاده از ISO 15939 برای اندازه‌گیری و پیش‌بینی خروجی نرم‌افزار و همچنین برآورده کردن معیارهای اطلاعات کیفی این سازمان‌ها استفاده کرد. سودمندی روش‌های ML حتی برای چنین ارزیابی‌ای مستند شده‌اند. در ISO 9126 [۱۳۹]، کیفیت به‌عنوان «کامل بودن ویژگی‌ها و ویژگی‌های محصول نرم‌افزاری که قادر به برآورده کردن نیازهای اعلام‌شده یا ضمنی است»، تعریف می‌شود. در ISO 25000 [۱٤۰]، ظرفیت محصولات نرم‌افزاری برای برآورد کردن مشخصات مشخص‌شده، تحت برخی شرایط رویکرد سازگاری اتخاذ شده است. شناسایی و تخصیص منابع جایی که بیشتر نرم‌افزار برای ارزیابی کیفیت نرم‌افزار استفاده می‌شود، در مراحل اولیهٔ توسعه ضروری است [۱٤۱].

٤-۳. توصیه‌ها

این بخش توصیه‌هایی را برای حل مسائل و چالش‌های ارزیابی SPM مورد استفاده در تکنیک ML ارائه می‌دهد (شکل ۸).

[Downloaded from jots.knu.edu.af on 2026-07-04]



شکل ۸: دسته‌بندی توصیه‌ها برای استفاده از ارزیابی پروژهٔ نرم‌افزاری

٤-۳-۱. توصیه‌هایی برای تخمین تلاش نرم‌افزار

اندازه‌گیری مشارکت نرم‌افزار برای برنامه‌های نرم‌افزاری موفق، ضروری است. اگر بخواهیم در کسب‌وکارهای نرم‌افزاری به‌دنبال بودجهٔ خاصی باشیم، آماده‌سازی و هزینهٔ پروژه به‌درستی انجام می‌شود و تخمین تلاش محاسباتی دقیق، ضروری است. با توجه به تغییرات قیمت، می‌توان از سودهای تجاری بیش از حد برآوردشده را صرف‌نظر کرد. از سوی دیگر، دست‌کم گرفتن، برنامه و هزینه‌ها را تحت‌الشعاع قرار می‌دهد و برای کسب‌وکار، سرمایه هزینه می‌کند. از آنجایی که هزینهٔ تلاش، منعکس‌کنندهٔ هزینهٔ بالا است، ادبیات، در عوض، از محاسبهٔ تلاش نرم‌افزاری و اصطلاحات محاسبهٔ هزینهٔ نرم‌افزار استفاده می‌کند که به ارزش تقریبی اشاره دارد.

سؤال چهارم تحقیق. معیارهای پیش‌بینی و سطح دقت فعلی آن‌ها که توسط تکنیک‌های تخمین مختلف اثبات می‌شود، چیست؟

مدل ترکیبی، برای تخمین تکمیل فعالیت‌های برنامه در مدت طولانی، زمینهٔ مطالعاتی ضروری است؛ بنابراین، ما چندین مدل مستقل با عملکرد خوب را برای بهبود دقت و قابلیت اطمینان اثرات پیش‌بینی می‌خواستیم. نمونهٔ اولیهٔ ترکیبی شامل سه مجموعهٔ مجزا از ویژگی‌ها است که عبارت‌اند از ۱. یکی ویژگی مبتنی‌بر متنی است که به‌عنوان جمع شناخته می‌شود؛ ۲. یکی شامل ویژگی‌های مبتنی‌بر متن است که به‌عنوان تعریف شناخته می‌شوند؛ ۳. دارای انواع ویژگی‌های مبتنی‌بر فراداده است. در مقایسه با مدل‌های مدل قبلی در مقالهٔ [۱۳۷]، این مدل برای تعداد وسیع‌تری از وظایف مناسب است، زیرا تنها به یک نوع منبع داده که همیشه در دسترس نیست، محدود نمی‌شود. این بخش بر تخمین تکمیل فعالیت‌ها متمرکز بود، اما تحقیقات بیشتری برای بررسی اینکه آیا استراتژی‌های ترکیبی مشابه به مدل‌های تخمین زمان تکمیل گسترش می‌یابند یا خیر، مورد نیاز است.

[Downloaded from jots.knu.edu.af on 2026-07-04]

همگنی داده‌ها و همگنی هزینه دامنه کاربرد نتایج. این سؤال قبلاً در ادبیات تخمین ماشین بحث نشده بود. تأثیر مقیاس داده‌های آموزشی موفقیت پیش‌بینی و موضوعی حیاتی که در تحقیق دیگری بررسی شده است را بررسی کنید. مطالعات انجام‌شده برای رسیدگی به این مشکل هنوز تأیید نکرده است که این موضوع به‌دلیل تفاوت در کیفیت داده‌ها و الگوریتم‌های پیش‌بینی، مسئله‌ای بی‌پاسخ است. بااین‌حال، آخرین یافته‌های تجربی، مدیران پروژه را هدایت می‌کند تا تعیین کنند که چه مقدار داده برای آموزش الگوریتم لازم است [۱۰۹]. از روش‌های مختلف ML برای اندازه‌گیری هزینه‌های برنامه و تجزیه‌وتحلیل عملکرد استفاده کنید. همچنین ممکن است سایر محققان را تشویق و هدایت کند تا در این زمینه کار کنند. رویکرد کار بالقوه می‌تواند ویژگی‌های خاص دامنه را هدف قرار دهد، به‌طوری که کیفیت داده‌های ویژگی‌ها بهتر شود و خروجی پیش‌بینی آن‌ها افزایش یابد. این کار ممکن است با تجزیه‌وتحلیل داده‌ها در حوزهٔ دستگاه تعبیه‌شده به دست آمده باشد.

استفاده از روش Case Point مدل Treeboost، تعهد نرم‌افزار را با تمرکز بر سه متغیر مستقل مقیاس، کارایی و پیچیدگی نرم‌افزار پیش‌بینی می‌کند. مدل Treeboost پیش‌بینی عملکرد برنامه را براساس فراین نقطهٔ استفاده پیشنهاد کرد. از مدل Treeboost استفاده شد. اندازهٔ نرم‌افزار، بهره‌وری و پیچیدگی شامل ورودی‌های مدل می‌شوند. مدل رگرسیون چند خطی‌ای ایجاد شد و مدل Treeboost با استفاده از چهار معیار عملکرد MMRE، PRED، MdMRE و MSE در برابر مدل رگرسیون چندخطی و مدل نقطهٔ موردی ارزیابی شد. الگوی Treeboost برای تلاش برنامه با عملکرد مثبت استفاده شد [۲۲]. یافته‌ها مثبت هستند و اندازه‌گیری دقیق تنش اولیه را افزایش می‌دهند.

روش‌های خوشه‌بندی، طیف وسیعی از راه‌های ممکن را دارند. روش‌های خوشه‌بندی اضافی، دانش‌آموزان ساده، ویژگی‌های ورودی پروژه، ویژگی‌های خوشه‌بندی، مقادیر پارامترها و پروتکل‌های تنظیم (خودکار) و همچنین توصیه‌های مکانیزم به‌روزرسانی سازمان‌یافته‌تر برای پروژه‌های CC ممکن است مورد توجه قرار گیرند [۱۴۲،۱۴۳]. Dycom، استفاده از رویکردهای خوشه‌بندی را در ایجاد زیرمجموعه‌های CC گسترش داد. سه روش برای خوشه‌بندی وجود دارد؛ به‌عنوان مثال، سلسله‌مراتب خوشه‌ها، K-means و حداکثر کردن ترجیحات [۱۴۴،۱۴۵،۱۴۶]. Dycom Clustering مشابه Dycom اصلی است که براساس چهار زیرمجموعهٔ CC از چهار جدول SEE با اندازه‌های مختلف است؛ برای مثال، این تحقیق شامل طرح‌بندی توالت است. خوشه‌بندی Dycom مربوط به K-mean به جداسازی برنامه‌های CC کمک می‌کند، با تحویل برابر با Dycom یا کارایی پیش‌بینی بهتری دارد. بااین‌حال، تعداد زیر مجموعه‌های CC نیز نیاز به تعریف قبلی دارد و انتخاب اشتباه ممکن است تأثیر منفی بر نتایج پیش‌بینی داشته باشد.

[Downloaded from jots.knu.edu.af on 2026-07-04]

با استفاده از فناوری استدلال مبتنی‌بر مورد (CBR)، چنین عواملی باعث می‌شوند که CBR تعاملی برای رفع نیازهای مدیران پیشنهاد شود. این روش به‌دنبال ارتقای درک مکانیسم برآورد هزینه و ارائهٔ دانش کافی به مدیران است. درک چرخهٔ CBR ساده است، زیرا ذخیره و بازیابی اطلاعات در حافظهٔ انسان را شبیه‌سازی می‌کند. مدیران، با اقداماتی که انجام داده‌اند بهتر می‌توانند درک کنند که محاسبات چگونه انجام می‌شوند. روش ANGEL با استراتژی CBR-C [۱۴۷] بهتر عمل کرد. توضیح اصلی برای این امر این است که رویکرد CBR-C فعلی دارای تابع نمایه‌سازی‌ای است که نسبتاً خطای پیش‌بینی را کاهش می‌دهد. روش CBR برای غلبه بر مسائل جاری با پرداختن به مشکلات مرتبط در گذشته است. برآورد تعهد CBR به تفصیل بیان شده است. قوانین اساسی CBR به شرح زیر است که قابل مقایسه‌ترین پروژه‌های گذشته برای تخمین اینکه چگونه هزینهٔ پروژهٔ فعلی می‌تواند از معیارهای مقایسه استفاده کند، انتخاب می‌شوند [۱۴۳]. ۱. آیا شکلی از الگوریتم وزن‌دهی تابع برای ارزیابی تلاش استدلال گاه‌به‌گاه وجود دارد که روش‌های فعلی انتخاب تابع را بهبود بخشد؟ ۲. مجموعهٔ داده‌ها چقدر بر این نتایج متکی هستند؟ ۳. آیا می‌توان دقت مجموعهٔ داده‌ها را با جستجوی راه‌هایی برای نشان دادن اینکه یافته‌های ما دقیق یا بدون نویز هستند، اندازه‌گیری کرد؟ ٤. اندازهٔ فاز بررسی وزن با ویژگی‌های یک مجموعهٔ داده را چگونه توصیف کنیم؟

از امتیاز مورد استفاده (UCP) به‌دلیل تقریبی سود آن‌ها در مراحل اولیهٔ رشد محصول استفاده می‌شود. قبل از آموزش مدل‌های جمع‌آوری داده‌ها، استراتژی استانداردسازی پیش‌پردازش اجرا شد. عملکرد تخمینی مدل‌های RF، MLP و SVM ارزیابی شدند [۱۴۸]. متریک UCP برای اندازه‌گیری تعهد با استفاده از نمودارهای موردی زبان مدل‌سازی جهانی (UML) استفاده شد. UCP امکان پیش‌بینی برنامه را در مراحل اولیهٔ فراین طراحی فراهم می‌کند. مقیاس یکنواخت، کارایی و مقادیر دشواری، ورودی‌های حیاتی همهٔ مدل‌ها برای پیش‌بینی آخرین تلاش هستند. فناوری RF و پارامتر ورودی به‌عنوان مرجع قبلی شبیه‌سازی RF برای پیش‌بینی تلاش استفاده شد [۱۴۹]. این تحقیق می‌تواند با افزودن انواع خاصی از ML مانند انواع تقویت درخت مثل XBoost و... گسترش یابد.

٤-۳-۲. توصیه‌هایی برای اقدامات مبتنی‌بر خُبره

دانش، تخصص، تجربه و شهود متخصص براساس دانش گروه‌های پروژه در ۱۰ زمینه مانند قیمت‌ها، زمان، مسافت، بهره‌وری و مدیریت منابع است (اغلب به برآوردهای بیش از حد خوش‌بینانه کمک می‌کند). این نشان‌دهندهٔ مجموعه‌ای از فراینها، سیاست‌ها و شیوه‌هایی است که دانش مورد نیاز برای اجرای پروژه‌ای را ایجاد و جمع‌آوری می‌کند. دو لایهٔ دانش (میکرو و ماکرو) ایجاد شد.

