

مدیریت پروژه نرم افزار با استفاده از تکنیک یادگیری ماشین – مروری

چکیده

برنامه ریزی و ارزیابی مدیریت پروژه در فعالیتهای عملکردی، از اهمیت بالایی برخوردار است. بدون یک برنامه واقعی و منطقی، مدیریت پروژه به طور کارآمد، آسان نیست. این مقاله مروری گسترده و جامع از مقالات مربوط به کاربرد یادگیری ماشین در مدیریت پروژه نرم افزاری را ارائه می دهد. علاوه بر این، این مقاله تجزیه و تحلیل ادبیات گسترده ای از (۱) یادگیری ماشین، (۲) مدیریت پروژه نرم افزار، و (۳) تکنیک های سه کتابخانه اصلی، Science Directs، Web Science، و IEEE Explore را ارائه می دهد. یکصد و یازده مقاله در این سه مخزن به چهار دسته تقسیم می شوند. دسته اول شامل مقالات تحقیقی و نظرسنجی در مورد مدیریت پروژه نرم افزاری است. دسته دوم شامل مقالاتی است که مبتنی بر روش ها و استراتژی های یادگیری ماشین هستند که در پروژه ها استفاده می شوند. دسته سوم شامل مطالعات مربوط به مراحل و آزمون هایی است که پارامترهای مورد استفاده در مدیریت یادگیری ماشین هستند و کلاس های نهایی نتایج حاصل از مطالعه، مشارکت مطالعات در تولید، و ارتقای پیش بینی پروژه های یادگیری ماشین می باشند. مشارکت ما همچنین دیدگاه جامع تری دارد و زمینه ای را ارائه می دهد که برای کار بالقوه در مدیریت ریسک پروژه مهم است. در نتیجه، ما نشان داده ایم که ارزیابی ریسک پروژه توسط یادگیری ماشین در به حداقل رساندن ضرر پروژه موفق تر است، در نتیجه احتمال موفقیت پروژه را افزایش می دهد و راهی جایگزین برای کاهش موثر احتمالات شکست پروژه و افزایش نسبت خروجی برای رشد را ارائه می کند و همچنین تجزیه و تحلیل پیش بینی خطای نرم افزار را بر اساس دقت تسهیل می کند.

کلید واژه ها: تکنیک یادگیری ماشین؛ برآورد پروژه نرم افزار؛ برآورد نرم افزار؛ مدیریت پروژه نرم افزار؛ ارزیابی ریسک پروژه

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

۱. معرفی

بهبود کارایی و حفظ پایداری یک پروژه نرم افزاری از موانعی است که مدیران پروژه با آن روبرو هستند. احتمال شکست پروژه عموماً به دلیل کمبود دانش، مهارت، منابع و فناوری در حین اجرای پروژه است. [1،2،3]

دانشی که از مجموعه داده های پروژه های گذشته به دست می آید، می تواند برای توسعه مدل های پیش بینی با استفاده از روش شناسی ریاضی، از جمله رگرسیون خطی و مطالعه روش های مرتبط با یادگیری ماشین (ML)، مانند شبکه مصنوعی شبکه (ANN) و ماشین های بردار پشتیبانی (SVM). روش های پیش بینی، روشی را ارائه می دهند که بر شواهد فعلی و گذشته ی پروژه برای پیش بینی آینده ی پروژه متمرکز است. الگوریتم های مختلف ML به دلیل زیاد بودن تعداد آنها، تعدادی از آنها هنوز مورد مطالعه قرار نگرفته اند. با توجه به یافته های موارد مطالعه شده در این مقاله، دلیل استفاده از پروژه های خودکار، مسائل ارزیابی مدیریت پروژه و روش شناسی توسعه ML مطرح می شود. نتایج تجربی مورد ارزیابی قرار خواهند گرفت.

اگرچه ادبیات پروژه، موفقیت و شکست پروژه را توصیف می‌کند، بحث‌های طولانی در مورد اینکه چگونه می‌توان پیشرفت پروژه را اندازه‌گیری کرد وجود دارد. درک عملکرد پروژه و ارزیابی موفقیت پروژه متفاوت است [۴]. هیوز^۱ و سایر اعضا [۵] در موسسه مدیریت پروژه [6] (PMI) بین متغیرهای موفقیت پروژه و عملکرد پروژه تفاوت قائل می‌شوند.

آستانه‌های پیشرفت پروژه برای اندازه‌گیری موفقیت و شکست یک پروژه ارزیابی می‌شوند. بازخورد نیز برای پیشرفت پروژه در نظر گرفته می‌شود. از لحاظ تاریخی، ارائه نتایج لازم و استفاده از منابع انتخاب شده توسط یک پروژه موفق در طول مدت پروژه‌ی مشخص [۷] متمایز می‌شود. [6] PMI ابتکاراتی را شناسایی می‌کند که با موفقیت به اهداف پروژه، معیارها و جاه طلبی‌های ذینفعان دست می‌یابد. محققانی مانند کومو علاءدوانی^۲ [۸]، کیتس و ملاقاسمی^۳ [۹]، پارسونز^۴ [۱۰]، و روزنفلد^۵ [۱۱] اثرات معیارهای کلاسیک هدف، مانند هزینه پروژه (بالا تر 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 همچنین به هزینه‌ها، هزینه‌های زمانی، اهداف محقق نشده یا حتی لغو پروژه‌ها کمک می‌کند و به یک خطر طبیعی و ناخواسته پروژه و اثرات نامطلوب بر قابلیت اطمینان پروژه‌های نرم‌افزاری تبدیل می‌شود [۱۲]. الزامات اصلاح مشخصات (از نظر گسترش چندگانه، حذف و اصلاح) در طول پروژه توسعه نرم افزار از جمله عوامل اصلی ایجاد مشکلات برای پروژه است. [1314،15،16]

بخش دوم این مقاله شامل تجزیه و تحلیل توضیحی در مورد اصول ارزیابی پروژه نرم افزاری و فناوری آموزش کامپیوتر است. ساختار این مقاله به شرح زیر است. بخش ۳ رویکرد را تعریف می‌کند، از جمله منبع مطالب، الزامات واجد شرایط بودن تحقیق، بررسی ادبیات سیستماتیک (SLR)، و اثرات نتایج جستجو از انتشارات. همچنین سؤالات تحقیق (RQs) را برای این تحقیق شناسایی می‌کند و با تهدیدهایی که اعتبار آنها را تهدید می‌کند و به چالش‌های عمده برای اثربخشی SLRها اشاره می‌کند. پرس و جوهای هر موضوع از سه وب سایت مقاله به چهار کلاس تقسیم شدند، طبقه بندی ادبی در مدیریت پروژه نرم افزاری با استفاده از تکنیک ML. بخش ۴ به الهام، مشکلات و توصیه‌ها در آن حوزه تحقیقاتی و رویکردی مدرن برای مدیریت ریسک پروژه‌های نرم‌افزاری می‌پردازد. در نهایت، بخش ۵ نتیجه‌گیری را ارائه می‌دهد.

۲. مطالعه مقدماتی

در این بخش، برخی از مفاهیم تخمین پروژه نرم افزاری و تکنیک یادگیری ماشین را بررسی و تصریح می‌کنیم.

۲.۱. تخمین تلاش نرم افزاری

پیش بینی تلاش و مدت زمان توسعه نرم افزار وظیفه حیاتی برای مدیریت پروژه نرم افزاری موثر (SPM)^۶ است. دقت و قابلیت اطمینان مکانیسم‌های پیش بینی نیز ضروری است. داشتن ارزیابی دقیق تلاش، به ویژه در مرحله اولیه پروژه نرم افزاری، ممکن است به طور قابل توجهی خطرات بالایی را که در طول توسعه، یک محصول نرم افزاری متحمل می‌شود، کاهش دهد. متأسفانه،