آن‌ها ذی‌نفعان خُبره و روشی را که برای ارزیابی ورودی‌های ذی‌نفعان استفاده شد و موضوعات بازار را شناسایی کردند که نمایه‌هایی را برای بازتاب مشارکت ذی‌نفعان در هر

[Downloaded from jots.knu.edu.af on 2026-07-04]

موضوع ایجاد می‌کند. همچنین، با در نظر گرفتن عوامل اساسی کمک‌کنندهٔ ذی‌نفعان خاص، اغلب راه‌حل‌ها را در طیف وسیع‌تری از ابتکارات بررسی می‌کنند تا اینکه هر کدام را به‌طور مساوی مشاهده کنیم. رویکردهای تعریف‌شده در مقالهٔ [۱۵۰] تقریباً به‌طور کامل روی محتویات اشیاء اختصاص داده‌شده به هر موضوع متمرکز هستند. در موارد خاص، استفادهٔ تصادفی از کلمات، مستلزم توزیع مقادیر سری کوچک برای موضوعات مفهومی مشابه است. به‌عنوان مثال، موضوعات حمل‌ونقل و نظارت حتی اگر از نظر فنی به یکدیگر مرتبط بودند، مرتبط نبودند. تکنیک‌های فیلتر همکاری می‌توانند این مشکل را با ایجاد اجتماعات ذی‌نفعان مشابه کاهش دهند تا مشخص شود آیا ذی‌نفع از موضوعات مورد نظر اطلاع می‌دهد یا خیر.

در روش توصیهٔ مدل فراین نرم‌افزاری، به مدیران پروژه توصیه می‌شود از بهترین مدل چرخهٔ نرم‌افزاری برای پروژهٔ فعلی براساس اثبات علم دادهٔ موجود در مراحل اولیهٔ اجرا استفاده کنند. رویکرد مدیریت پروژه، چهارچوب جامعه را با توصیه برای ساختار پروتکل به چالش می‌کشد و تفاوت‌ها در الگوریتم‌های طبقه‌بندی و انتخاب جایگزین را تجزیه‌وتحلیل می‌کند که با مدل پیشنهادی شامل ارقام ایجاد نرم‌افزار تاریخی برای تخمین مدل فراین پروژهٔ نرم‌افزار مدرن تنها با برخی جزئیات همراه است [۱۵۱]. استفاده از این شکل از اطلاعات به‌طور کامل به مدیران پروژه در انتخاب سیستم نرم‌افزاری مناسب برای پروژهٔ فعلی در نقطه‌ای دقیق در مرحله ایجاد کمک می‌کند؛ همچنین، اثر متقابل بین مدل‌های فاز و اشکال مختلف پروژهٔ فاکتور را ارزیابی می‌کند تا به مدیران پروژه اجازه دهد مناسب‌ترین مدل فراین را انتخاب کنند و از بهترین رویکرد استفاده می‌کند. چهارچوب توصیهٔ خودکار متمرکز بر مدل فراین نرم‌افزار ML استفاده می‌شود تا مدیریت پروژه را قادر سازد تا با توجه به نتایج مهندسی نرم‌افزار تاریخی تصمیم بگیرد که کدام مدل فراین نرم‌افزار در مرحلهٔ توسعهٔ اولیه برای پروژه‌ای جدید مناسب‌تر است [۱۵۲].

معماری توصیه‌های شخصی‌شده، رابطی را ارائه می‌دهد که به تیم‌های تحقیقاتی داده کمک می‌کند به‌طور مؤثر در طرح‌های ML مشارکت کنند. خروجی، بازخورد سفارشی میلیون‌ها متقاضی را می‌دهد که در کمتر از یک ثانیه به پرسش‌ها پاسخ می‌دهد و دانش جدیدی به ارمغان می‌آورد؛ بنابراین، روی پیاده‌سازی منبع باز Antelope اجرا می‌شود و ایدهٔ تاریخچهٔ پرونده، پلتفرم مهندسی اطلاعات انعطاف‌پذیری را ادغام می‌کند. ممکن است از طیف وسیعی از ابزارهای ML همراه با فناوری استفاده شود که در اینجا توسعه یافته است و می‌تواند امکان ادغام عمیق‌تر را فراهم کند، حتی برخی که فقط با سیستم‌های مدیریت دادهٔ سنتی رابطه دارند [۱۵۳].

۳/۳/٤. توصیه‌هایی برای فراین نرم‌افزار مدیریت

Agile در تلاش است تا ابتدا با ایجاد مهم‌ترین ویژگی‌ها تأثیر سازگاری نازک را به حداقل برساند. در مقایسه، پروژه‌های معماری در محیط‌های پروژهٔ بزرگ‌تر نیز با مشکلات خروجی مواجه می‌شوند که مشخصات کلی سیستم را منصفانه می‌دانند که این می‌تواند در مورد جوامع چابکی که با محصولات غیرچابکی برخورد می‌کنند، که باعث تعارض بین سطوح چابک و

[Downloaded from jots.knu.edu.af on 2026-07-04]

غیرچابک می‌شود، اعمال شود [۱۵٤]. هنگام پیروی از رویکرد چابک، گرایش‌ها ممکن است اختلاف را برای ارائهٔ راه‌حل کوتاه‌مدت با بهبودهای کوتاه‌مدت و طرح حل تعارض ارزشی بلندمدت، بهتر شناسایی کنند. اکنون فرهنگ‌های مختلف در حال هم‌گرایی و تعامل هستند و دیگر نظام‌های منطقه‌ای با هم تلاقی می‌کنند که می‌تواند شامل روابط مالیاتی، فناوری، عملکردی، شرکتی یا ارتباطی باشد [۱۵۵].

تضادهای مبتنی‌بر ارزش سیستم‌های چابک نرم‌افزار چابک مستقل هستند. در ادبیات وجود دارد که سازمانی عامل مهمی در توسعهٔ چابکی است. ادبیات نشان می‌دهد اگر پروژهٔ چابکی محیط پروژهٔ خود را تحت‌تأثیر قرار ندهد یا آن را تغییر ندهد، از انجام کارهایی که باید به دست آید، حمایت بسیار کمی وجود دارد. هیچ راهنمایی‌ای در مورد نظارت و مدیریت اختلافات و مداخلات و مدیریت ارائه نشده است؛ این رویکرد شامل توسعهٔ محصول چابک، تضادهای مبتنی‌بر ارزش و ابتکارهای نهادی توسط [۱۱۳] است که مشکلات و راه‌حل‌ها را تشخیص می‌دهد. تجربهٔ ما از تیم توسعهٔ نرم‌افزار Agile است که در فضایی غیرپروژه‌ای کار می‌کند و ممکن است تأثیر و/یا تغییر قابل توجهی نداشته باشد. آرمان‌های افراد و نهادها بخشی از محیط هستند؛ بنابراین مفاهیم چابک باید ارزیابی شوند.

روش تخمین خودکار Agile به‌طور مؤثر روش تخمین کارت خودکار را برای داده‌های انسانی پیش‌بینی‌شدهٔ تاریخی با آخرین الگوریتم‌های ML اعمال می‌کند [۳۱]. رویکرد «برآورد خودکار» محبوب‌ترین شکل آماده‌سازی دستی پوکر را در محیط‌های چابک تقویت می‌کند [۱۵٤]. خودتخمینی از ویژگی‌های کارت داستان در محیط چابکی استفاده می‌کند که دقت تخمین را با کاهش تأثیر تخمین‌های اشتباه به حداکثر می‌رساند؛ نشان می‌دهد تخمین خودکار آمادگی پوکر را در قسمت آخر پروژه افزایش می‌دهد و ارزش نوشتن کارت‌های داستانی با طراحی مناسب را تعیین کنید.

کنترل پروژه ازطریق روش‌های هوش محاسباتی با مدیریت داده‌های عددی و زبانی، نویز خطای محاسباتی، درک انسانی و اصول تصمیم‌گیری مبهم مرتبط است و راه‌ها و ابزارهای فنی جدید برای مدیریت سرمایه‌گذاری‌ها و برنامه‌های کاربردی دسترسی آزاد در دهه‌های اخیر برای هوش رایانه‌ای را بررسی می‌کند؛ همچنین مروری بر الگوها و مکان‌های نوظهور برای توسعه، ارزیابی بخش‌های خاص با کاربرد موضوعی قوی وجود دارد [۲۵]. ورودی به نیاز پیش‌بینی‌شده برای ایجاد مدل‌های کنترل پروژهٔ مدرن و منابع فناوری اطلاعات اشاره دارد که شامل چهارچوب‌های مبتنی‌بر ML و مراقبت از عدم دقت اطلاعات، ابهام توسط معیارهای اصلی موفقیت است که به تمام زمینه‌های دانش مرتبط است. معرفی کتابخانه‌های ارزیابی یادگیری مدرن و چهارچوب‌های توسعهٔ منبع باز برای مدیریت پروژه، حوزه‌ای از مطالعه را باز می‌کند که به هم‌گرایی فنی منابع فناوری اطلاعات مرتبط است [۱۵٦ و ۱۵۷].

[Downloaded from jots.knu.edu.af on 2026-07-04]

برنامه‌ریزی انتشار استراتژیک (SRP) گام مهمی در رشد نرم‌افزارهای تکراری است. SRP شامل تحویل، مانند ترکیب، کنترل‌های نرم ازجمله زمان، منابع، قیمت یا پول، ویژگی‌ها یا الزامات انتشار است. SRP-Plugin نشان می‌دهد برنامه‌های کاربردی مورد استفاده با برنامهٔ مشترک به بهبود بهره‌وری فراین توسعه کمک می‌کنند [۱۲۱]. این افزونه اکوسیستم فضای بصری غنی‌ای با قابلیت‌های آماده‌سازی نسخهٔ پیشرفته دارد که ظرفیت آماده‌سازی برای پرتاب‌ها، افزایش دوام و تقویت همکاری بین سهام‌داران پروژه را افزایش می‌دهد. SRP-Plugin با رویکردی قوی، دقیق و سازمان‌یافته و توانایی Release Planner برای تولید طرح‌های انتشار پیچیده، Visual Lab را بهبود می‌بخشد؛ بااین‌حال آماده‌سازی زمان‌بندی‌های انتشار فقط اولین مرحله در پیچیده کردن ایده‌هایی است که راهنمایی گسترده‌تری برای تصمیم‌گیری‌های برنامه‌ریزی انتشار استراتژیک ارائه می‌دهد.

٤/٣/٤. توصیه‌هایی برای پیش‌بینی ریسک

اگر پروژه‌های مرتبط با نرم‌افزار می‌خواهند کارایی خود را مستقل از حوزهٔ کسب‌وکارشان بهبود بخشند، مدیریت ریسک ضروری است. انتظارات مصرف‌کننده هنوز در سیستم‌های اثبات‌شده، که تحت نظارت دقیق قرار دارند، مورد توجه قرار نمی‌گیرند [۱۵۸]. تعیین خطرات برای ابتکار ملاحظه‌ای اساسی در ارزیابی موفقیت پروژه یا رگرسیون است. تقریباً هر سازمانی از ابزارهای پیچیده برای طبقه‌بندی، به حداقل رساندن و حذف کلی آسیب استفاده می‌کند.

آیا ابتکار با ریسک بالا به‌موقع شناسایی می‌شود؟ بااین‌حال مدل‌های فعلی در درجهٔ اول بر این فرض متمرکز هستند که تمام هزینه‌های طبقه‌بندی خطا معادل هستند و تخمین احتمال در پروژهٔ نرم‌افزاری هستند. هزینهٔ پیش‌بینی پروژه‌ای شکست‌خورده به‌عنوان پروژه‌ای که احتمالاً به موفقیت می‌رسد با پیش‌بینی پروژه‌ای که احتمالاً به‌عنوان پروژه‌ای شکست‌خورده موفق می‌شود، متفاوت است. تا آنجا که ما درک می‌کنیم، درحالی‌که معمولاً در چندین زمینه استفاده می‌شود [۱۵۹]، رویکرد یادگیری حساس به هزینه هنوز در زمینهٔ مدیریت ریسک پروژهٔ نرم‌افزار برون‌سپاری اعمال نمی‌شود. در منطقهٔ مورد مطالعه، مدل پیش‌بینی ریسک پروژهٔ نرم‌افزاری، دو حفرهٔ تحقیقاتی عمده وجود دارد: اولاً، مدل‌های پیش‌بینی ریسک که منحصر به پروژهٔ نرم‌افزاری برون‌سپاری هستند که به‌ندرت بررسی می‌شوند؛ دوماً، اگرچه مطالعات پیش‌بینی ریسک پروژهٔ نرم‌افزاری جامع هستند، هیچ محققی روش‌های یادگیری حساس به هزینه را در پیش‌بینی ریسک پروژهٔ نرم‌افزاری اعمال نکرده است.

مدیریت نرم‌افزار مبتنی‌بر Agile برای بخش عمده‌ای از عملکرد پروژه در زمان‌های اخیر سازنده بوده است. تهدیدهای مرتبط با زمان، که بر انتشار محصولات قابل‌تحویل تأثیر می‌گذارند، تهدیدهای مبتنی‌بر زمان‌بندی را نشان می‌دهند؛ زیرا منابع مالی، پیش‌بینی‌های غیرقابل اعتماد آن زمان و اقدامات مثبت مدیر پروژه به‌درستی توزیع نشده‌اند. تهدیدات بودجه منعکس‌کنندهٔ مخاطرات مالی است که ممکن است از انبوه وجوه رخ دهد. چنین عواملی ممکن

[Downloaded from jots.knu.edu.af on 2026-07-04]

است به گسترش دسترسی ناخواستهٔ پروژه، استفادهٔ کم از خروجی‌های موجود و مدیریت ضعیف نسبت داده شوند [۱۶۰].

اَشکال خطر عملیاتی با رویه‌های منظم پروژه همراه است که رویه‌های اشتباه، برنامه‌ریزی ناکافی و قدرت تیمی از دلایل این تهدیدات هستند. اگر دربارهٔ ارزش روش‌های چابک و کاربرد چهارچوب‌های مدرن در کنترل ریسک توسط منابع کافی بحث کنید، در آینده تیم را قادر به تعیین اثر تهدیدها با استفاده از ارزیابی پارامترهای ریسک می‌سازد؛ همچنین درصورت استفاده از چنین معیارهایی، احتمال قوی خروجی صدا وجود دارد. مدل‌های ریسک پیش‌بینی‌کننده تخمین ۵۰ درصدی از نوع تأخیرهای مخاطره‌آمیز نرم‌افزار، تعداد درخواست‌های اسکن کلمهٔ کلیدی را کاهش می‌دهند. دقت مدل‌های بیزی دریافتی با استفاده از چندین مقیاس طبقه‌بندی اندازه‌گیری و مقایسه می‌شود. معماری شبکهٔ بهینه‌شده برای درخت عملکرد، آزمایشی موفقیت‌آمیز را برای همهٔ مجموعهٔ داده‌ها نشان می‌دهد. رابطهٔ بین متغیرهای به دست آمده که توسط مهندسان ضرورت نیز برای تعیین سطح خطر در یک موقعیت مشخص می‌شود. شبکه‌های بیزی، روش‌های ضروری در مهندسی ضرورت برای اتوماسیون مدیریت ریسک هستند. هدف مدیریت ریسک در توسعهٔ نرم‌افزار، شناسایی، اندازه‌گیری، آماده‌سازی و واکنش به ریسک‌های بالقوه برای جلوگیری از تأثیر آن‌ها بر پروژهٔ نرم‌افزاری است.

ارزیابی ریسک پروژهٔ نرم‌افزاری مبتنی‌بر شبکهٔ Credal روشی مدرن است که برای ارزیابی آسیب‌پذیری برنامهٔ پروژهٔ متمرکز بر شبکهٔ Credal معرفی شده است که با استفاده از مجموعهٔ اعتباری برای اندازه‌گیری تأثیر عوامل خطر [۱۱۶]، با نظرات متناقض کارشناسان و تفاوت‌های آن‌ها مقابله خواهد کرد.

استنتاج شبکهٔ اعتباری، پیش‌بینی ریسک و تشخیص ریسک را انجام داد. نتایج موردی نشان می‌دهد اصل روش صحیح است و ارزیابی پروژهٔ نرم‌افزاری به‌خوبی پیش‌بینی شده است. هدف مدل‌سازی تهدیدها برای پیش‌بینی ریسک‌ها، پیامدهای خطرات و تعریف عوامل خطر اصلی است که استراتژی ریسک و مدیریت ریسک را ارتقا می‌دهد؛ بااین‌حال دورهٔ توسعهٔ طولانی، پیچیدگی بالای محصول و بی‌ثباتی فوق العاده روش، پیش‌بینی و ارزیابی خطر پروژهٔ نرم‌افزار را غیرممکن می‌کند. فلسفهٔ ارزیابی ریسک پروژهٔ نرم‌افزاری کنونی، اغلب ریسک‌ها را با در نظر گرفتن ویژگی‌های سیستم‌های نرم‌افزاری و روش‌های توسعهٔ اطلاعات از مدیریت کلی پروژه تحلیل می‌کند.

۵/۳/٤. توصیه‌هایی برای مدل‌های پیش‌بینی خطای نرم‌افزار

پیش‌بینی تلاش و مدت‌زمان توسعهٔ نرم‌افزار، وظیفه‌ای حیاتی برای SPM فعال است. دقت و قابلیت اطمینان روش‌های پیش‌بینی نیز مهم هستند. چندین الگوریتم ML برای پیش‌بینی مدت‌زمان نرم‌افزار استفاده شد.

[Downloaded from jots.knu.edu.af on 2026-07-04]

رویکردهای مبتنی‌بر پیش‌بینی به یک ویژگی پیش‌بینی نیاز دارند که تعهد و دورهٔ بالقوهٔ پروژه را با توجه به داده‌های فعلی/گذشته پروژه پیش‌بینی می‌کند؛ با وجود این، الگوریتم‌های متعدد ML اغلب با وجود تعداد زیادی از الگوریتم‌های ML ارزیابی نمی‌شوند. برای مدل ماشین ساخت، براساس چندین جزئیات پروژه، الگوریتم‌های مختلف ML استفاده می‌شوند [۱۶۱]. مدل ML، شبکهٔ رگرسیون عصبی جهانی (GRNN) نام گرفت تا کارایی متخصصان فناوری را پیش‌بینی کند. GRNN می‌تواند برای پیش‌بینی بهره‌وری پزشکان برای خطوط جدید و اصلاح‌شدهٔ کد، کدها و برنامه‌نویسان استفاده شود. تجربیات به‌عنوان متغیرهای مستقل دقت پیش‌بینی GRNN بهتر از رگرسیون آماری زمانی استفاده می‌شود که از دو مدل برای پیش‌بینی بهره‌وری متخصصان نرم‌افزار اختصاص داده‌شده به‌صورت جداگانه استفاده می‌شود.

پیش‌بینی با استفاده از تکنیک‌های داده‌کاوی کارآمد برای این بخش از تجزیه‌وتحلیل، موفقیت/شکست پروژه‌ها را با محاسبهٔ فاز به فاز، به‌جای سبک ارزیابی منظم از جنبهٔ کل پروژه، اندازه‌گیری می‌کند. روش داده‌کاوی برای خوشه‌بندی و گروه‌بندی روش‌ها با جمع‌آوری داده‌ها از ابتکارات مختلف ازطریق چندین بخش محاسباتی استفاده می‌شود. آن‌ها همچنین پیشنهاد کردند که تکمیل پروژه درحال افزایش است و پروژهٔ شکست‌خورده، موفقیت‌آمیز خواهد بود. برای پیش‌بینی شدت خطای برنامه، ناگوانی و باانسالی مدل رابط کاربری گرافیکی مدرنی معرفی کردند [۱۶۲]. رویکرد خوشه‌بندی برای ساخت خوشهٔ شدت مخزن اشکالات نرم‌افزاری با طول رفع مشکل استفاده شد. مدل پیشنهادی با استفاده از کد منبع باز، که اغلب توسط مخزن مسائل نرم‌افزار باز MySql ارائه می‌شود، اعمال می‌شود [۱۶۳].

مدل طبقه‌بندی برای پیش‌بینی لغزش هزینه با استفاده از تکنیک‌های داده‌کاوی از بودجه و زمان‌بندی برای برنامه‌ریزی اولیهٔ پروژهٔ ICT استفاده می‌کند؛ سپس لغزش هزینه را در دستهٔ پروژه پیش‌بینی می‌کند. سه دسته سقوط وجود دارد که طبیعی تلقی می‌شوند؛ لغزش متوسط و سقوط زیاد نیاز به اقدام دارند [۱۱۵،۱۶٤]. هدف توضیح این است که چگونه مدل طبقه‌بندی‌ای با استفاده از تکنیک‌های داده‌کاوی برای پیش‌بینی تلفات هزینه ساخته می‌شود. مدل پیشنهادی از ورودی (به‌عنوان مثال، بودجهٔ اولیه و زمان‌بندی) تعداد محدودی از پارامترهای پروژه استفاده می‌کند و پروژه را به یکی از سه دستهٔ (عادی، متوسط و بزرگ) تقسیم می‌کند.

تکنیک کشف الگو، اجرای آزمایشی در زیست‌شناسی محاسباتی با استراتژی کشف الگو است که به‌طور مؤثر اجرا شده است. این فناوری، عادت‌های رابطه‌ای، که از سوابق به ارث رسیده‌اند، آشکار می‌کند و شاغلین کسب‌وکار را قادر می‌سازد تا دانش معناداری به دست آورند و اعتماد را در تصمیم‌گیری بهبود بخشند. روندهای آماری مرتبط با نتایج درجهٔ خوب برای مجموعهٔ داده‌های آزمایش‌شده تولید شد [۱۶۵]؛ همچنین تأثیرات روی نتایج تکنیک‌های بودجه‌ای متعدد را نشان می‌دهد. اولین تحقیق از روش منحصر به فردکاوی الگوها برای

[Downloaded from jots.knu.edu.af on 2026-07-04]

شناسایی نرم‌افزار معیوب در مهندسی نرم‌افزار استفاده می‌کند. یافته‌ها توانایی چنین استراتژی‌ای را برای ارائهٔ عملکرد رتبه‌بندی مثبت و دانش مفید برای تصمیم‌گیرندگان نشان داده‌اند.

ایجاد مدل‌های پیش‌بینی پروژهٔ نرم‌افزاری برای افزایش تلاش جهت شناسایی عیوب و توسعهٔ مدل پیش‌بینی نقص برای شرکت بزرگ نرم‌افزاری صنعتی‌ای ارزشمند است. سیستم و اندازه‌گیری روش برای ایجاد مدل نشان می‌دهد حتی اگر چهار درصد از برنامه دارای نقص است، هدف این است که عامل ناکارآمدی را در فراین توسعهٔ مستمر پروژهٔ برنامه‌ای عظیم وارد کرده، ابتکار ارزیابی را کاهش دهد و مطالعهٔ تجربهٔ آخرین استراتژی را ارائه دهد [١٦٦]. این مدل، پیش‌بینی خطای بسیار قابل اعتمادی برای برنامهٔ معیوب چهار درصدی ارائه می‌کند. این مدل از RF استفاده می‌کند که از NB، رگرسیون لجستیک و DT قابل اعتمادتر است.