¹ - Hughes

² - Como Aladwani

³ - Cates and Mollaghasemi

⁴ - Parsons

⁵ - Rosenfeld

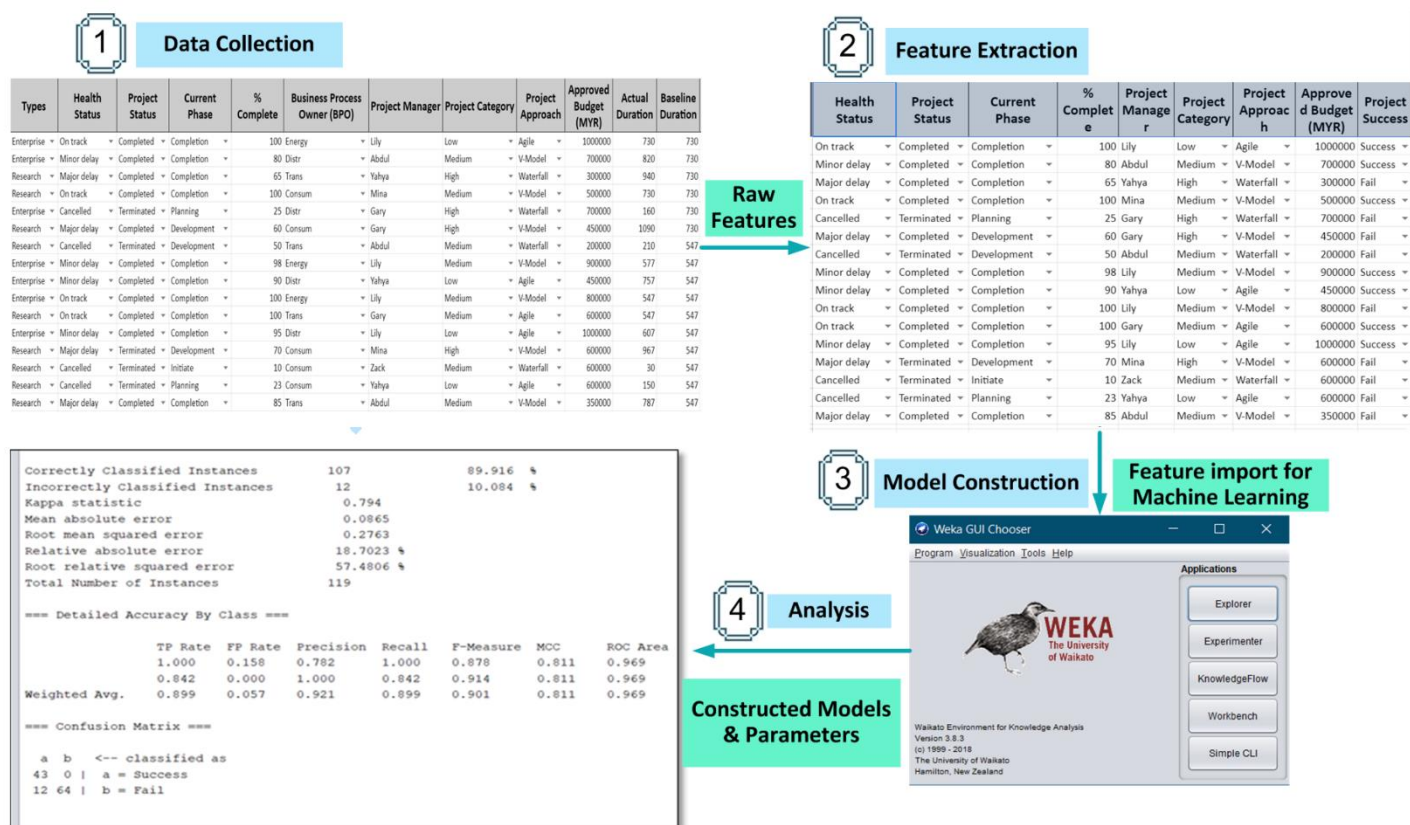
⁶ - Software Project Management

بسیاری از تکنیک‌های برآورد موجود، اغلب به طور اساسی اشتباه هستند و بیشتر پروژه‌ها با بیش از حد تلاش مواجه می‌شوند. با این حال، مشخص شد که برآورد پروژه نرم افزاری بر اساس الگوریتم‌های ML می‌تواند تخمین تلاش دقیق تری را ارائه دهد.

۲.۲. یادگیری ماشین (ML)

ML یک برنامه‌ی کاربردی از هوش مصنوعی است که سیستم‌هایی را برای یادگیری و بهبود از تجربه بدون برنامه ریزی صریح به طور خودکار فراهم می‌کند. به عبارت دیگر، هدف اصلی ML این است که به رایانه‌ها اجازه دهد تا به طور خودکار بدون دخالت یا کمک انسان یاد بگیرند و سپس اقدامات مورد نیاز را بر اساس آن تنظیم کنند. علاوه بر این، ML پردازش حجم عظیمی از اطلاعات را امکان پذیر می‌کند.

۳.۲. برآورد مدیریت پروژه نرم افزار بر اساس ML



شکل ۱ روش تخمین مدیریت پروژه نرم افزاری را نشان می‌دهد که می‌توان آن را به صورت زیر خلاصه کرد.

شکل ۱. نمونه‌ای از روش شناسی توسعه برآورد مدیریت پروژه نرم افزاری.

مرحله ۱. جمع آوری داده‌ها: استخراج توکن‌ها(نشانه‌ها)، توکن‌های کلمه، ویژگی‌های ضروری برای محاسبه تلاش در پروژه هستند. یک بخش کلیدی از مواد پروژه توسط توکن‌ها تعریف می‌شود. به عنوان مؤلفه اصلی مدل تخمین، از مدل‌سازی زبان Unigram در مورد توکن‌ها استفاده شد.

مرحله ۲. استخراج ویژگی: پس از استخراج توکن‌ها، ویژگی‌های پروژه برای تجزیه و تحلیل بیشتر انتخاب شد: فرکانس مدت-فرکانس معکوس سند (TF-IDF) هر توکن «فرکانس مدت» ابزاری برای محاسبه معنای عبارتی است که فراوانی کلمه و شمارش معکوس رکوردها یا کلمات از جمله این کلمه را در نظر می‌گیرد. این روشی برای اندازه‌گیری معنای یک کلمه است.

مرحله ۳. ساخت مدل: برای الگوریتم‌های طبقه‌بندی مختلف، ویژگی‌های مشتق شده به عنوان ورودی Weka استفاده می‌شود. در زیر مروری بر نحوه انتخاب الگوریتم‌های یادگیری موجود است: بیز ساده^۷، یک طبقه‌بندی احتمالی است که بر قضیه بیز متمرکز است که ویژگی‌های مستقل را از کلاس درس می‌گیرد. بیز ساده، با سادگی خود، با ابعاد بالای داده‌ها با ادعای ذهنی آزادی مبارزه می‌کند، که می‌تواند از رویکردهای طبقه‌بندی پیچیده‌تر نیز فراتر رود. درخت تصمیم J48: منبع باز جاو C4.5. این یک الگوریتم مولد درخت تصمیم است که در آن مجموعه‌ی تنینگ^۸ به صورت خطی قابل تفکیک نیست. درختان تصمیم به خوبی با نتایج آموزشی مطابقت دارند. جنگل تصادفی: طبقه‌بندی‌کننده دسته‌بندی است که از چندین درخت تصمیم و خروجی‌های کلاس تشکیل شده است که در آن مدل آماری گروه‌های خروجی درختان جداگانه هستند.

مرحله ۴. تجزیه و تحلیل: این مطالعه بهترین مدل‌های ML را برای عملکرد بهتر پیدا کرده و نشان می‌دهد که محاسبه ریسک پروژه با استفاده از یادگیری ماشین در به حداقل رساندن خطای پروژه مؤثرتر است، که احتمال پاسخ پروژه را بهبود می‌بخشد و راه جایگزینی برای کارآمدی برای کاهش احتمالات و افزایش نسبت خروجی برای رشد ارائه می‌دهد.

۳. روش شناسی

با جستجوی دو عبارت «یادگیری ماشین» و «مدیریت پروژه نرم‌افزار» مقالاتی را به دست آوردیم که با SPM مرتبط بودند. جستجو در سه کتابخانه دیجیتال انجام شد: (1) Web of Science (WOS)، به دلیل اینکه به مقالات تحقیقاتی چند رشته‌ای در زمینه‌های علم، هنر و غیره می‌پردازد، (2) IEEE به این دلیل که مقالاتی را ارائه می‌دهد که در زمینه مهندسی برق و الکترونیک تخصصی هستند، و (3) کتابخانه دیجیتال ACM که دارای پایگاه داده جامعی است که حاوی مقالات علمی در زمینه محاسبات و فناوری اطلاعات است.

مقالات و ادبیات مهم ارجاع شده توسط خروجی‌های جستجو بر اساس دو معیار انتخاب و دسته‌بندی شدند: (۱) استفاده از سه تکرار در فرآیند فیلتر کردن، که مقالات اضافی و تکراری را حذف می‌کند، به استثنای مقالات نامربوط با استفاده از عنوان. (۲) غربالگری اولیه را انجام می‌دهد و مقالات انتخاب شده با مطالعه دقیق نتایج جستجوی محدود شده صفحه را با استفاده از SPM دنبال می‌کند.

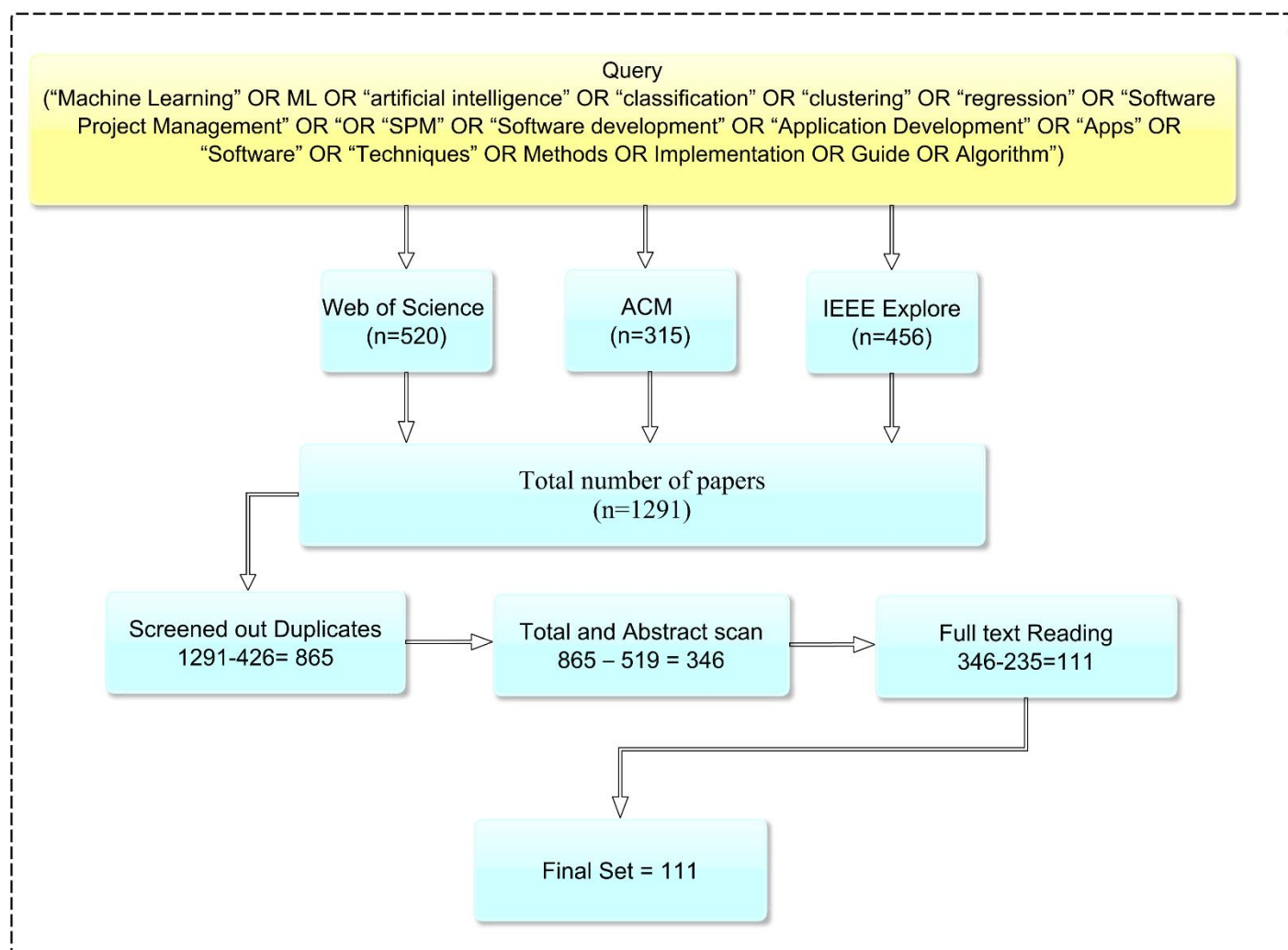
بسیاری از جستجوها در سه پایگاه داده‌ی ذکر شده در مارس ۲۰۲۰ با استفاده از چندین کلمه‌ی کلیدی (یا عبارات)، مانند "یادگیری ماشین" یا ML یا "هوش مصنوعی" یا "طبقه‌بندی" یا "خوشه‌بندی" یا "رگرسیون" و "مدیریت پروژه نرم‌افزاری" یا «SPM» یا «توسعه نرم‌افزار» یا «توسعه برنامه‌ها» یا «برنامه‌ها» یا «نرم‌افزار» و «تکنیک‌ها» یا «روش‌ها» یا «پیاده‌سازی» یا «راهنما» یا «الگوریتم» انجام شد.