مدل‌های پیش‌بینی شکست با استفاده از الگوریتم‌های طبقه‌بندی چهارچوب پیاده‌سازی‌شده، با استفاده از الگوریتم‌های طبقه‌بندی ML به‌طور خودکار، مدل‌های پیش‌بینی شکست را می‌سازند و عملکرد تکنیک‌های مختلف برای پروژه‌های فایرفاکس و Netbeans را مقایسه می‌کنند. این محاسبه براساس مدل هزینه–فایده برای تعیین اهمیت تحقیقات اولیهٔ اضافی است. اهمیت تحقیقات بیشتر در مراحل اولیه در این مدل، براساس عملکرد احتمالی آن در جلوگیری از خرابی در هزینهٔ نسبی خطاهایی است که با هزینه‌های آن مرتبط هستند [١٦٧]. پیش‌بینی‌های منطقی دو پارامتر پیش‌بینی عددی کاربرد بهتری را برای برخی اَشکال و برنامه‌های خطا نسبت به مجموعه‌ای از خطوط مبنا می‌دهد. این نشان می‌دهد که پیش‌بینی خودکار شکست ممکن است راه‌حلی سودمند برای فعالیت‌های توسعهٔ نیازمندی‌های راهنمایی در محیط‌های آنلاین در طول ایجاد نیازهای آنلاین باشد.

پیش‌بینی نقص بین پروژه (CPDP) حوزهٔ تحقیقاتی است که داده‌های سایر برنامه‌ها در آن می‌توانند توسط پروژه نرم‌افزاری‌ای با داده‌های محلی ناکافی برای ساخت پیش‌بینی‌کننده‌های خطا استفاده شوند. جزئیات پروژه باید به‌دقت بررسی شوند تا پیش از اجرای محلی به CPDP کمک کند. چندین فیلتر افزایش کارایی CPDP خاص توسط محققان، توسعه یافته و معرفی شده‌اند [١٦٨]؛ بااین‌حال، تکنیک فیلتر داده به‌طور کلی و به‌ویژه در CPDP هنوز نامشخص است. این نشان می‌دهد تکنیک فیلتر داده به‌طور چشمگیری کارایی پیش‌بینی خطای بین پروژه‌ای را افزایش می‌دهد و فیلتر انتخابی سلسله‌مراتبی حتی حیاتی‌تر است. علاوه‌بر این، پیش‌بینی‌کنندهٔ نقص وابسته به داده‌های بین پروژه‌ای ممکن است با استفاده از تکنیک فیلتر دادهٔ صحیح، پیش‌بینی‌کنندهٔ آموزش‌دیده با استفاده از داده‌های پروژهٔ داخلی را دور بزنند. در واقع، CPDP مورد نیاز است؛ زیرا از داده‌های منبع/پروژه برچسب‌گذاری‌شده برای ساخت مدل و پیش‌بینی خطا برای پروژه‌ای هدف استفاده می‌کند [٨٤].

سؤال پنجم تحقیق: کدام الگوریتم یادگیری ماشینی تمایل به ناچیز پنداشتن و کدام الگوریتم دست کم گرفتن دارد؟

[Downloaded from jots.knu.edu.af on 2026-07-04]

پیش‌بینی خطای نرم‌افزار دقت مبتنی‌بر: پیش‌بینی اولیهٔ خطاها در نرم‌افزار با استفاده از تکنیک پیش‌بینی خاصی ممکن است هزینه و تلاش را به حداقل برساند. تکنیک‌های مختلف ML برای پیش‌بینی خطا استفاده شده‌اند و ثابت شده است که مفید هستند. جدول ۲ دقت پیش‌بینی، که توسط مطالعات اولیه گزارش شده‌اند، ادغام می‌کند. ذکر این نکته دلگرم‌کننده است که از ۱۱۱ مطالعهٔ اولیهٔ تحت‌پوشش دوربین‌های SLR، نزدیک به ۲۲ مورد دقت مدل را گزارش می‌دهند. SVR و KNN پرکاربردترین معیارهایی هستند که دقت آن‌ها را تقریباً ۹۵٪ مطالعات گزارش کرده‌اند. زمانی که داده‌ها، براساس نوع سازمان یا نوع صنعت، به‌جای محدود شدن به مجموعهٔ دادهٔ شرکت، بین گروه‌های همگن مختلف تفکیک می‌شوند، شواهد نسبتاً بیشتری دربارهٔ دقت بهتر وجود دارد. طبقه‌بندی دقت، برآورد تلاش را بهبود بخشیده است. ما نتایج تخمین‌زده‌شده را در نرم‌افزارهای مختلف بین الگوریتم‌ها مقایسه کرده‌ایم. این الگوریتم‌ها می‌توانند در مراحل اولیهٔ چرخهٔ عمر نرم‌افزار استفاده شوند و می‌توانند به SPM کمک کنند تا پیش از شروع پروژه، تخمین تلاش را به‌طور مؤثری انجام دهد. اندازهٔ نرم‌افزار، بهره‌وری، پیچیدگی و پایداری نیاز، عوامل ورودی این مدل‌ها هستند.

جدول ۲: دقت پیش‌بینی در مطالعات انجام‌شده

مرجع	نوع یادگیری ماشین	دیتاست	مدل	دستیابی به پیش‌بینی	مزایا	محدودیت‌ها
[73]	kNN	IBM commercial projects called RQM and RTC	مدل ترکیبی از سه مجموعه ویژگی مستقل شامل ویژگی های مبتنی‌بر ابرادهٔ اولیه، عنوان و شرح وظایف نرم‌افزار استفاده می کند.	دقت ۸۸٪	برآورد تلاش خودکار برای تعداد بیشتری از کارها	در مجموعهٔ داده‌های این مطالعه، عکس‌های فوری تاریخ نداشتند تا مطمئن شوند که ارزش نهایی ویژگی های گنجانده‌شده برای همهٔ وظایف با مقدار آن‌ها قبل از تخصیص به توسعه‌دهنده برابر است.

[Downloaded from jots.knu.edu.af on 2026-07-04]

[Downloaded from jots.knu.edu.af on 2026-07-04]

[47]	[72]	[48]
K-NN	Naïve Bayes	Logistic linear regression
PROMISE Repository	Data sets University Student Projects developed in 2005)	KitchenMax CocNasaCoc81 ISBSG2000 ISBSG2001 ISBSG
Software effort estimation	Software Effort Estimation	DYCOM
دقت ۹۲٪	دقت ۸۷٪	دقت ٦٦٪
بررسی اینکه تنظیمات پارامتر تا چه حد بر عملکرد ML در SEE تأثیر می‌گذارد و چه ماشین‌های یادگیری‌ای به پارامترهای خود حساس‌تر هستند.	براساس تکنیک‌های ML برای داده‌های غیرکمّی و در دو فاز انجام می‌شود	بهترین استفاده از داده‌های CC، به‌طوری که می‌تواند مقدار داده‌های WC را کاهش دهد و در عین حال عملکرد را در مقایسه با مدل‌های WC SEE حفظ یا بهبود بخشد.
بررسی سایر ماشین‌های یادگیری و مجموعهٔ داده‌ها؛ انواع دیگر اندازهٔ اثر، به‌ویژه موارد غیرپارامتری و سایر اندازه‌های پنجره برای ارزیابی روند یادگیری آنلاین	کارایی سایر تکنیک‌های ML مانند SVM، یادگیری درخت تصمیم و ... را می‌توان برای تخمین تلاش استفاده کرد.	بررسی حساسیت Dycom به مقادیر پارامترها، یادگیرندگان پایه، ویژگی‌های ورودی و تکنیک‌های تقسیم پروژه‌های CC به بخش‌های مختلف

[Downloaded from jots.knu.edu.af on 2026-07-04]

[37]	[81]	[60]
Fuzzy logic	ANN	SVR
ISBSG, COCOMO and DESHARNAIS datasets	NASA 93	NASA93 dataset
HYBRID Models	Experiments Models	Software Effort Estimation
دقت ٪۹۷	دقت ٪۹۵	دقت ٪۹۵
به مسئلهٔ برآورد هزینهٔ نرم‌افزار می‌پردازد و رویکرد جایگزینی را پیشنهاد می‌کند که ساختارهای درخت تصمیم‌گیری قوی را با منطق فازی ترکیب می‌کند.	بررسی تأثیر طبقه‌بندی تخمین میزان تلاش مورد نیاز در پروژه‌های توسعهٔ نرم‌افزار	مقایسه‌ای بین تکنیک‌های محاسبات نرم و رگرسیون آماری از نظر مشکل رگرسیون تخمین توسعهٔ نرم‌افزار.
مجموعهٔ وسیع‌تری از انواع ویژگی‌ها مانند ویژگی‌های طبقه‌بندی و بیشتر روی مواردی که در مراحل اولیه توسعه پروژه در دسترس هستند تمرکز کنید تا به موضوع پیشنهاد مدل‌های هزینه بهتر و کاربردی‌تر بپردازید.	پیاده‌سازی مدلی برای تخمین مقدار نهایی تلاش مورد نیاز در پروژه‌های جدید، برای برآورد تلاش جزئی در مراحل مختلف فرایند توسعهٔ پروژه.	نیاز به تحقیقات بیشتر در آینده برای ارزیابی کارایی تکنیک‌های محاسبات نرم در مقایسه با روش‌های رایج رگرسیون آماری، به‌ویژه در زمینهٔ تخمین تلاش نرم‌افزاری.

[Downloaded from jots.knu.edu.af on 2026-07-04]

[107]	[109]
KNN	SVR
KEMERER, MAXWELL, MIYAZAKI 1, NASA 60, NASA 63, NASA93	International Software Benchmarking Standards Group (ISBSG) repository
Software Cost Estimation (SCE) models	Data homogeneity
دقت ۹۱٪	دقت ۹۸٪
روش‌های مبتنی‌بر مدل از فرمولی واحد و مقادیر ثابتی استفاده می‌کنند و این روش‌ها پاسخگوی پیشرفت‌های روزافزون در زمینهٔ مهندسی نرم‌افزار نیستند.	بررسی همگنی داده‌های هزینه بر حسب حوزه‌های کاربردی و تمرکز بر دامنه تعبیه شده.
نسبت به الگوریتم‌های مقایسه‌ای عملکرد خوبی ندارد و دلیل آن می‌تواند عدم وجود داده‌های منسجم باشد.	فرایند جمع‌آوری داده‌ها در حوزهٔ سیستم‌های جاسازی‌شده ممکن است بر جستجوی ویژگی‌های خاص دامنه متمرکز شود؛ به‌طوری که محتوای اطلاعاتی ویژگی‌ها غنی‌تر شده و در نتیجه عملکرد پیش‌بینی الگوریتم بهبود می‌یابد.

[Downloaded from jots.knu.edu.af on 2026-07-04]

[89]	[46]
SVR	Decision tree
ISBSG dataset	Kemerer Bank Test equipment DSI Moser, Desharnais Finnish, ISBSG CCCS, Company X
Software project estimation	Software effort prediction
دقت ٪۷۲	دقت ٪۹۲
با استفاده از یافته‌های تحقیقاتی و بهترین شیوه‌های صنعت، شکاف بین نتایج تحقیقات به‌روز و پیاده‌سازی‌ها در سازمان‌ها را با پیشنهاد رویکردهای استقرار و نگهداری مؤثر و عملی ML کاهش دهید.	بهبود دقت پیش‌بینی تلاش نرم‌افزار با تولید مجموعه با استفاده از دو روش انتساب به عنوان عناصر.
تمرکز بر تأیید رویکرد پیشنهادی ازطریق اثبات مفهوم با سازمان‌های مختلف برای اعتبارسنجی دقت مدل و تنظیم چهارچوب استقرار و نگهداری	از نظر پارامترهای آموزشی و قوانین ترکیبی که می‌توان از آن‌ها استفاده کرد. دوم، مطالعات تجربی کاربرد MIAMI در مجموعهٔ داده‌های سایر حوزه‌های داده‌کاوی باید برای ارزیابی عملکرد آن در زمینه‌ای کلی‌تر انجام شود.

[Downloaded from jots.knu.edu.af on 2026-07-04]

[94]	[92]
ANN	Neural networks
ISBSG datasets	Historical data
Software development effort estimation	I-Competere
دقت ۹۷٪	دقت ۹۳٪
در ارتباط با تبدیل ویژگی، انتخاب ویژگی و تکنیک‌های تنظیم پارامتر برای تخمین دقیق تلاش توسعه بررسی شد و مدلی به‌عنوان بخشی از سیستم خبره‌ای پیشنهاد شد.	ارائۀ ابزاری برای پیش‌بینی شکاف‌های شایستگی در پرسنل مدیریت کلیدی با پیش‌بینی برنامه‌ریزی و زمان‌بندی سطوح شایستگی
مدل پیشنهادی در مجموعۀ داده‌های جدیدی که برای آزمایش‌ها و تجزیه‌وتحلیل ما در دسترس قرار می‌گیرند، استفاده خواهد شد.	ارائۀ ابزاری برای پیش‌بینی شکاف‌های شایستگی در پرسنل مدیریت کلیدی با پیش‌بینی سطوح شایستگی برنامه‌ریزی و زمان‌بندی با تمرکز بر گنجاندن انواع دیگر پروژه‌ها به‌منظور اثبات اینکه چهارچوب پیشنهادی می‌تواند هنگام پیش‌بینی شکاف‌های شایستگی در پروژه‌های مختلف تطبیق داده شود.

[Downloaded from jots.knu.edu.af on 2026-07-04]

[166]	[101]	[55]
Logistic linear regression	Random Forest	Random forest
Cross-Project Software Fault Prediction Using Data-Leveraging Technique to Improve Software	Real data	13 data sets
Source + target	Defect Prediction	Misclassification cost-sensitive
دقت ۹۵٪	دقت ۹۰٪	دقت ۹۵٪
ساخت مدل پیش‌بینی با استفاده از یادگیری انتقال مبتنی‌بر آنی ازطریق روش اهرم داده	ساخت مدل پیش‌بینی نقص برای پروژه بزرگ نرم‌افزار صنعتی	تجزیه‌وتحلیل مزایای تکنیک‌هایی که هزینه‌های طبقه‌بندی اشتباه را در توسعهٔ مدل‌های پیش‌بینی خطای نرم‌افزار گنجانده‌اند.
مجموعهٔ داده‌های بیشتری را از همان دامنه و با استفاده از سایر الگوریتم‌های ماشینی با مقایسهٔ نتایج آن‌ها گنجاند.	پیاده‌سازی مدل به‌عنوان الگوریتم آنلاینی که با هر نسخه یاد می‌گیرد.	نشان دهید که در پروژه‌هایی که هزینهٔ طبقه‌بندی اشتباه دقیق ناشناخته است، سناریوی محتمل در عمل، مدل‌های حساس به هزینه را با نسبت‌های

[Downloaded from jots.knu.edu.af on 2026-07-04]

[42]	[83]	[108]
KNN	ANN	Decision tree
Several dataset	Experiments on 45 open source project dataset	Company effort data set
EBMFO	Fault prediction model	Evolutionary-based Decision Trees
دقت ٨٩٪	دقت ٩٨٪	دقت ٦٤٪
بهینه‌سازی شعلهٔ پروانهٔ دودویی (EBMFO) با نمونه‌گیری مصنوعی تطبیقی (ADASYN) برای پیش‌بینی خطاهای نرم‌افزار	برای اعتبارسنجی معیارهای کد منبع و انتخاب مجموعهٔ مناسب از معیارها با هدف بهبود عملکرد مدل پیش‌بینی خطا	به‌کارگیری الگوریتم تکاملی برای تولید درخت تصمیم متناسب با مجموعهٔ داده‌های تلاش نرم‌افزاری ارائه‌شده توسط شرکت بزرگ فناوری اطلاعات در سراسر جهان
مطالعهٔ اهمیت ویژگی‌ها برای افزایش عملکرد طبقه‌بندی‌کننده‌ها و دقت مدل SFP	کاهش ویژگی‌های ویژگی با استفاده از چهارچوب پیشنهادی	تعیین اثربخشی آن در برآورد سایر معیارهای مهم نرم‌افزار در مجموعهٔ داده‌های توسعهٔ نرم‌افزار خصوصی و عمومی

[86]	SVM	Quanxi Mi data set	Defect management (DM)	دقت ۹۷٪	تمرکز روی جنبهٔ رویهٔ فرایندهای نرم‌افزار و فرمول‌بندی مسئله به‌عنوان کار طبقه‌بندی دنباله‌ای که با استفاده از ML حل می‌شود.	بررسی جنبه‌های اضافی فرایندهای نرم‌افزاری و سایر تکنیک‌های ML برای توسعهٔ راه‌حل‌های پیشرفته‌تر
[77]	Random Forest	NASA namely CM1, PC1 and JM1	Software Effort Estimation	دقت ۹۹٪	بررسی انتخاب مناسب تکنیک‌های داده‌کاوی به‌منظور برآورد دقیق میزان موفقیت و شکست پروژه ها براساس نقص به‌عنوان یکی از عوامل تعدیل‌کننده.	فرایند برآورد پروژه از این پس باعث بهبود کیفیت، بهره‌وری و پایداری شرکت در فضای صنعتی می شود.

۵. نتیجه‌گیری

تجزیه‌وتحلیل ادبی نتیجه گرفت که مطالعهٔ گسترده‌ای در مدیریت پروژهٔ نرم‌افزاری روی روش‌های ML انجام شده است. گسترش مشاغل در گذر زمان پیوسته بوده است. الگوریتم‌های ANN، منطق فازی، ژنتیک و رگرسیون روش‌های حیاتی ML برای تخمین تلاش خودکار هستند. محاسبهٔ دقیق تلاش از روش‌های پیشرو در توسعهٔ نرم‌افزار است. این نرم‌افزار به‌طور خاص تحت‌تأثیر زمان و سختی قرار گرفت. مضامین اساسی ممکن است از کارهای مختلف ML در مدیریت پروژهٔ نرم‌افزاری استخراج شوند. سرمایه‌گذاری‌ها تقریباً به چهار گروه طبقه‌بندی می‌شوند: گروه اول شامل بررسی‌ها و بررسی‌های مرتبط با مدیریت پروژهٔ نرم‌افزاری است؛ گروه دوم مقالاتی را پوشش می‌دهد که بر مطالعات موردی روش‌های مدیریت پروژهٔ

[Downloaded from jots.knu.edu.af on 2026-07-04]

نرم‌افزار تمرکز دارند؛ گروه سوم شامل انتشارات تجربی است که در مدیریت ML از یک نوع ساختار یا مدل معماری استفاده شده است و گروه چهارم، مطالعهٔ مشارکت پژوهشی، تجزیه‌وتحلیل پروژه‌ای، فرم سازه یا مدل معماری است. بررسی عمیق این مقالات به مدیریت پروژهٔ نرم‌افزار کمک می‌کند تا رویکردهای ML را برای تعریف و توضیح تهدیدها، مزایا و توصیه‌ها بررسی کند؛ بااین‌حال، به‌دلیل حجم زیاد الگوریتم‌های ML، الگوریتم‌های مختلف مطالعهٔ ماشین تحلیل‌نشده باقی می‌مانند. دلایل استفاده از SPM خودکار، مشکلات ارزیابی آماده‌سازی پروژه و فناوری‌های مهندسی ML براساس یافته‌های ادبیات بررسی می‌شوند. اگرچه ادبیات مربوط به SPM عملکرد پروژه‌ها و ضرر را توضیح می‌دهد، اما سنت طولانی اختلاف‌نظر دربارهٔ محاسبهٔ پیشرفت پروژه وجود دارد. دربارهٔ اینکه چه‌چیزی توسعهٔ پروژه‌ای را منعکس می‌کند و چگونه برآورد می‌شود، اختلاف‌نظر وجود دارد. این دستورالعمل‌ها مشکلات پیش‌روی پروژه‌ای نرم‌افزاری در روش‌های ML را برطرف می‌کند و فرصت‌های کاری در این بخش ایجاد می‌کند. تحقیقات هنوز باید تخمین تلاش را براساس رویکردهای ML، که بر ارزیابی ریسک تمرکز دارد، بررسی کند. عامل دیگر استفاده از روش‌های فیلتر استاندارد برای به حداقل رساندن مشکل با ایجاد مناطق با ذی‌نفعان مشابه و پیش‌بینی این است که آیا ذی‌نفع از موضوع آگاه است یا خیر. این بررسی ادبیات، پاسخ‌های اولیه‌ای به سؤالات اساسی دربارهٔ برآورد مدیریت پروژهٔ نرم‌افزاری، که براساس ML است، ارائه می‌دهد.

[Downloaded from jots.knu.edu.af on 2026-07-04]

منابع

1. Oun, T.A.; Blackburn, T.D.; Olson, B.A.; Blessner, P. An enterprise-wide knowledge management approach to project management. Eng. Manag. J. 2016, 28, 179–192.
2. Maimone, C. Good Enough Project Management Practices for Researcher Support Projects. In Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning), Chicago, IL, USA, 28 July–1 August 2019; pp. 1–8.
3. Saleem, N. Empirical analysis of critical success factors for project management in global software development. In Proceedings of the 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE), Montreal, QC, Canada, 25–26 May 2019; pp. 68–71.
4. Gemünden, H.G. Success factors of global new product development programs, the definition of project success, knowledge sharing, and special issues of project management journal. Proj. Manag. J. 2015, 46, 2–11.
5. Hughes, S.W.; Tippett, D.D.; Thomas, W.K. Measuring project success in the construction industry. Eng. Manag. J. 2004, 16, 31–37.
6. Project Management Institute. Guide to the Project Management Body of Knowledge (Pmbok Guide); Project Management Institute: Newtown Square, PA, USA, 2013.
7. Kirsch, L.J. Software project management: An integrated perspective for an emerging paradigm. In Framing the Domains of IT Management: Projecting the Future... Through the Past; Pinnaflex Educational Resources inc: Ann Arbor, MI, USA, 2000; pp. 285–304.
8. Aladwani, A.M. IT project uncertainty, planning and success. Inf. Technol. People 2002, 210–226.
9. Cates, G.R.; Mollaghasemi, M. The project assessment by simulation technique. Eng. Manag. J. 2007, 19, 3–10.
10. Parsons, V.S. Project performance: How to assess the early stages. Eng. Manag. J. 2006, 18, 11–15.
11. Rosenfeld, Y. Root-cause analysis of construction-cost overruns. J. Constr. Eng. Manag. 2014, 140, 04013039.
12. Wang, J.; Li, J.; Wang, Q.; Zhang, H.; Wang, H. A simulation approach for impact analysis of requirement volatility considering dependency change. In Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality, Essen, Germany, 19–22 March 2012; pp. 59–76.
13. Ferreira, S.; Collofello, J.; Shunk, D.; Mackulak, G. Understanding the effects of requirements volatility in software engineering by using analytical modeling and software process simulation. J. Syst. Softw. 2009, 82, 1568–1577.
14. Tiwana, A.; Keil, M. The one-minute risk assessment tool. Commun. ACM 2004, 47, 73–77.
15. Sommerville, I. Software Engineering, 9th ed.; Pearson: London, UK, 2011; ISBN 0137035152.