شکل ۲ یک متن پرس و جو معمولی استفاده شده را نشان می‌دهد. ما نتایج جستجو را که مکاتبات، نامه‌ها، فصل‌های کتاب و غیره هستند، با استفاده از گزینه‌های پیشرفته موتورهای جستجو حذف کرده‌ایم. استثنائات برای به دست آوردن جدیدترین

⁷ - Naïve Bayes

⁸ - tanning collection

مقالات علمی و فقط مقالات دارای اهمیت زیادی هستند که قابلیت SPM را افزایش می‌دهند. تمرکز بر این است که تمامی مقالات و دست نوشته‌های علمی که تمامی معیارهای این اثر را برآورده می‌کنند، در بر گیرد. سپس، آن‌ها به کلاس‌هایی، یعنی: عمومی و درشت دانه تقسیم می‌شوند. مورد دوم در چهار بخش بعدی به دست آمده از نتایج مطالعه مورد بحث قرار می‌گیرد که در آن از موتور جستجوی Google Scholar برای تعیین جهت مطالعه استفاده شده است.



شکل ۲. راهنمای روش تحقیق.

شکل ۲ نشان می‌دهد که ۱۲۹۱ مقاله پس از انجام پرس و جوها جمع آوری شده است که از بین تمام اسناد، ۵۲۰ مقاله از WOS، ۴۵۶ مقاله از IEEE، و ۳۱۵ مقاله از کتابخانه‌های دیجیتال ACM. همه مقالات منتخب بین سال‌های ۲۰۰۹ و ۲۰۲۰ منتشر شدند. این مقالات بعداً به سه گروه تقسیم شدند. (۱) ۴۲۶ مقاله اضافی، (۲) ۵۱۹ مقاله بر اساس عناوین و چکیده‌ها نامربوط بودند و ۱۱۱ مقاله در معیارهای SPM قرار دارند.

همانطور که قبلاً مشخص شد، یک مقاله در صورتی که معیارهای انتخاب را برآورده نکند، که در زیر ذکر شده است، حذف می‌شود: (۱) مقاله به زبان انگلیسی نوشته نشده باشد. (۲) تکنیک‌ها و/یا روش‌ها تمرکز مقاله بودند. (۳) علاقه پژوهشی در مقاله فقط بر روی SPM بدون توسعه نرم افزار یا یادگیری ماشین متمرکز است.

علاوه بر این، پس از تکرار دوم، اگر SPM گنجانده نشده باشد، مقالات همچنان حذف می‌شوند یا: (۱) سهم مقاله هیچ جنبه‌ای از یادگیری ماشین و مدیریت پروژه را در نظر نمی‌گیرد. (۲) بحث روی مقاله فقط روی SPM متمرکز است و هیچ موضوع

دیگری را مورد بحث قرار نمی دهد. در این کار، مقاله ها تحت ML گسترده قرار می گیرند، به موجب آن، مقالات باقی مانده بعداً در دسته هایی دسته بندی می شوند که به چگونگی افزایش SPM می پردازند.

۱.۳. تهدید به اعتبار

مطالعات دیگر به چالش های قابل توجهی برای اثربخشی SLR ها اشاره کرده اند [۱۷، ۱۸]، و روندهای استفاده از الگوریتم های یادگیری ماشین، مجموعه داده های معیار، روش های اعتبارسنجی و معیارهای اندازه را برای تخمین تلاش نرم افزار برجسته کرده اند. چهار استراتژی مختلف برای به حداقل رساندن خطراتی که توسط این TTV ها به صورت استراتژیک ایجاد می شود، استفاده شد.

اول، اعتبار سازه: چارچوب با اجرای یک جستجوی دستی و خودکار جملات تأیید شد تا داده های SPM محاسبه شده از جمع آوری داده ها به حداقل برسد. علاوه بر این، مقالات انتخاب شده با تجزیه و تحلیل کامل نتایج جستجوی کاهش یافته، SPM را ارزیابی می کنند.

دوم، اعتبار داخلی: روش هایی که توسط [۱۷، ۱۸] برای حل اعتبار داخلی استفاده شد. علاوه بر این، برای جلوگیری از سوگیری در طول جستجوی جامع برای مقالات مجلات، تکنیکی که دو مرحله از جستجو را ترکیب می کند برای یک رویکرد انتخاب جامع استفاده شد. کلیه مقالات مورد علاقه از پایگاه های داده مورد استفاده برای تحقیقات مرتبط استخراج شده است [۱۷، ۱۸، ۱۹] و تحت فرآیندهای انتخاب کامل قرار گرفتند که در شکل ۲ نشان داده شده است.

سوم، اعتبار خارجی: اعتبار خارجی با ادغام ده سال چارچوب زمانی مطالعات SPM مورد بررسی قرار گرفت - که منجر به نتایج کلی شد. یک رابطه موازی بین مجموعه انباشته مقالات و مقالات موجود وجود دارد، که نشان می دهد این SLR می تواند یک گزارش تعمیم یافته را حفظ کند که با معیارهای اعتبار خارجی تحقیق مطابقت دارد.

در نهایت، اعتبار نتیجه گیری: اعتبار نتیجه گیری با استفاده از روش ها و دستورالعمل های SLR به کار گرفته شده توسط محققان از نشریات معتبر، مانند [۱۹]، که نتایج را برای بازتولید گاه شماری تحقیقاتی این SLR با نتایج قابل اندازه گیری و یکسان ممکن می سازد انجام شد.

۲.۳. سوالات تحقیق

با توجه به انجام یک بررسی سیستماتیک ادبیات تحقیق، سوالات تحقیق نقش برجسته ای در تصمیم گیری استراتژی جستجو و تجزیه و تحلیل دارند. ما سوالات پژوهشی زیر (RQs) را برای این تحقیق شناسایی کردیم:

ادبیات تحقیق موجود در مورد مدیریت پروژه نرم افزاری با استفاده از تکنیک های یادگیری ماشین چه چیزی را نشان می دهد؟ آیا می توانیم مدل های مبتنی بر یادگیری ماشین بهتری را از نظر پیش بینی دقت با اعمال تبدیل ویژگی و انتخاب ویژگی برای کاهش احتمالات شکست پروژه به طور مؤثر بسازیم؟

شکاف های موجود برای چشم انداز تحقیق در زمینه مدیریت پروژه نرم افزاری چیست؟

معیارهای پیش بینی و سطح دقت فعلی آن ها که توسط تکنیک های تخمین مختلف اثبات می شود، چیست؟

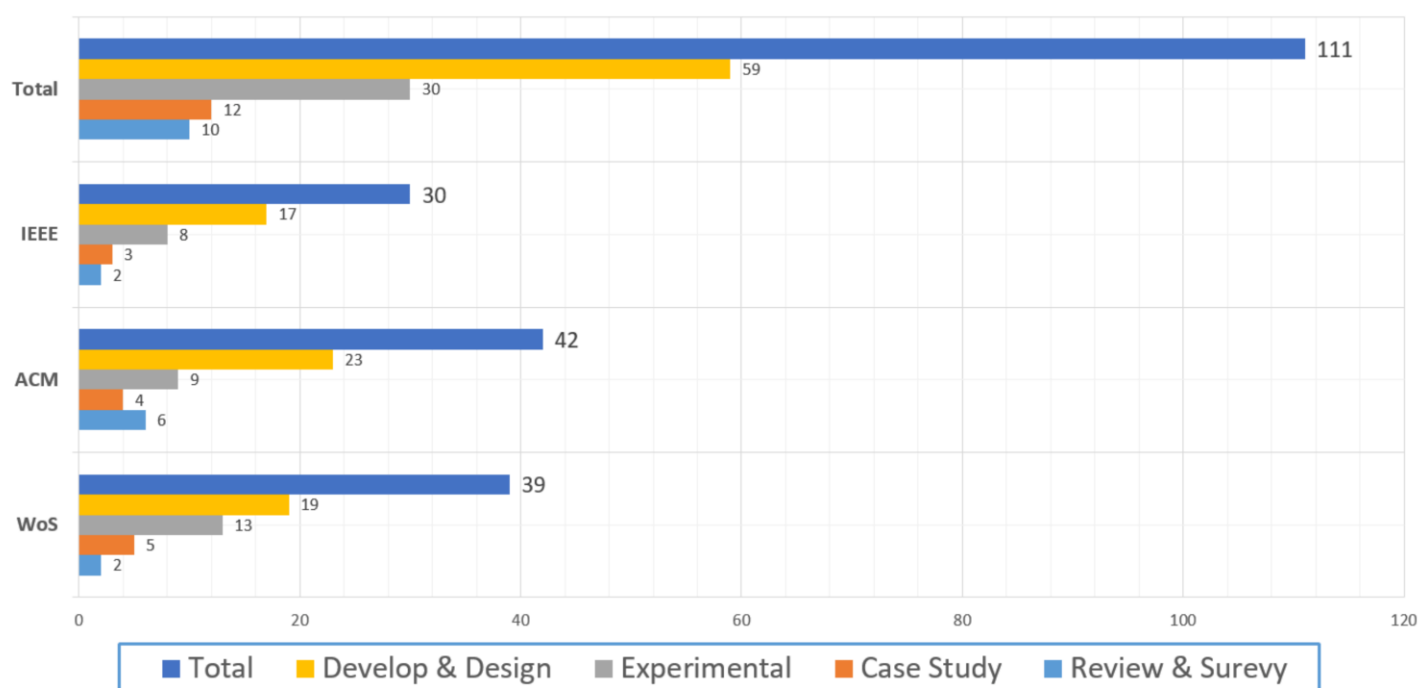
کدام الگوریتم یادگیری ماشین تمایل به دست کم گرفتن و کدام الگوریتم دست کم گرفتن دارد؟

۳.۳. اطلاعات آماری در گردآوری مقالات

نتیجه بررسی در قالب پاسخ به سوالات تحقیق مطرح می‌شود.