[Downloaded from jots.knu.edu.af on 2026-07-04]

16. Ali, N.; Hwang, S.; Hong, J.E. Your Opinions Let us Know: Mining Social Network Sites to Evolve Software Product Lines. Ksii Trans. Internet Inf. Syst. 2019, 13.
17. Malhotra, R.; Chug, A. Software maintainability: Systematic literature review and current trends. Int. J. Softw. Eng. Knowl. Eng. 2016, 26, 1221–1253.
18. Sharma, P.; Singh, J. Systematic literature review on software effort estimation using machine learning approaches. In Proceedings of the 2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS), Jammu, India, 11–12 December 2017; pp. 43–47.
19. Alsalemi, A.M.; Yeoh, E.-T. A Systematic Literature Review of Requirements Volatility Prediction. In Proceedings of the 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), Mysore, India, 8–9 September 2017; pp. 55–64.
20. Alsolai, H.; Roper, M. A systematic literature review of machine learning techniques for software maintainability prediction. Inf. Softw. Technol. 2020, 119, 106214.
21. Idri, A.; Abnane, I.; Abran, A. Systematic mapping study of missing values techniques in software engineering data. In Proceedings of the 2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Takamatsu, Japan, 1–3 June 2015; pp. 1–8.
22. Pillai, S.P.; Madhukumar, S.; Radharamanan, T. Consolidating evidence based studies in software cost/effort estimation—A tertiary study. In Proceedings of the TENCON 2017 IEEE Region 10 Conference, Penang, Malaysia, 5–8 November 2017; pp. 833–838.
23. Sangwan, O.P. Software effort estimation using machine learning techniques. In Proceedings of the 2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence, Noida, India, 12–13 January 2017; pp. 92–98.
24. Stewart, C.A.; Hancock, D.Y.; Wernert, J.; Furlani, T.; Lifka, D.; Sill, A.; Berente, N.; McMullen, D.F.; Cheatham, T.; Apon, A.; et al. Assessment of financial returns on investments in cyberinfrastructure facilities: A survey of current methods. In Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning), Chicago, IL, USA, 28 July–1 August 2019; pp. 1–8.
25. García, J.A.L.; Peña, A.B.; Pérez, P.Y.P.; Pérez, R.B. Project control and computational intelligence: Trends and challenges. Int. J. Comput. Intell. Syst. 2017, 10, 320–335.
26. Raharjo, T.; Purwandari, B. Agile Project Management Challenges and Mapping Solutions: A Systematic Literature Review. In Proceedings of the 3rd International Conference on Software Engineering and Information Management, Sydney, NSW, Australia, 12–15 January 2020; pp. 123–129.
27. Cleland-Huang, J.; Czauderna, A.; Gibiec, M.; Emenecker, J. A machine learning approach for tracing regulatory codes to product specific requirements. In Proceedings of the 32nd ACM/IEEE International

[Downloaded from jots.knu.edu.af on 2026-07-04]

Conference on Software Engineering, Cape Town, South Africa, 2–8 May 2010; Volume 1, pp. 155–164.
28. Zhang, D.; Dang, Y.; Lou, J.-G.; Han, S.; Zhang, H.; Xie, T. Software analytics as a learning case in practice: Approaches and experiences. In Proceedings of the International Workshop on Machine Learning Technologies in Software Engineering, Lawrence, KS, USA, 12 November 2011; pp. 55–58.
29. Pospieszny, P. Software estimation: Towards prescriptive analytics. In Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement, Gothenburg, Sweden, 25–27 October 2017; pp. 221–226.
30. ManikReddy, P.; Iyer, J. Effective collaboration across the globe through digital dash boards and machine learning. In Proceedings of the 2018 IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE), Gothenburg, Sweden, 6 December 2018; pp. 30–34.
31. Moharreri, K.; Sapre, A.V.; Ramanathan, J.; Ramnath, R. Cost-effective supervised learning models for software effort estimation in agile environments. In Proceedings of the 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Atlanta, GA, USA, 10–14 June 2016; pp. 135–140.
32. Hosni, M.; Idri, A.; Nassif, A.B.; Abran, A. Heterogeneous ensembles for software development effort estimation. In Proceedings of the 2016 3rd International Conference on Soft Computing & Machine Intelligence (ISCMI), Dubai, United Arab Emirates, 23–25 November 2016; pp. 174–178.
33. Samath, S.; Udalagama, D.; Kurukulasooriya, H.; Premarathne, D.; Thelijjagoda, S. Collabcrew—An intelligent tool for dynamic task allocation within a software development team. In Proceedings of the 2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Malabe, Sri Lanka, 6–8 December 2017; pp. 1–9.
34. Li, Y.; Huang, Z.; Wang, Y.; Fang, B. Evaluating data filter on cross-project defect prediction: Comparison and improvements. IEEE Access 2017, 5, 25646–25656.
35. Ni, A.; Li, M. Poster: ACONA: Active Online Model Adaptation for Predicting Continuous Integration Build Failures. In Proceedings of the 2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion), Gothenburg, Sweden, 3 June 2018; pp. 366–367.
36. Sharma, P.; Singh, J. Machine Learning Based Effort Estimation Using Standardization. In Proceedings of the 2018 International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, 29 September 2018; pp. 716–720.
37. Papatheocharous, E.; Andreou, A.S. A hybrid software cost estimation approach utilizing decision trees and fuzzy logic. Int. J. Softw. Eng. Knowl. Eng. 2012, 22, 435–465.

[Downloaded from jots.knu.edu.af on 2026-07-04]

38. Hongming, Z.; Bin, F.; Xizhu, M.; Lijun, S.; Xiangzhou, X.Z.; Yong, H. A Cost-sensitive Intelligent Prediction Model for Outsourced Software Project Risk. In Proceedings of the WHICEB 2013 Proceedings, Wuhan, China, 25–26 May 2013.
39. Twala, B. Reasoning with Noisy Software Effort Data. Appl. Artif. Intell. 2014, 28, 533–554.
40. Wu, J.H.; Keung, J. Decision support for global software development with pattern discovery. In Proceedings of the 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 26–28 August 2016; pp. 182–185.
41. Rahman, M.T.; Islam, M.M. A Comparison of Machine Learning Algorithms to Estimate Effort in Varying Sized Software. In Proceedings of the 2019 IEEE Region 10 Symposium (TENSYMP), Kolkata, India, 7–9 June 2019; pp. 137–142.
42. Tumar, I.; Hassouneh, Y.; Turabieh, H.; Thaher, T. Enhanced binary moth flame optimization as a feature selection algorithm to predict software fault prediction. IEEE Access 2020, 8, 8041–8055.
43. Lopez-Martin, C.; Chavoya, A.; Meda-Campaña, M.E. A machine learning technique for predicting the productivity of practitioners from individually developed software projects. In Proceedings of the 15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Las Vegas, NV, USA, 30 June–2 July 2014; pp. 1–6.
44. Han, W.; Jiang, H.; Zhang, X.; Li, W. A Neural Network Based Algorithms for Project Duration Prediction. In Proceedings of the 2014 7th International Conference on Control and Automation, Hainan, China, 20–23 December 2014; pp. 60–63.
45. Basgalupp, M.P.; Barros, R.C.; da Silva, T.S.; de Carvalho, A.C. Software effort prediction: A hyper-heuristic decision-tree based approach. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, Coimbra, Portugal, 18–22 March 2013; pp. 1109–1116.
46. Twala, B.; Cartwright, M. Ensemble missing data techniques for software effort prediction. Intell. Data Anal. 2010, 14, 299–331.
47. Song, L.; Minku, L.L.; Yao, X. The impact of parameter tuning on software effort estimation using learning machines. In Proceedings of the 9th International Conference on Predictive Models in Software Engineering, Baltimore, MD, USA, 9 October 2013; pp. 1–10.
48. Minku, L.L.; Yao, X. How to make best use of cross-company data in software effort estimation? In Proceedings of the 36th International Conference on Software Engineering, Hyderabad, India, 31 May–7 June 2014; pp. 446–456.
49. Song, L.; Minku, L.L.; Yao, X. The potential benefit of relevance vector machine to software effort estimation. In Proceedings of the 10th International Conference on Predictive Models in Software Engineering, Torino, Italy, 17 September 2014; pp. 52–61.

[Downloaded from jots.knu.edu.af on 2026-07-04]

50. Scott, E.; Pfahl, D. Using developers' features to estimate story points. In Proceedings of the 2018 International Conference on Software and System Process, Gothenburg, Sweden, 26–27 May 2018; pp. 106–110.
51. Benala, T.R.; Bandarupalli, R. Least square support vector machine in analogy-based software development effort estimation. In Proceedings of the 2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, India, 23–25 December 2016; pp. 1–6.
52. Minku, L.L.; Hou, S. Clustering Dycom: An online cross-company software effort estimation study. In Proceedings of the 13th International Conference on Predictive Models and Data Analytics in Software Engineering, Toronto, ON, Canada, 8 November 2017; pp. 12–21.
53. Brady, A.; Menzies, T. Case-based reasoning vs parametric models for software quality optimization. In Proceedings of the 6th International Conference on Predictive Models in Software Engineering, Timisoara, Romania, 12–13 September 2010; pp. 1–10.
54. Borges, R.; Menzies, T. Learning to change projects. In Proceedings of the 8th International Conference on Predictive Models in Software Engineering, Lund, Sweden, 21–22 September 2012; pp. 11–18.
55. Jiang, Y.; Cukic, B. Misclassification cost-sensitive fault prediction models. In Proceedings of the 5th International Conference on Predictor Models in Software Engineering, Vancouver, BC, Canada, 18–19 May 2009; pp. 1–10.
56. Weld, D.S.; Dai, P. Execution control for crowdsourcing. In Proceedings of the 24th Annual ACM Symposium Adjunct on User Interface Software and Technology, Santa Barbara, CA, USA, 16–19 October 2011; pp. 57–58.
57. Shepperd, M. The scientific basis for prediction research. In Proceedings of the 8th International Conference on Predictive Models in Software Engineering, Lund, Sweden, 21–22 September 2012.
58. Karim, M.R.; Alam, S.D.A.; Kabeer, S.J.; Ruhe, G.; Baluta, B.; Mahmud, S. Applying data analytics towards optimized issue management: An industrial case study. In Proceedings of the 2016 IEEE/ACM 4th International Workshop on Conducting Empirical Studies in Industry (CESI), Austin, TX, USA, 17 May 2016; pp. 7–13.
59. Castro-Herrera, C.; Cleland-Huang, J. A machine learning approach for identifying expert stakeholders. In Proceedings of the 2009 Second International Workshop on Managing Requirements Knowledge, Atlanta, GA, USA, 1 September 2009; pp. 45–49.
60. Abdellatif, T.M. A Comparison Study Between Soft Computing and Statistical Regression Techniques for Software Effort Estimation. In Proceedings of the 2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE), Quebec, QC, Canada, 16 May 2018; pp. 1–5.
61. Mendes, E.; Turhan, B.; Rodríguez, P.; Freitas, V. Estimating the value of decisions relating to managing and developing software-intensive Products and Projects. In Proceedings of the 11th International Conference on Predictive Models and Data Analytics in Software Engineering, Beijing, China, 21 October 2015; pp. 1–4.

[Downloaded from jots.knu.edu.af on 2026-07-04]

62. Asif, M.; Ahmed, J. A Novel Case Base Reasoning and Frequent Pattern Based Decision Support System for Mitigating Software Risk Factors. IEEE Access 2020, 8, 102278–102291.
63. Qu, Y.; Yang, T.-Z. Research on occurrence frequency of IT projects risk based on fuzzy influence diagram. In Proceedings of the 2016 International Conference on Machine Learning and Cybernetics (ICMLC), Jeju, Korea, 13 July 2016; pp. 166–171.
64. Sree, S.R.; Ramesh, S. Analytical Structure of a Fuzzy Logic Controller for Software Development Effort Estimation. In Computational Intelligence in Data Mining—Volume 1; Springer: Berlin, Germany, 2016; pp. 209–216.
65. Raza, M.B.; Kirkham, T.; Harrison, R.; Monfared, R.; Haq, I.; Wood, S. Evolving knowledge based product lifecycle management from a digital ecosystem to support automated manufacturing. In Proceedings of the International Conference on Management of Emergent Digital EcoSystems, Lyon, France, 27–30 October 2009; pp. 437–441.
66. Yang, G.; Zhang, T.; Lee, B. Utilizing a multi-developer network-based developer recommendation algorithm to fix bugs effectively. In Proceedings of the 29th Annual ACM Symposium on Applied Computing, Gyeongju, Korea, 28 March 2014; pp. 1134–1139.
67. Amasaki, S.; Lokan, C. A Virtual Study of Moving Windows for Software Effort Estimation Using Finnish Datasets. In Proceedings of the International Conference on Product-Focused Software Process Improvement, Innsbruck, Austria, 28 October 2017; pp. 71–79.
68. Qahtani, A.M. An Empirical Study of Agile Testing in A Distributed Software Development Project. In Proceedings of the 2020 3rd International Conference on Geoinformatics and Data Analysis, Marseille, France, 17 April 2020; pp. 110–114.
69. Bruegge, B.; David, J.; Helming, J.; Koegel, M. Classification of tasks using machine learning. In Proceedings of the 5th International Conference on Predictor Models in Software Engineering, Vancouver, BC, Canada, 18–19 May 2009; pp. 1–11.
70. Minku, L.L.; Yao, X. Software effort estimation as a multiobjective learning problem. ACM Trans. Softw. Eng. Methodol. (TOSEM) 2013, 22, 1–32.
71. Shivhare, J.; Rath, S.K. Software effort estimation using machine learning techniques. In Proceedings of the 7th India Software Engineering Conference, Noida, Chennai, India, 21 February 2014; pp. 1–6.
72. Ramaswamy, V.; Suma, V.; Pushphavathi, T. An approach to predict software project success by cascading clustering and classification. In Proceedings of the International Conference on Software Engineering and Mobile Application Modelling and Development (ICSEMA 2012), Chennai, India, 21 December 2012.
73. Iwata, K.; Nakashima, T.; Anan, Y.; Ishii, N. Effort estimation for embedded software development projects by combining machine learning with classification. In Proceedings of the 2016 4th Intl Conf on Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud

[Downloaded from jots.knu.edu.af on 2026-07-04]

Computing, Data Science & Engineering (ACIT-CSII-BCD), Las Vegas, NV, USA, 14 December 2016; pp. 265–270.
74. Ionescu, V.-S. An approach to software development effort estimation using machine learning. In Proceedings of the 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 9 September 2017; pp. 197–203.
75. BaniMustafa, A. Predicting software effort estimation using machine learning techniques. In Proceedings of the 2018 8th International Conference on Computer Science and Information Technology (CSIT), Amman, Jordan, 12 July 2018; pp. 249–256.
76. Menzies, T.; Bird, C.; Zimmermann, T.; Schulte, W.; Kocaganeli, E. The inductive software engineering manifesto: Principles for industrial data mining. In Proceedings of the International Workshop on Machine Learning Technologies in Software Engineering, Lawrence, KS, USA, 12 November 2011; pp. 19–26.
77. Dehghan, A.; Blincoe, K.; Damian, D. A hybrid model for task completion effort estimation. In Proceedings of the 2nd International Workshop on Software Analytics, Seattle, WA, USA, 13 November 2016; pp. 22–28.
78. Tollin, I.; Fontana, F.A.; Zanoni, M.; Roveda, R. Change prediction through coding rules violations. In Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, Karlskrona, Sweden, 15–16 June 2017; pp. 61–64.
79. Hu, Y.; Zhang, X.; Sun, X.; Liu, M.; Du, J. An intelligent model for software project risk prediction. In Proceedings of the 2009 International Conference on Information Management, Innovation Management and Industrial Engineering, Xi'an, China, 27 December 2009; pp. 629–632.
80. Manalif, E.; Capretz, L.F.; Nassif, A.B.; Ho, D. Fuzzy-ExCOM software project risk assessment. In Proceedings of the 2012 11th International Conference on Machine Learning and Applications, Boca Raton, FL, USA, 15 December 2012; pp. 320–325.
81. Rana, R.; Staron, M. Machine learning approach for quality assessment and prediction in large software organizations. In Proceedings of the 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 25 September 2015; pp. 1098–1101.
82. Tariq, S.; Usman, M.; Wong, R.; Zhuang, Y.; Fong, S. On learning software effort estimation. In Proceedings of the 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI), Bali, Indonesia, 9 December 2015; pp. 79–84.
83. Kumar, L.; Rath, S.; Sureka, A. An empirical analysis on effective fault prediction model developed using ensemble methods. In Proceedings of the 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), Turin, Italy, 8 July 2017; pp. 244–249.
84. Hu, Y.; Feng, B.; Mo, X.; Zhang, X.; Ngai, E.W.T.; Fan, M.; Liu, M. Cost-sensitive and ensemble-based prediction model for outsourced software project risk prediction. Decis. Support Syst. 2015, 72, 11–23.

[Downloaded from jots.knu.edu.af on 2026-07-04]

85. Pospieszny, P.; Czarnacka-Chrobot, B.; Kobylinski, A. An effective approach for software project effort and duration estimation with machine learning algorithms. J. Syst. Softw. 2018, 137, 184–196.
86. Lochmann, K.; Ramadani, J.; Wagner, S. Are comprehensive quality models necessary for evaluating software quality? In Proceedings of the 9th International Conference on Predictive Models in Software Engineering, Baltimore, MD, USA, 9 October 2013; pp. 1–9.
87. Chen, N.; Hoi, S.C.; Xiao, X. Software process evaluation: A machine learning framework with application to defect management process. Empir. Softw. Eng. 2014, 19, 1531–1564.
88. Song, Q.; Zhu, X.; Wang, G.; Sun, H.; Jiang, H.; Xue, C.; Xu, B.; Song, W. A machine learning based software process model recommendation method. J. Syst. Softw. 2016, 118, 85–100.
89. Fitzgerald, C.; Letier, E.; Finkelstein, A. Early failure prediction in feature request management systems. In Proceedings of the 2011 IEEE 19th International Requirements Engineering Conference, Trento, Italy, 2 September 2011; pp. 229–238.
90. Joseph, H.R. Poster: Software Development Risk Management: Using Machine Learning for Generating Risk Prompts. In Proceedings of the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, Florence, Italy, 24 May 2015; pp. 833–834.
91. ERTUĞRUL, E.; Baytar, Z.; ÇATAL, Ç.; MURATLI, Ö.C. Performance tuning for machine learning-based software development effort prediction models. Turk. J. Electr. Eng. Comput. Sci. 2019, 27, 1308–1324.
92. Colomo-Palacios, R.; González-Carrasco, I.; López-Cuadrado, J.L.; Trigo, A.; Varajao, J.E. I-Competere: Using applied intelligence in search of competency gaps in software project managers. Inf. Syst. Front. 2014, 16, 607–625.
93. Nassif, A.B.; Azzeh, M.; Capretz, L.F.; Ho, D. Neural network models for software development effort estimation: A comparative study. Neural Comput. Appl. 2016, 27, 2369–2381. [Green Version]
94. Desai, V.S.; Mohanty, R. ANN-Cuckoo Optimization Technique to Predict Software Cost Estimation. In Proceedings of the 2018 Conference on Information and Communication Technology (CICT), Jabalpur, India, 28 October 2018; pp. 1–6.
95. Schleier-Smith, J. An architecture for Agile machine learning in real-time applications. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, 13 August 2015; pp. 2059–2068.
96. Volf, Z.; Shmueli, E. Screening heuristics for project gating systems. In Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, Paderborn, Germany, 8 August 2017; pp. 872–877.
97. Liyi, M.; Shiyu, Z.; Jian, G. A project risk forecast model based on support vector machine. In Proceedings of the 2010 IEEE International Conference on Software Engineering and Service Sciences, Beijing, China, 18 July 2010; pp. 463–466.

[Downloaded from jots.knu.edu.af on 2026-07-04]

98. Lopez-Martin, C.; Banitaan, S.; Garcia-Floriano, A.; Yanez-Marquez, C. Support vector regression for predicting the enhancement duration of software projects. In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 21 December 2017; pp. 562–567.
99. Chou, J.-S.; Cheng, M.-Y.; Wu, Y.-W.; Wu, C.-C. Forecasting enterprise resource planning software effort using evolutionary support vector machine inference model. Int. J. Proj. Manag. 2012, 30, 967–977.
100. Song, L.; Minku, L.L.; Yao, X. Software effort interval prediction via Bayesian inference and synthetic bootstrap resampling. Acm Trans. Softw. Eng. Methodol. (TOSEM) 2019, 28, 1–46.
101. Dahab, S.A.; Porras, J.J.H.; Maag, S. A Software Measurement Plan Management Guided by an Automated Metrics Suggestion Framework. In Proceedings of the 2017 European Conference on Electrical Engineering and Computer Science (EECS), Bern, Switzerland, 19 November 2017; pp. 9–16.
102. Koroglu, Y.; Sen, A.; Kutluay, D.; Bayraktar, A.; Tosun, Y.; Cinar, M.; Kaya, H. Defect prediction on a legacy industrial software: A case study on software with few defects. In Proceedings of the 2016 IEEE/ACM 4th International Workshop on Conducting Empirical Studies in Industry (CESI), Austin, TX, USA, 17 May 2016; pp. 14–20.
103. Azzeh, M.; Banitaan, S. An Application of Classification and Class Decomposition to Use Case Point Estimation Method. In Proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 11 December 2015; pp. 1268–1271.
104. Petkovic, D.; Sosnick-Pérez, M.; Huang, S.; Todtenhoefer, R.; Okada, K.; Arora, S.; Sreenivasen, R.; Flores, L.; Dubey, S. Setap: Software engineering teamwork assessment and prediction using machine learning. In Proceedings of the 2014 IEEE Frontiers in Education Conference (FIE) Proceedings, Madrid, Spain, 25 October 2014; pp. 1–8.
105. del Águila, I.M.; Sagrado, J.D. Requirement risk level forecast using Bayesian networks classifiers. Int. J. Softw. Eng. Knowl. Eng. 2011, 21, 167–190. [Green Version]
106. Alsri, A.; Almuhammadi, S.; Mahmood, S. A model for work distribution in global software development based on machine learning techniques. In Proceedings of the 2014 Science and Information Conference, London, UK, 29 August 2014; pp. 399–403.
107. Miandoab, E.E.; Gharehchopogh, F.S. A novel hybrid algorithm for software cost estimation based on cuckoo optimization and k-nearest neighbors algorithms. Eng. Technol. Appl. Sci. Res. 2016, 6, 1018–1022.
108. Basgalupp, M.P.; Barros, R.C.; Ruiz, D.D. Predicting software maintenance effort through evolutionary-based decision trees. In Proceedings of the 27th Annual ACM Symposium on Applied Computing, Riva del Garda, Italy, 29 March 2012; pp. 1209–1214.

[Downloaded from jots.knu.edu.af on 2026-07-04]

109. Bakır, A.; Turhan, B.; Bener, A.B. A new perspective on data homogeneity in software cost estimation: A study in the embedded systems domain. Softw. Qual. J. 2010, 18, 57–80.
110. Helming, J.; Koegel, M.; Hodaie, Z. Towards automation of iteration planning. In Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications, Orlando, FL, USA, 29 October 2009; pp. 965–972.
111. Choetkiertikul, M.; Dam, H.K.; Tran, T.; Pham, T.; Ghose, A.; Menzies, T. A deep learning model for estimating story points. IEEE Trans. Softw. Eng. 2018, 45, 637–656. [Green Version]
112. Niinimäki, T.; Piri, A.; Hynninen, P.; Lassenius, C. Studying communication in agile software development: A research framework and pilot study. In Proceedings of the ICMI-MLMI'09 Workshop on Multimodal Sensor-Based Systems and Mobile Phones for Social Computing, Cambridge, MA, USA, 6 November 2009; pp. 1–4.
113. Pechau, J. Rafting the agile waterfall: Value based conflicts of agile software development. In Proceedings of the 16th European Conference on Pattern Languages of Programs, Irsee, Germany, 17 July 2011; pp. 1–15.
114. Gousios, G.; Zaidman, A. A dataset for pull-based development research. In Proceedings of the 11th Working Conference on Mining Software Repositories, Hyderabad, India, 18 May 2014; pp. 368–371.
115. Makris, C.; Vikatos, P.; Visser, J. Classification model for predicting cost slippage in governmental ICT projects. In Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain, 17 April 2015; pp. 1238–1241.
116. Qu, Y.; Tang, X.-L. Software project risk assessing model based on credal networks. In Proceedings of the 2010 International Conference on Machine Learning and Cybernetics, Qingdao, China, 14 July 2010; pp. 1976–1979.
117. Gouthaman, P.; Sankaranarayanan, S. Agile Software Risk Management Architecture for IoT-Fog based systems. In Proceedings of the 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 14 December 2018; pp. 48–51.
118. Andrés, J.D.; Landajo, M.; Lorca, P. Using nonlinear quantile regression for the estimation of software cost. In Proceedings of the International Conference on Hybrid Artificial Intelligence Systems, Oviedo, Spain, 22 June 2018; pp. 422–432.
119. Pa, R.S.; Snsvsc, R. Improving efficiency of fuzzy models for effort estimation by cascading & clustering techniques. Procedia Comput. Sci. 2016, 85, 278–285.
120. Nassif, A.B.; Azzeh, M.; Idri, A.; Abran, A. Software development effort estimation using regression fuzzy models. Comput. Intell. Neurosci. 2019, 2019.
121. Mohebzada, J.G.; Ruhe, G.; Eberlein, A. SRP-plugin: A strategic release planning plug-in for visual studio 2010. In Proceedings of the 1st Workshop on Developing Tools as Plug-ins, Honolulu, HI, USA, 28 May 2011; pp. 36–39.