سوال اول تحقیق: ادبیات تحقیق موجود در مورد مدیریت پروژه نرم افزاری با استفاده از تکنیک‌های یادگیری ماشین چه چیزی را نشان می‌دهد؟

شکل ۳ طبقه بندی را نشان می‌دهد. رکوردها را می‌توان به چهار کلاس اساسی دسته بندی کرد. (۱) مرور و بررسی^۹، (۲) مطالعه موردی^{۱۰}، (۳) تجربی^{۱۱} و (۴) تحلیل و معماری^{۱۲}. دسته 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. دسته دوم اثرات، محرک‌ها، اقدامات متقابل و شرایط را مورد بحث قرار می‌دهد و فناوری‌هایی را برای بهبود کنترل اثربخشی پیشنهاد می‌کند. دسته سوم اثرات یک روش مورد استفاده برای طبقه بندی متغیرهای متعدد را ارائه می‌دهد که می‌تواند جنبه‌های مختلف روش یا محصول را در حین تولید تحت تأثیر قرار دهد. دسته چهارم شامل ساختارها، روش‌ها و تخصص برای یک مأموریت است.



شکل ۳. تعدادی مقاله در دسته بندی‌های مختلف بر اساس مجلات چاپی گنجانده شده است.

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

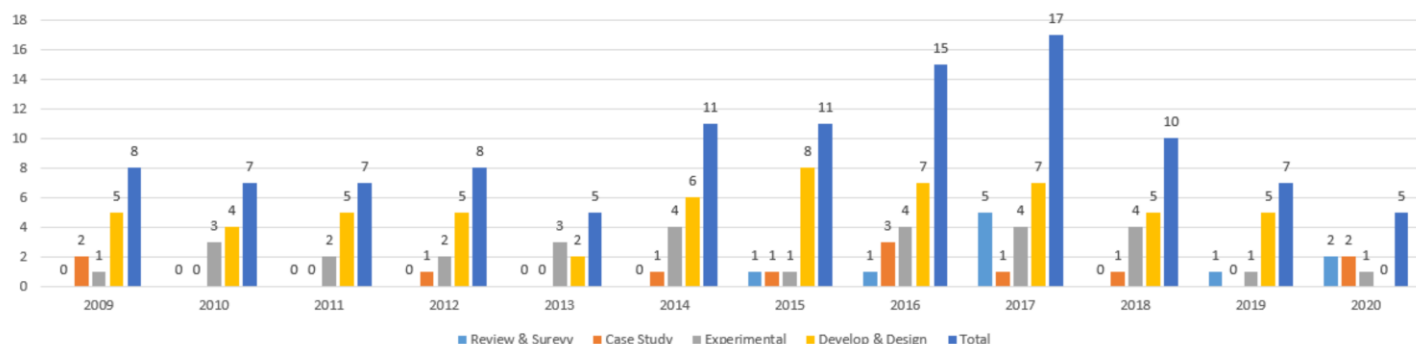
⁹ - Review and Survey

¹⁰ - Case Study

¹¹ - Experimental

¹² - Analytics and Architecture

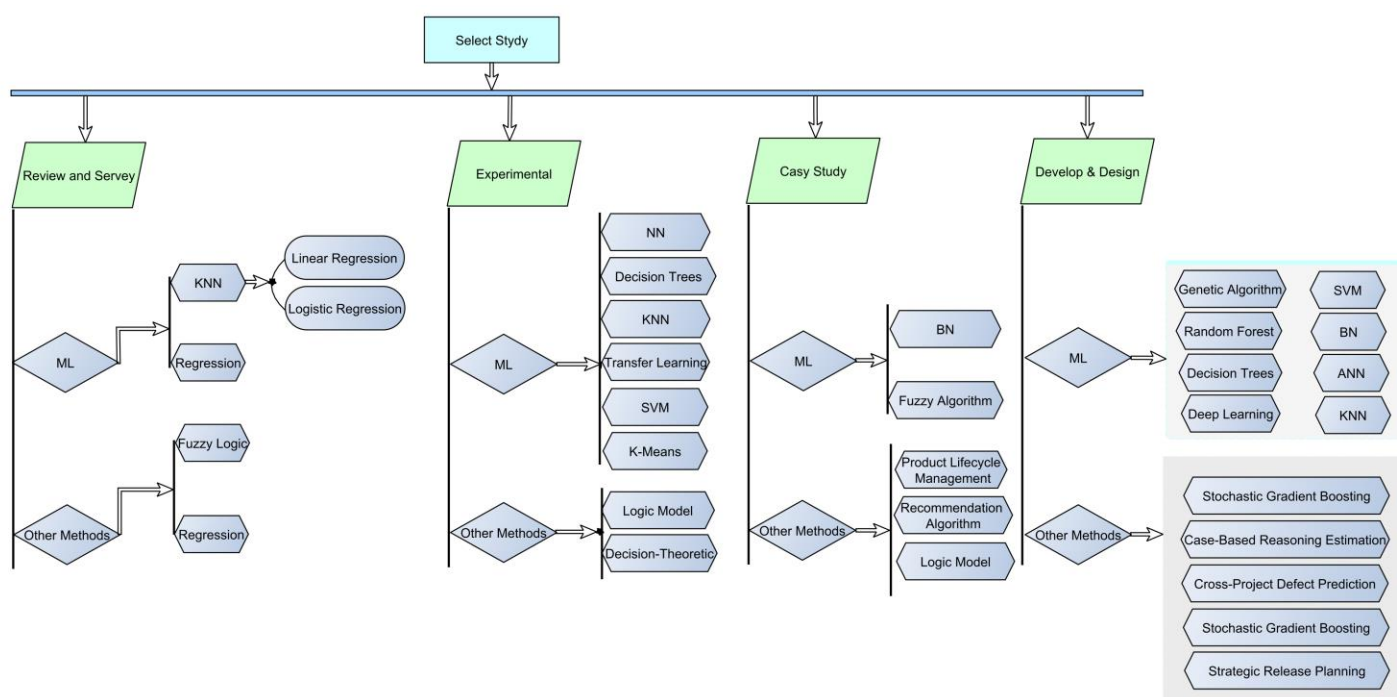
شکل ۴ تعداد انتشارات را بر اساس زمینه‌ها و منطقه‌ای که مطالعه و مطالعات در SPM در آن توسعه یافته است، نشان می‌دهد. یافته‌ها به ۱۰ مقاله از ۱۱۱ مقاله تقسیم شدند، ۱۲ مقاله از ۱۱۱ مقاله مربوط به روش‌های تحلیل موردی ML و استراتژی‌های SPM، و ۳۰ مقاله از ۱۱۱ مقاله، مراحل سیستماتیک و معیارهای آزمایشی برای بررسی 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 مطالعه ریاضی گروه‌های متعدد را نشان می‌دهد.



شکل ۴. مقالات منتشر شده بین سال‌های ۲۰۲۰ تا ۲۰۰۵.

شکل ۴، از طرف دیگر، شامل مقالاتی است که وابسته به سال انتشار هستند و مقالات طبقه‌بندی شده علمی را بین سال‌های ۲۰۲۰ تا ۲۰۰۹ نشان می‌دهد. در سال ۲۰۰۹، تنها هشت مقاله نوشته شده و ۴۹ مقاله از سال ۲۰۱۰ تا ۲۰۱۵ منتشر شده است. در مقایسه، برای سال‌های ۲۰۱۶، ۲۰۱۷، ۲۰۱۸، ۲۰۱۹ و ۲۰۲۰ به ترتیب ۱۵، ۱۶، ۱۱، ۷ و ۵ مقاله نوشته شده است. منابع اولیه‌ی تجزیه و تحلیل بر اساس تحقیقات ML-SPM گنجانده شده و دستورالعمل‌های کلی آن مورد ارزیابی قرار گرفت.

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 حوزه اصلی مشاهده شده است. همانطور که در شکل ۵ مشاهده می‌شود، ما متوجه برخی مضامین از ادبیات تحقیق شدیم و یک طبقه‌بندی ارائه کردیم. در حالی که فیلدهای خاصی با هم همپوشانی داشتند، چندین زیرمجموعه ایجاد کردیم.



شکل ۵. طبقه‌بندی ادبیات در مورد ارزیابی ریسک پروژه نرم‌افزاری با استفاده از تکنیک یادگیری ماشین.

۴.۳. مرور و بررسی مقالات

اسناد تجزیه و تحلیل و تحقیق، آخرین ادراک از فن‌آوری‌های ML را در تهیه و ارزیابی SPM و کاربرد الگوریتم‌های ML نشان می‌دهند.

۱.۴.۳. مطالعات انجام شده در مورد یادگیری ماشین و استفاده از آنها در SPM

این بخش، فرآیندهای ML را مورد بحث قرار می‌دهد و از آنها استفاده می‌کند. این 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 نشریه این گروه به ترتیب سه زیرمجموعه وجود داشت.

این خوشه فرعی بر روی الگوریتم K-Nearest Neighbor (KNN) انجام شد. در [۲۰]، مشاهدات، معیارها، مجموعه داده‌ها، معیارهای محاسباتی، چالش‌های ML، مدل‌های مختلف پیش‌بینی‌ها و مدل‌های مجموعه مورد استفاده در منطقه پیش‌بینی تعمیر و نگهداری مورد ارزیابی و تجزیه و تحلیل قرار گرفت. مقاله‌ی [۲۱] با KNN برای مدیریت مقادیر گم‌شده در ساختارهای داده مهندسی اطلاعات، نگرانی فزاینده‌ای را برای فناوری‌های ML نشان داده است.

مطالعات طبقه‌بندی دیگری بر روی رگرسیون انجام شد. مقاله [۱۹] روش‌ها و پیش‌بینی‌کننده‌های پیش‌بینی نوسانات و معیارهای طبقه‌بندی را شناسایی کرد. ویژگی‌هایی که به عنوان شاخص‌های پارامترهای نوسانات ادبیات و تکنیک‌های پیش‌بینی مورد استفاده برای افزایش دقت نوسانات الزامات پیش‌بینی مورد استفاده قرار گرفتند، ایجاد شده‌اند. مشخصات با نوسان برای برنامه‌های نرم‌افزاری حیاتی هستند، زیرا مستقیماً منجر به هزینه‌ها و دوره بیش از حد می‌شوند. در [22]، SLR برای کمک به مکانیسم رسمی یافته‌های تکرارپذیر پیشنهاد شد. مطالعه نمی‌تواند کاربرد دقیق را توسط سازماندهی یک مجموعه داده، مانند سایر مجموعه‌های داده، حل کند. مقاله‌ی [۱۸] به استفاده از روش‌های ML برای محاسبه تلاش برنامه پرداخته است. مطالعه سیستمیک نشان داد که رویکردهای ML، مقیاس‌های اندازه، مجموعه داده‌های مقایسه‌ای، روش‌های ارزیابی و غیره تأثیرگذار بودند.

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

۲.۴.۳. سایر روش‌ها

این بخش به بحث و استفاده از سایر رویکردها می‌پردازد. این مقالات در موضوعات مختلف و پیاده‌سازی گروه‌بندی شده‌اند. آثار فهرست شده در یک گروه مدل منطقی خاص گروه‌بندی می‌شوند.

نویسنده [۲۴] از ارزیابی بازده 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. مقاله [۱۷] تحقیقات اخیر تعمیر و نگهداری برنامه را به طور گسترده تجزیه و تحلیل می‌کند. نتایج مطالعه نشان داد که استفاده در پیش‌بینی تعمیر و نگهداری از الگوریتم‌های ML از سال ۲۰۰۵ افزایش یافته است. مشکلات بر

اساس حوزه‌های تخصصی بدنه دانش مدیریت پروژه (PMBOC) طبقه‌بندی شدند و آنها نیز مورد تجزیه و تحلیل قرار گرفتند [۲۵،۲۶]. مشکلات با استفاده از ابزارهای هوش مصنوعی و موانع در چابک PM مورد بررسی قرار گرفت. این کمک به نیاز مورد انتظار برای توسعه مدل‌های PM مدرن و تکنیک‌های IT مربوط می‌شود که روش‌های مبتنی بر ML را با درمان عدم دقت، ابهام یا ابهام با شاخص‌های عملکرد حیاتی که با حوزه‌های تخصصی مرتبط هستند، ادغام می‌کند.

۵.۳. مطالعات تجربی

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

سوال دوم تحقیق: آیا می‌توانیم مدل‌های مبتنی بر یادگیری ماشین بهتری را از نظر پیش‌بینی دقت با اعمال تبدیل ویژگی و انتخاب ویژگی برای کاهش احتمالات شکست پروژه به طور مؤثر بسازیم؟

۱.۵.۳. مطالعات انجام شده بر روی روش‌های یادگیری ماشین

بر اساس رویکردهای روش شناسی تولید ML، مقالات انتخاب شده در گروه‌های وسیعی دسته‌بندی شدند. ۲۷ نشریه در هفت زیرمجموعه در گروه زیر سازماندهی شدند. این بخش شامل ۱۵ مقاله است که برای SPM با الگوریتم‌های مختلف ML استفاده می‌شود. جدول ۱ ML، تعریف، دامنه، و دیگر جنبه‌های اصلی تاسیس تجربی را در تضاد قرار می‌دهد.

| محدودیت سیستم جدید | محدودیت سیستم قدیمی | استخراج ویژگی | دامنه | شرح | نوع ML | مرجع |
|---|---|-------------------------------------|---|--|--------------|--------|
| داده کاوی را بر روی مدل دقیق تری از کدهای نظارتی HIPAA اعمال کرده که حقوق خاصی را نشان می‌دهد. | موفقیت محدود برای ردیابی کدهای نظارتی به دلیل تفاوت در اصطلاحات که می‌تواند بین کدها و الزامات سطح محصول وجود داشته باشد. | - | امنیت و حریم خصوصی در حوزه مراقبت‌های بهداشتی | دو رویکرد ML را برای تقویت هماهنگی بین کدهای نظارتی و مشخصات در سطح کالا ارزیابی کرد. | SVM | [27] |
| تخصص در طراحی و پیاده‌سازی ابزارهای پردازش داده مقیاس پذیر و ابزارهای یادگیری | تجزیه و تحلیل ماشین کامل، تجزیه و تحلیل نرم افزار، ML، پردازش داده‌ها و تجسم دانش | - | طیف گسترده‌ای از تجربه و آگاهی میدانی | استدلال کرد که تجزیه و تحلیل اطلاعات از فناوری های محاسباتی استفاده می‌کند | چندین نوع ML | [28] |
| در دسترس بودن داده های دانه‌ای در مورد ویژگی های پروژه و محصول | برنامه ریزی و شاخص های تاریخی کالا بسته به روش یادگیری | - | برآورد تلاش و مدت | ارزیابی ماشین را توسعه دهید، استفاده از سرمایه را به حداکثر برسانید | چندین نوع ML | [29] |
| توسعه داده شده برای تولید داده‌ها در مورد مشارکت فردی و تیمی، که می‌تواند برای مدیریت مفید باشد | ایجاد مفهوم آنباشر حدود یک دهه پیش | - | ارزیابی پروژه، سرعت تیم و برآورد زمان | راه حل جدیدی را برای رسیدگی به این معضل فراگیر از طریق ترکیب مدرن دیجیتال کردن و ML نشان می‌دهد. | چندین نوع ML | [30] |
| مجموعه داده‌ها و توابع بزرگتر در این آزمایش شامل نمی‌شود | هیچ چارچوبی برای رشد چابک وجود ندارد که مناسب ترین باشد | استخراج توکن | تخمین تلاش توسعه نرم افزار | تکمیل پوکر برنامه ریزی دستی Agile | چندین نوع ML | [31] |
| هدف ارزیابی تأثیر تعداد شرکت کنندگان گروه بود | در هر صورت کافی دیده شده است | ارقام مجموعه داده شامل تعداد سرمایه | تخمین تلاش نرم افزاری | بسیاری از استراتژی‌های انفرادی برای پیش‌بینی تلاش | چندین نوع ML | [32] |