[Downloaded from jots.knu.edu.af on 2026-07-04]

122. Baolong, Y.; Hong, W.; Haodong, Z. Research and application of data management based on Data Management Maturity Model (DMM). In Proceedings of the 2018 10th International Conference on Machine Learning and Computing, Macau, China, 10 February 2018; pp. 157–160.
123. Sigweni, B. Feature weighting for case-based reasoning software project effort estimation. In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, London, UK, 13–14 May 2014; pp. 1–4.
124. Huang, Z.-W. Cost Estimation of Software Project Development by Using Case-Based Reasoning Technology with Clustering Index Mechanism. In Proceedings of the 2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC), Kaohsiung, Taiwan, 7–9 December 2009; pp. 1049–1052.
125. Wang, Y.-H.; Jia, J.; Qu, Y. The "Earth-Moon" model on software project risk management. In Proceedings of the 2010 International Conference on Machine Learning and Cybernetics, Qingdao, China, 14 July 2010; pp. 1999–2003.
126. Amasaki, S.; Kawata, K.; Yokogawa, T. Improving cross-project defect prediction methods with data simplification. In Proceedings of the 2015 41st Euromicro Conference on Software Engineering and Advanced Applications, Madeira, Portugal, 28 August 2015; pp. 96–103.
127. Nassif, A.B.; Capretz, L.F.; Ho, D.; Azzeh, M. A treeboost model for software effort estimation based on use case points. In Proceedings of the 2012 11th International Conference on Machine Learning and Applications, Boca Raton, FL, USA, 15 December 2012; pp. 314–319.
128. Wagner, S. A literature survey of the quality economics of defect-detection techniques. In Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering, Rio de Janeiro Brazil, 21–22 September 2006; pp. 194–203.
129. Pressman, R.S. Software Engineering: A Practitioner's Approach; Palgrave Macmillan: London, UK, 2005.
130. Nassif, A.B.; Ho, D.; Capretz, L.F. Towards an early software estimation using log-linear regression and a multilayer perceptron model. J. Syst. Softw. 2013, 86, 144–160. [Green Version]
131. Menzies, T.; Mizuno, O.; Takagi, Y.; Kikuno, T. Explanation vs performance in data mining: A case study with predicting runaway projects. J. Softw. Eng. Appl. 2009, 2, 221. [Green Version]
132. Kitchenham, B.; Mendes, E.; Travassos, G.H. A systematic review of cross- vs. In within-company cost estimation studies. In Proceedings of the 10th International Conference on Evaluation and Assessment in Software Engineering (EASE) 10, Swindon, UK, 11 April 2006; pp. 1–10.
133. Mahdi, M.N.; Yusof, M.Z.M.H.A.; Cheng, L.K.; Azmi, M.S.M.; Ahmad, A.R. Design and Development of Machine Learning Technique for Software Project Risk Assessment-A Review. In Proceedings of the 2020 8th International Conference on Information Technology and Multimedia (ICIMU), Selangor, Malaysia, 26 August 2020; pp. 354–362.

[Downloaded from jots.knu.edu.af on 2026-07-04]

134. Lee, T.; Gu, T.; Baik, J. MND-SCEMP: An empirical study of a software cost estimation modeling process in the defense domain. Empir. Softw. Eng. 2014, 19, 213–240.
135. Mitchell, S.M.; Seaman, C.B. A comparison of software cost, duration, and quality for waterfall vs iterative and incremental development: A systematic review. In Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement, Lake Buena Vista, FL, USA, 16 October 2009; pp. 511–515.
136. Jorgensen, M.; Shepperd, M. A systematic review of software development cost estimation studies. IEEE Trans. Softw. Eng. 2006, 33, 33–53. [Green Version]
137. González-Ladrón-de-Guevara, F.; Fernández-Diego, M.; Lokan, C. The usage of ISBSG data fields in software effort estimation: A systematic mapping study. J. Syst. Softw. 2016, 113, 188–215. [Green Version]
138. Iranmanesh, S.H.; Hojati, Z.T. Intelligent systems in project performance measurement and evaluation. In Proceedings of the Intelligent Techniques in Engineering Management, 5 May 2015; Springer: Berlin, Germany, 2015; pp. 581–619.
139. Mellegård, N.; Staron, M. Characterizing model usage in embedded software engineering: A case study. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, Copenhagen, Denmark, 23–26 August 2010; pp. 245–252.
140. Antonellis, P.; Antoniou, D.; Kanellopoulos, Y.; Makris, C.; Theodoridis, E.; Tjortjis, C.; Tsirakis, N.; A data mining methodology for evaluating maintainability according to ISO/IEC-9126 software engineering–product quality standard. Special Session on System Quality and Maintainability-SQM2007. 2007. Available online: https://www.ihu.edu.gr/tjortjis/A%20Data%20Mining%20Methodology%20for%20Evaluating%20Maintainability%20according%20to%20SQM07.pdf (accessed on 5 May 2021).
141. Azar, D.; Harmanani, H.; Korkmaz, R. A hybrid heuristic approach to optimize rule-based software quality estimation models. Inf. Softw. Technol. 2009, 51, 1365–1376.
142. Mahdi, M.N.; Azmi, M.S.M.; Cheng, L.K.; Yusof, A.; Ahmad, A.R. Software Project Management Using Machine Learning Technique-A Review. In Proceedings of the 2020 8th International Conference on Information Technology and Multimedia (ICIMU), Selangor, Malaysia, 26 August 2020; pp. 363–370.
143. Zhang, H.; Dai, G. The strategy of traffic congestion management based on case-based reasoning. Int. J. Syst. Assur. Eng. Manag. 2019, 10, 142–147.
144. Agrawal, A.; Menzies, T. “Better Data” is Better than “Better Data Miners” (Benefits of Tuning SMOTE for Defect Prediction). arXiv 2017, arXiv:1705.03697.
145. Amasaki, S.; Takahara, Y.; Yokogawa, T. Performance evaluation of windowing approach on effort estimation by analogy. In Proceedings of the 2011 Joint Conference of the 21st International Workshop on Software

[Downloaded from jots.knu.edu.af on 2026-07-04]

Measurement and the 6th International Conference on Software Process and Product Measurement, Nara, Japan, 4 November 2011; pp. 188–195.
146. Arcuri, A.; Briand, L. A practical guide for using statistical tests to assess randomized algorithms in software engineering. In Proceedings of the 2011 33rd International Conference on Software Engineering (ICSE), Honolulu, HI, USA, 28 May 2011; pp. 1–10.
147. Wan, S.; Li, D.; Gao, J.; Li, J. A knowledge based machine tool maintenance planning system Using case-based reasoning techniques. Robot. Comput. Integr. Manuf. 2019, 58, 80–96.
148. Kaur, A.; Kaur, K. Effort Estimation for Mobile Applications Using Use Case Point (UCP). In Proceedings of the Smart Innovations in Communication and Computational Sciences, Bangkok, Thailand, 30 June 2019; pp. 163–172.
149. Srivastava, A.; Singh, S.; Abbas, S.Q. Performance Measure of the Proposed Cost Estimation Model: Advance Use Case Point Method. In Proceedings of the Soft Computing: Theories and Applications, Lviv, Ukraine, 20 September 2019; pp. 223–233.
150. Larsson, S.; Jansson, M.; Boholm, Å. Expert stakeholders' perception of nanotechnology: Risk, benefit, knowledge, and regulation. J. Nanoparticle Res. 2019, 21, 57.
151. Poth, A.; Sasabe, S.; Mas, A.; Mesquida, A.L. Lean and agile software process improvement in traditional and agile environments. J. Software Evol. Process. 2019, 31, e1986.
152. Sievi-Korte, O.; Beecham, S.; Richardson, I. Challenges and recommended practices for software architecting in global software development. Inf. Softw. Technol. 2019, 106, 234–253.
153. Lops, P.; Gemmis, M.D.; Semeraro, G. Content-based recommender systems: State of the art and trends. In Recommender Systems Handbook; Springer: Berlin, Germany, 2011; pp. 73–105.
154. Fauzi, S.S.M.; Ramli, N.; Nasir, M.H.N.M. Software Configuration Management A Result from the Assessment and its Recommendation. In Proceedings of the 2009 International Conference on Information Management and Engineering, Kuala Lumpur, Malaysia, 3–5 April 2009; pp. 416–419.
155. Khomyakov, I.; Mirgalimova, R.; Sillitti, A. An investigation of the project management approaches of agile and plan-based companies. In Proceedings of the 35th Annual ACM Symposium on Applied Computing, Brno, Czech Republic, 3 April 2020; pp. 1662–1665.
156. Prakash, B.; Viswanathan, V.A. Survey on Software Estimation Techniques in Traditional and Agile Development Models. Indones. J. Electr. Eng. Comput. Sci. 2017, 7, 867–876.
157. Picha, P.; Brada, P. Software process anti-pattern detection in project data. In Proceedings of the 24th European Conference on Pattern Languages of Programs, Irsee, Germany, 19 July 2019; pp. 1–12.
158. Kappen, T.H.; Vergouwe, Y.; Wolfswinkel, L.V.; Kalkman, C.; Moons, K.; Klei, W.V. Impact of adding therapeutic recommendations to risk assessments

[Downloaded from jots.knu.edu.af on 2026-07-04]

from a prediction model for postoperative nausea and vomiting. Br. J. Anaesth. 2015, 114, 252–260.
159. Kanimozhi, U.; Ganapathy, S.; Manjula, D.; Kannan, A. An intelligent risk prediction system for breast cancer using fuzzy temporal rules. Natl. Acad. Sci. Lett. 2019, 42, 227–232.
160. Matharu, G.S.; Mishra, A.; Singh, H.; Upadhyay, P. Empirical study of agile software development methodologies: A comparative analysis. ACM SIGSOFT Softw. Eng. Notes 2015, 40, 1–6.
161. Yang, M.Q.; Elnitski, L.L. Prediction-based approaches to characterize bidirectional promoters in the mammalian genome. BMC Genom. 2008, 9, S2. [PubMed][Green Version]
162. Nagwani, N.K.; Bhansali, A. A data mining model to predict software bug complexity using bug estimation and clustering. In Proceedings of the 2010 International Conference on Recent Trends in Information, Telecommunication and Computing, Kerala, India, 13 March 2010; pp. 13–17.
163. Shan, X.; Jiang, G.; Huang, T. A framework of estimating software project success potential based on association rule mining. In Proceedings of the 2009 International Conference on Management and Service Science, Beijing, China, 22 September 2009; pp. 1–4.
164. Khan, B.; Iqbal, D.; Badshah, S. Cross-Project Software Fault Prediction Using Data Leveraging Technique to Improve Software Quality. In Proceedings of the Evaluation and Assessment in Software Engineering, Trondheim, Norway, 17 April 2020; pp. 434–438.
165. Chelly, Z.; Elouedi, Z. Improving the dendritic cell algorithm performance using fuzzy-rough set theory as a pattern discovery technique. In Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA, Ostrava, Czech Republic, 25 June 2014; pp. 23–32.
166. Ghotra, B.; McIntosh, S.; Hassan, A.E. Revisiting the impact of classification techniques on the performance of defect prediction models. In Proceedings of the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, Florence, Italy, 16–24 May 2015; pp. 789–800.
167. Li, J.; Ji, X.; Jia, Y.; Zhu, B.; Wang, G.; Li, Z.; Liu, X. Hard drive failure prediction using classification and regression trees. In Proceedings of the 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Atlanta, GA, USA, 26 June 2014; pp. 383–394.
168. Ryu, D.; Choi, O.; Baik, J. Value-cognitive boosting with a support vector machine for cross-project defect prediction. Empir. Softw. Eng. 2016, 21, 43–71.

[Downloaded from jots.knu.edu.af on 2026-07-04]