| محدودیت سیستم جدید | محدودیت سیستم قدیمی | استخراج ویژگی | دامنه | شرح | نوع ML | مرجع |
|--|--|------------------------------------|--|---|-----------------------------------|------|
| | | گذاری ها و تعداد ویژگی ها می باشد | | توسعه نرم افزار سیستم پیشنهاد شد | | |
| نتایج این تحقیق به نفع چارچوب بلادرنگ است و بینشی در مورد کارایی، دقت و سطح قابلیت اطمینان ارائه می کند. | در اصل برای مقابله با مشکل نرم افزار ساخته شده است | CollabCrew ETL | مدیریت پروژه نرم افزاری | هدف دستیابی به یک راه حل با پیاده سازی یک دستگاه هوشمند بود که اعضای تیم را خلاقانه به یک ماموریت خاص اختصاص می دهد | چندین نوع ML [33] | |
| طبقه بندی کننده دیگری برای ساختمان مدل به غیر از NB یا SVM پیدا کنید | استراتژی داده Iter به طور قابل توجهی کارایی پیش بینی نقص بین پروژه را بهبود می بخشد و روش انتخابی سلسله مراتبی پیشنهاد شده به طور قابل توجهی عملکرد را بهبود می بخشد. | رویکردهای مبتنی بر ویژگی | پیش بینی نقص بین پروژه ها | مقایسه گسترده ای از داده های شناخته شده ارائه شده است | NB و SVM [34] | |
| به نتایج بهبود یافته با نگرانی کمتر در مورد طرح CI واقعی می رسد، که نشان می دهد که ACONA می تواند به طور چشمگیری هزینه های CI را بیش از روش های فعلی به حداقل برساند. | استفاده از طبقه بندی های آموزش دیده برای ارائه پیش بینی های خوب برای پروژه فعلی با جریان داده ها در مورد داده های تاریخی گسترده از پروژه های دیگر | - | مدیریت فرآیند توسعه نرم افزار؛ مدیریت ریسک | یک راه حل مدل سازگار آنلاین فعال به ACONA ارائه دهید، که مجموعه ای از دسته ها را به صورت پویا با پروژه های مختلف تطبیق می دهد. | چندین نوع ML [35] | |
| ترکیب سایر مدل های ML مانند Treeboost و XBoost و غیره و اعتبارسنجی با مجموعه داده های متنوع دیگر | برآورد دقیق تلاش پروژه نرم افزاری | ویژگی های غیر خطی | تلاش پروژه نرم افزاری | هدف از پیش بینی تلاش | RF، پرسپترون چندلایه و SVM [36] | |
| مدل هایی در حوزه ی رشد واقعی | مدل های دقیق و مفید ساخته شده در واقع محدود هستند، حتی اگر به ذینفعان فناوری مزایای مالی قابل توجهی بدهند. | ویژگی های زیر مجموعه از ISBSG | برآورد هزینه نرم افزار | در موارد خاص، ارقام قابل اعتمادی را ارائه می دهد | FL، DT [37] | |
| از طبقه بندی کننده های قوی تری برای بهبود دقت پیش بینی ریسک پروژه نرم افزاری برون سپاری شده استفاده می کند | مدل های موجود عمدتاً بر این فرض متمرکز شده اند که تمام هزینه های طبقه بندی اشتباه معادل هستند، که با این واقعیت که پیش بینی ریسک در منطقه پروژه نرم افزاری وجود دارد، ارتباطی ندارد. | 25 عامل خطر را انتخاب کرد | پروژه نرم افزاری | پروژه توسعه خارجی یکی از رویکردهای کلیدی برای ساخت نرم افزاری است که میزان شکست بالایی دارد. مدل هوشمند پیش بینی ریسک می تواند به زمان بندی پروژه های پرخطر کمک کند | SVM [38] | |
| با در نظر گرفتن یک مطالعه شبیه سازی دقیق تر با استفاده از انواع بسیار متعادل تر مجموعه داده های مورد نیاز برای درک شایستگی STOCHS، به ویژه مجموعه داده های بزرگ تر، گسترش یافته است. | راه حل هایی برای مشکل حوزه های پر سر و صدا در پیش بینی تلاش نرم افزار از دیدگاه احتمالی | ویژگی به طور تصادفی انتخاب شده است | پیش بینی تلاش نرم افزاری | تأثیر دامنه های پر سر و صدا بر هشت دقت ML و الگوریتم های تشخیص روندهای آماری را بررسی می کند. | SVM [39] | |
| بررسی و مقایسه با سایر روش های داده کاوی | برای ارتقای تصمیمات نرم افزار PM با داده کاوی و تولید نتایج عملی | انتخاب زیر مجموعه ویژگی | توسعه جهانی نرم افزار | از یک استراتژی طراحی مهندسی اطلاعات خاص برای شناسایی نرم افزار معیوب استفاده کرد | K-Means [40] | |
| با استفاده از سایر الگوریتم های ML و اعتبارسنجی با سایر مجموعه های داده متنوع تقویت می شود | با توجه به مقایسه الگوریتم های ML برای تخمین تلاش در نرم افزار با اندازه های مختلف | - | تخمین تلاش نرم افزاری | تخمین تلاش نرم افزار حیاتی ترین کار در مهندسی نرم افزار و PM است | DT [41] | |

| محدودیت سیستم جدید | محدودیت سیستم قدیمی | استخراج ویژگی | دامنه | شرح | نوع ML | مرجع |
|--|---|--|-------------------------------|---|----------------|--------|
| اهمیت ویژگی‌ها را برای افزایش عملکرد طبقه بندی کنندگان و دقت مدل SFP مطالعه کرد | بهبود عملکرد همه طبقه بندی کنندگان پس از حل مشکلات نامتعادل | فرکانس انتخاب هر ویژگی از همه مجموعه داده‌ها با استفاده از EBMFOV3 | پیش بینی خطای نرم افزار (SFP) | رویکرد هوشمند برای پیش‌بینی خطای نرم‌افزار بر اساس بهینه‌سازی شعله پروانه دودویی با نمونه‌گیری مصنوعی تطبیقی معرفی شد. | DT، kNN و LDA | [42] |
| استفاده از یک شبکه عصبی با ویژگی پایه شعاعی برای پیش‌بینی کارایی متخصصان و تیم‌های توسعه دهنده | توسعه دهندگان و مدیران به خروجی متخصصان فناوری اشاره می‌کنند که معمولاً به عنوان نسبت اندازه به زمان محاسبه می‌شود. | - | متخصصین نرم افزار | به عنوان رگرسیون عمومی شبکه عصبی برای پیش‌بینی کارایی در عملکرد برنامه‌ها نامگذاری شد. | شبکه عصبی | [43] |
| پیش‌بینی با توجه به جزئیات پرونده فعلی / گذشته عمل می‌کند و کار بالقوه و طول پرونده را تخمین می‌زند | الگوریتم‌ها را با توجه به ضریب همبستگی آنها ارزیابی کرد | - | SPM | چندین الگوریتم ML برای پیش‌بینی مدت زمان نرم‌افزار | ANN، SVM | [44] |
| اثربخشی فراابتکاری در ارزیابی سایر شاخص‌های نرم‌افزار اولیه، ایجاد داده در نرم‌افزارهای خصوصی و عمومی | استفاده از HEAD-DT برای ایجاد یک الگوریتم مبتنی بر قراردادهای قضاوت که با برنامه نگهداری داده‌ها سازگار است. | - | پیش‌بینی تلاش نرم‌افزاری | تصمیمات در حال تکامل پیشنهادی از طریق یک الگوریتم تکاملی و درخت مربوطه برای پیش‌بینی تلاش تعمیر و نگهداری دستگاه | درخت تصمیم | [45] |
| داده‌های ناقص و برآورد ماشینی تحلیل نظری و مشاهده‌ای | آغاز درک بهتر و استفاده از نهادهای تصمیم‌گیرنده به عنوان طبقه بندی بخشی از روش‌های انتساب مجموعه | ویژگی چهار بعدی | پیش‌بینی نرم افزار | ابزاری برای تقویت عملکرد پیش‌بینی تلاش برنامه پیشنهاد شده است | درخت تصمیم | [46] |
| بررسی ML و مجموعه داده‌های اضافی؛ سایر اشکال اندازه عمل، از جمله موارد غیر پارامتریک؛ و اندازه پنجره‌های اضافی برای ارزیابی یادگیری آنلاین | آزمایش‌های سیستمیک بر روی سه مجموعه داده با پنج ML در تنظیمات پارامترهای چندانگانه انجام شد | - | تخمین تلاش نرم‌افزاری | برای کشف اینکه چگونه پارامترها با پارامترهای خود تطبیق بیشتری دارند و هر چند وقت یکبار ممکن است خروجی MLs در SEE تحت تأثیر قرار گیرد. | k-NN | [47] |
| حساسیت Dycom به مقادیر پارامتر، مردمک ساده، ورودی‌ها و جدا کردن CC به بخش‌های جداگانه | این سیستم نه تنها از دانش WC بسیار کمتری نسبت به یک مدل WC قابل مقایسه استفاده می‌کند، بلکه خروجی معادل/بهرتر تولید می‌کند. | تعداد سرمایه‌گذاری با هر مشخصه | تخمین تلاش نرم‌افزاری | هدف جزئیات محاسبه تلاش ماشینی بین شرکتی (CC) استفاده صریح از دانش یا مدل‌های CC برای پیش‌بینی داده‌های CC یا داده‌های مدل در موقعیت‌های WC است. | درختان رگرسیون | [48] |
| با استفاده از ارزیابی اعتبار خودکار RVM، سه مورد منحصر به فرد ایجاد شد و توصیه‌هایی در مورد اینکه آیا تلاش مورد نیاز پیشنهاد شده است یا خیر. | نشان داده است که RVM یک شاخص برجسته SEE است و نیاز به تجزیه و تحلیل و استفاده بیشتری دارد | ویژگی‌های خاص حساب SEE | تخمین تلاش نرم‌افزاری | مطالعات سیستماتیک نشان می‌دهد که RVM در مقایسه با رویکردهای پیشرفته SEE بسیار موفق است | SVM | [49] |
| پیش‌بینی‌ها را می‌توان با در نظر گرفتن ویژگی‌های جدید مرتبط با ویژگی‌های توسعه انسانی افزایش داد | ویژگی‌های توسعه برای ساخت مدل‌های آماری که نقاط داستان را برای پرونده‌های منبع باز تحلیل می‌کنند، استفاده شده‌اند | ویژگی‌های محاسبه شده بر روی معیارهای وظیفه طبقه بندی وابسته به ویژگی‌های اولیه است | برآورد تلاش | محاسبه درست تلاش کمک می‌کند تا مشخص شود کدام چالش‌ها باید در دور بعدی اصلاح یا حل شوند | SVM | [50] |
| در بسط این مطالعه، گزینه‌های دیگری برای تابع هسته در LS-SVM به غیر از تابع پایه شعاعی وجود دارد. | به عنوان یک روش پایه برای تخمین تلاش توسعه نرم‌افزار در نظر گرفته شده است | غیر عادی بودن و ویژگی‌های طبقه بندی مجموعه داده‌های مختلف | تخمین تلاش توسعه نرم افزار | روش‌های کالیبراسیون به جز تنظیم غیر خطی مبتنی بر ANN بستگی دارد | ANN | [51] |

| محدودیت سیستم جدید | محدودیت سیستم قدیمی | استخراج ویژگی | دامنه | شرح | نوع ML | مرجع |
|---|---|--|-----------------------|---|---------|------|
| فرآیندهای خوشه بندی، یادگیرندگان ساده، ویژگی های پروژه ورودی، توابع پروژه خوشه بندی، مقادیر پارامترها | خوشه بندی Dycom با-K Means به جداسازی برنامه های CC کمک می کند و کارایی پیش بینی خوب یا بهتری نسبت به Dycom ایجاد می کند. | برای توصیف پروژه های آموزشی برای خوشه بندی می توان از ویژگی های متفاوتی استفاده کرد. | تخمین تلاش نرم افزاری | رویکردهای خوشه بندی تعمیم داده می شوند تا برای ساخت زیر مجموعه های CC مورد استفاده قرار گیرند. سه روش مجزا برای خوشه بندی مورد بررسی قرار گرفته است | K-Means | [52] |

۲.۵.۳. مطالعات انجام شده بر روی سایر روش ها

این بخش بحث می کند و سیستم ML را گسترش نمی دهد. آثار انتخاب شده به گروه هایی از جمله مدل ها یا تکنیک ها تقسیم می شوند. این مقالات در گروه ها دستورالعمل های مختلفی را با پارامترهای موجود در ارزیابی با توجه به تجزیه و تحلیل خود شناسایی می کنند.

مدل اول، حالت منطقی. تأکید [۵۳] بر افزایش ویژگی های کیفیت، مانند خطاها، ماه ها و تنش بود. طرفداران مدل پارامتری ادعا می کنند که مدل های مستقل از دامنه ممکن است با داده های محلی تطبیق داده شوند. نویسندگان [۵۴] ادغام مرجع + تجسم را در پیشرفت های پروژه توصیه کردند.

مدل دوم، مدل پارامتریک. نویسندگان [۵۵] مزایای روش های محاسبه نادرست هزینه را هنگام توسعه الگوهایی برای خرابی های نرم افزار پیش بینی کننده که از اطلاعات پروژه مخزن متقابل استفاده می کنند، بررسی کردند. در این شرایط، متوجه می شوید که آموزش های حساس به هزینه دارای نکاتی نیست که بیشتر از طبقه بندی کننده های حساس به هزینه باشد.

مدل سوم، تکنیک های بهینه سازی نظری تصمیم. در [۵۶]، تکنیک های بهینه سازی نظری تصمیم ارائه شد که می تواند بهترین پارامترها را برای طیف وسیعی از گردش های کاری انتخاب کند. آزمایش های اولیه نشان می دهد که گردش های کاری بهینه شده به طور قابل توجهی مقرون به صرفه تر از پارامترهای تنظیم شده دستی هستند. آنها استدلال می کنند که روش های هوش مصنوعی (AI)، مانند ML، تئوری تصمیم گیری و بهینه سازی می توانند این مشکلات را حل کنند و ساخت سریع جریان های کاری مؤثر با منبع جمعی را تسهیل کنند.

۳.۶. مطالعه موردی

این بخش سوم، یک پروژه، کمپین یا شرکتی را تجزیه و تحلیل می کند که موقعیتی را شناسایی می کند، راه حل های پیشنهادی، اقدامات اجرایی را شناسایی می کند و عواملی را که با استفاده از تکنیک ML به شکست یا موفقیت در تکنیک های توسعه SPM کمک کرده اند، شناسایی می کند.

مقالات این بخش بر روی روش های ML تمرکز دارند و آثار منتخب بسته به روش های ML در تکنیک های توسعه SPM در دسته های گسترده طبقه بندی شدند. هفت مقاله این دسته به سه زیرمجموعه تقسیم شدند.

این بخش شامل پنج مقاله ML است که از بسیاری از الگوریتم های SPM استفاده می کنند. بر اساس رویکردهای ML روش شناسی توسعه نرم افزار، کار به گروه های گسترده تقسیم می شود. مقالات [۵۷، ۵۸] بر بهبود قابلیت پیش بینی برآورد و تخصیص تلاش مورد نیاز برای تطبیق با مشتری، مدیریت پروژه و مسائل مختلف توسعه تمرکز داشتند. از طریق راه حل، نیاز به پرداختن به این مسائل مربوط به پروتکل های گزارش دهی 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 همه موضوعات منعکس می‌کند. علاوه بر این، تکنیک‌های محاسباتی و رگرسیون پیش‌بینی‌کننده نرم‌افزار در مقابل هم قرار گرفتند.

مقالات این دسته در مورد الگوریتم شبکه‌های بیزی مطالعه می‌کنند. در [۶۱]، راه حلی برای تخمین ارزش با استفاده از ترکیبی از راه حل‌های کیفی و ML ارائه شده است که در آن یک مدل احتمالی شامل دانش سهامداران مختلف ارائه می‌شود. برای پیش‌بینی ارزش کلی یک تصمیم معین مربوط به مدیریت و توسعه محصول استفاده می‌شود. نویسندگان [۶۲] مدلی را پیاده‌سازی کردند که به طور خودکار رابطه بین عوامل خطر و کاهش را از طریق یک سیستم پشتیبانی تصمیم‌گیری هوشمند (DSS) شناسایی می‌کند. روش پیشنهادی محدودیت‌های مدیریت ریسک فعلی را که به طور گسترده به آن اشاره شده است، مانند فقدان DSS یکنواخت و ارتباط بین ریسک‌های نرم‌افزار و کاهش، پوشش می‌دهد.

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

مطالعات انجام شده بر 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.

اولین مدل، که در مدیریت چرخه عمر محصول (PLM) انجام شد. نویسندگان PLM [65] را ارائه کردند. این روش لایه‌ای از عملکرد ایجاد می‌کند تا امکان تکرار بعدی PLM در اطراف یک شبکه PLM ایجاد شده را فراهم کند. سپس این PLM جدید با استفاده از مطالعه موردی Ford Powertrain در یک اکوسیستم اتوماسیون کارخانه دیجیتال ادغام خواهد شد.

مدل دوم، بر اساس یک الگوریتم توصیه انجام شده است. یک الگوریتم نرم‌افزاری جدید توسط [۶۶] پیشنهاد شده است. ابتدا یک ویژگی مبتنی بر باگ و یک مکانیسم غربالگری خاص برای تأیید اعتبار رفع‌کننده متقاضی اضافه کنید، شبکه‌ای از تعهدات چند برنامه‌نویس را با گرفتن طیفی از نظرات و قول‌ها ایجاد کنید، آنها را در رتبه‌بندی قرار دهید و سپس مناسب‌ترین رفع‌کننده اشکال را تعیین کنید. نتیجه نشان می‌دهد که راه حل با موفقیت تابع تریاژ خطا را اجرا می‌کند.

مدل سوم، مدل منطقی. دو مقاله [۶۷،۶۸] از یک مطالعه موردی واقعی در دامنه توزیع شده استفاده کرده و آزمایش چابک را برای یک تیم منتخب اعمال کرد و نتیجه آنها را با سه گروه دیگر مقایسه کرد تا تأثیر مشارکت مشتری در فرآیند آزمایش برای غلبه بر چالش‌های توسعه‌ی توزیع شده را تعیین کند. با این حال، گروهی که از تست چابک استفاده می‌کنند، بیش از ۹۹٪ از تمام درخواست‌های وارد شده به فرآیند آزمایش را تأیید کردند، که تفاوت قابل توجهی از بهره‌وری هر پروژه توسعه پشتیبانی می‌کند.

پنجاه و نه مقاله؛ مطالعه 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.

۱.۷.۳. مطالعات انجام شده بر روی روش‌های یادگیری ماشین

آثار برگزیده بسته به روش‌های ML در SPM به دسته‌های گسترده طبقه‌بندی شدند. چهل و سه مقاله در این دسته به ۹ زیر شاخه تقسیم شدند.

این بخش شامل پانزده مقاله فرآیند ML است که از الگوریتم‌های مختلف SPM استفاده می‌کنند. بر اساس روش ML در تولید نرم افزار، آثار منتخب در دسته‌های بزرگ دسته بندی می‌شوند.

کاربردهای حوزه SPM توسط مقالات دسته‌ی اول [۶۹،۷۰] مورد ارزیابی قرار گرفتند: رفتارها به عنوان کار و به طور مناسبی، طبقه بندی می‌شوند. این مورد یک مشکل یادگیری چند هدفی را در طراحی مدل برای تخمین تلاش سیستم نشان داد. همچنین این مورد به درک مصالحه بین معیارهای مختلف عملکرد با ایجاد مدل‌های SEE که به طور 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.

دو مقاله [۷۴،۷۵] روش خودکار مبتنی بر ML برای تخمین تلاش نرم افزار بر اساس متن کار را معرفی کردند. یک ANN که برای ساده کردن توابع تخمین تلاش استفاده می‌شود. ارزیابی نرم افزار 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 بر خلاف روش‌های مختلف دیگر، تقریباً همیشه در دسترس هستند.

در [۷۶]، نویسندگان نتیجه‌ی بازخورد را در مورد معیارهای اجتماعی کار داده کاوی کاربردی، تخمین تلاش، تولید مورد آزمایشی و غیره نشان دادند. سپس نتایج آن تحلیل غیررسمی در هفت اصل و ده‌ها نکته دیگر رسمیت یافت و نظام‌مند شد. هدف، تشریح رویکردهایی برای نتایج موفقیت آمیز داده کاوی صنعتی است، اما اضافه کنم که برخی از این اصول برای داده کاوی دانشگاهی صادق باشند.

تحقیق در مورد [۷۷-۷۹] یک مدل هیبریدی جدید را به درستی ایجاد کرده است. این مدل حتی برای طیف وسیع تری از فعالیت‌ها ایده آل است، زیرا در یک پایگاه داده قابل استفاده است. دو الگوریتم ML، ANN و SVM، عملکرد مدل ما را بررسی می‌کنند. آزمایش‌ها، نسخه قوی‌تری نسبت به مدل پیش‌بینی خطر SVM ما را نشان می‌دهند.

بقیه‌ی 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 سرمایه گذاری برق اعمال می‌شود. اختصاص پروژه و عملکرد روش برنامه ریزی شده بر اساس پارامترهایی مانند عدد صحیح، میانگین اشتباه مطلق، منبع و اشتباه مطلق نسبی محاسبه و تجزیه و تحلیل می‌شود.

نویسندگان [۸۳-۸۵] چارچوب آزمایشی از معیارهای کد منبع ارائه کرد و بهترین مجموعه متریک را برای عملکرد مدل انتخاب کرد. روش برآورد هزینه برای آزمایش مدل‌های شکست پیش‌بینی‌کننده استفاده می‌شود. هدف همچنین رفع این محدودیت‌ها با

بستن فاصله بین نتایج آزمایش تجدیدنظر شده و اجرای بالقوه در فعالیت ML های کارآمد در چرخه عمر طرح اولیه رشد پروژه بود.

سایر مقالات [۸۶-۸۸] پیشنهاد کردند که سیستم انتخاب مدل های نرم افزاری توسط پروژه مورد استفاده قرار می گیرد و با استفاده از اثبات مهندسی نرم افزار تاریخی، مدل فرآیند نرم افزاری را انتخاب می کند که برای پروژه فعلی در مراحل اولیه توسعه مناسب است. نویسندگان [۸۹] معماری ارائه می کنند که مدل های تجزیه و تحلیل خودکار شکست را با استفاده از الگوریتم های طبقه بندی ML برای آزمایش نتایج حاصل از تکنیک های مختلف برای فایرفاکس و Netbeans توسعه می دهد. آنها نشان می دهند که چگونه مدل های پیش بینی خودکار در تقریب این دو پارامتر به طور واقعی تر از انواع خطوط پایه برای حالت ها و پروژه های تلفات خاص کارآمدتر هستند.

این بخش مقالاتی را توصیف می کند که از شبکه های عصبی مصنوعی (ANN) استفاده کرده اند. دو مقاله [۹۰،۹۱] در مورد ایجاد محرک خطر سیستم های ML، مناسب ترین مشوق های ریسک را در مورد الزامات، سناریوها و برجسب های طبقه بندی برای ایجاد یک پروژه نرم افزاری ارائه می کنند. مطالعه باید مستقل از هر یک از این طبقه بندی ها مشاهده شود، زیرا طبقه بندی ها مستقل از علل خطر هستند. نویسندگان [۹۲،۹۳]، در مورد کاربرد یک برنامه های ANN محور و بهینه شده تیم توسعه نرم افزار برای تشخیص شکاف های ظرفیت و آماده سازی برنامه ریزی و زمان بندی مهارت های SPM، تسهیل پیش بینی و پیش بینی استعدادها بر اساس فناوری های هوشمند کاربردی، بیان منابع و تکنیک های توسعه کارکنان. یکی از اهداف کلیدی [۹۴] کمک به پیش بینی SCE با استفاده از فرآیند یادگیری ANN فعلی است. اثرات میانگین ریشه و بزرگی متناسب متوسط خطا هستند.

دسته بندی دیگری از مقاله ها ماشین بردار پشتیبان (SVM) را مطالعه می کند. دو مقاله، کار سیستم های Gating پروژه و استفاده از آنها برای تمیز دادن خط ساختمان را توضیح می دهند. دسته اولیه برای سیستم های CI [95, 96]. سه اکتشاف برای رسیدگی به ارسال ها برای سیستم Gating پیشنهاد و بررسی می شوند. نشان داده می شود که این نتیجه از استفاده از غربالگری با نرخ موفقیت بالا و نظارت مستمر در سطح پایین است. این یک نتیجه قوی است. سومین و آخرین ارزیابی اکتشافی اهرم ML را برای بهینه سازی انتخاب آزمایش می کند. دیگر مقالات [۹۷،۹۸،۹۹،۱۰۰]، مدل ارزیابی ریسک پروژه مربع کمتر (LS-SVM) را توسعه داده اند. شبیه سازی نشان می دهد که نتیجه پیش بینی شده SVM موفقیت آمیز است. برای تحلیل مدل ارزیابی ریسک پروژه از رویکرد LS-SVM استفاده شد. داده های ارزیابی ریسک متخصص برای آموزش مدل رگرسیون LS-SVM برای ارتباط نقشه برداری بین خطر و ویژگی ها استفاده می شود. همچنین یافته ها دقت و تعمیم قوی مدل LS-SVM را نشان می دهند. آخرین مقاله [۱۰۱] پیشنهاد کرد که مراحل باید با روش های یادگیری SVM در زمان اجرا تجزیه و تحلیل و درجه بندی شوند. مکانیسمی را تعریف کرده است که با انتخاب معیارها از برنامه موجود یا تغییر جهت نرم افزار اندازه گیری ها، تقویت برنامه های محاسباتی با معیارهای تطبیق پذیری مشخص می شود.

مقالات این بخش، جنگل تصادفی را توصیف می کنند. دو مقاله [۱۰۲،۱۰۳] یک مدل پیش بینی بسیار قابل اعتماد ایجاد کرده اند. در مرحله ایجاد نرم افزار در حال انجام و ابتکار تحقیق، رویکرد ذکر شده در یک مجله عملی باید شامل پیش بینی نقص باشد. نتایج تدریس را با دقیق ترین پیش بینی کننده خطا در بخش در طبقه بندی های متعدد، از جمله NB، DT، یا RF مقایسه و دوباره بررسی کنید. در مقاله [۱۰۴] نتایج ارزیابی فعالیت تیمی کامل، که شامل بیش از ۴۰ مرحله عینی و قابل مشاهده است که توسط گروه های دانش آموزی که در طرح های کلاسی همکاری می کنند، انجام شده است. همچنین، چارچوب ML از الگوریتم RF برای پیش بینی رفتار هم تیمی ها و نتایج تیم استفاده می کند.

فقط یک مقاله در شبکه بیزی انجام شد. مقاله‌ی [۱۰۵] از پایگاه‌های اطلاعاتی متعددی برای جمع‌آوری معیارهایی استفاده کرد که از مشخصات طراحی برای سه برنامه جداگانه ناسا، که برای ابزار فضاپیما، یک چارچوب پیش‌بینی زمین در زمان واقعی و برنامه‌های کاربردی ماهواره‌های پرواز ساخته شده‌اند، گرفته شده‌اند. کاربرد BN در مهندسی نیازمندی‌ها را بررسی کنید و به طور خاص بر شناسایی و ارزیابی الزامات پرخطر تمرکز کنید.

دو مقاله در مورد K-Nearest Neighbor (k-NN) انجام شده است. مدل پیشنهادی نشان داده شده در مقاله‌ی [۱۰۶] به مدیران پروژه مکان‌های مختلفی 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 مقاله‌ی [۱۰۷] وجود دارد که ترکیبی از بهینه‌سازی الگوریتم‌های COA-Cuckoo و KNN است. یافته‌ها نشان می‌دهد که هزینه‌ی پیش‌بینی شده قابل اعتمادتر است.

دو مقاله دیگر طبقه‌بندی در مورد درختان تصمیم انجام شده است. در مقاله‌ی [۱۰۸]، یک متغیر گسسته پیشنهاد شده است و یک 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]، همگنی داده‌های هزینه را در حوزه‌های دستگاه، تجزیه و تحلیل می‌کند و بر حس جاسازی تمرکز می‌کند. مدل‌های داده بین دامنه‌ای Equatinf با مدل داده‌ی دامنه، سه نصب آزمایشی ایجاد می‌کنند.

یک مقاله که بر روی الگوریتم ژنتیک انجام شد [۱۱۰] پیشنهاد کرده است که از یک NN برای ایجاد فهرستی از متخصصان سازماندهی شده برای هر معیار استفاده شود. ترکیبی از رویکردهای برنامه ریزی تخلیه نیمه اتوماتیک و تخصیص موقعیت نیمه اتوماتیک نیز اجرا شده است. نتیجه انباشته یک برنامه تکراری برای جزئیاتی است که سازنده روی آن کار می‌کند.

یک مقاله در مورد یادگیری عمیق انجام شد. در مقاله‌ی [۱۱۱]، دو معماری یادگیری عمیق قابل اعتماد وجود داشت: شبکه بزرگراه^{۱۳} و حافظه دوربرد^{۱۴}. چارچوب 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 پایه‌ی معیار و شش گزینه به طور قابل اعتمادی بهتر عمل می‌کنند.

۲.۷.۳. مطالعات انجام شده بر روی سایر روش‌ها

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

مدل اول، مدل‌های فرآیند نرم‌افزار. دو مقاله‌ی [۱۱۲، ۱۱۳] تحقیقات همکاری پژوهشی در برنامه‌های چابک (agile) را بررسی کردند. نمونه‌ها به شناسایی تضادهای بالا، از جمله جایگزین‌های کوتاه‌مدت برای تعدیل‌های کوتاه‌مدت و حل و فصل اختلافات منافع با استفاده از تکنیک‌های چابک کمک می‌کنند. دو مقاله‌ی [۱۱۴، ۱۱۵] گردآوری پیشنهادها را تصریح کرده‌اند، ارزیابی ویژگی‌های انتخاب‌شده و پیاده‌سازی یک کیت ML برای مفهوم آماری R. به‌علاوه، توضیح داده است که چگونه تکنیک‌های داده‌کاوی را می‌توان برای ساخت یک مدل طبقه‌بندی پیش‌بینی تلفات استفاده کرد. مدل پیشنهادی داده‌ها را از چند پارامتر

¹³ - highway network

¹⁴ - long-range memory

پروژه جمع آوری می‌کند و یک پروژه را به یکی از سه کلاس طبقه بندی می‌کند. مقاله‌ی [۱۱۶] یک رویکرد ارزیابی ریسک پروژه نرم افزاری را پیشنهاد کرده است که از اعتبارات برای اندازه گیری اثرات عوامل خطر برای رفع نظرات نادقیق و تناقضات بین کارشناسان استفاده می‌کند. استنتاج شبکه اعتباری به پیش‌بینی و تشخیص خطر کمک کرد.

در مقاله‌ی [۱۱۷]، به اهمیت روش‌های چابک و سیستم‌های پیشرفته، مانند اینترنت اشیا، مه^{۱۵}، و ابر^{۱۶} پرداخته شده است. بنابراین، نرم افزاری از ادغام یک 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 چارچوب پیشنهادی گنجانده شده‌اند. نویسندگان مقاله‌ی [۱۱۸] به طور سیستماتیک یک سیستم کمیت قوی یکپارچه مدیریت ریسک را ارائه می‌دهند که بر مشارکت بین اندازه پروژه و مشارکت تصمیمات مخاطره آمیز تمرکز دارند. سازندگان این رویکرد در یک پایگاه داده پروژه واقعی از طریق یک برنامه کاربردی تحلیلی نشان داده شده است.

مدل دوم، کنترلر منطق فازی. در دو تا شش مرحله برآورد تلاش نرم افزار، دو مقاله [۱۱۹، ۱۲۰] افزایش راندمان کنترل‌کننده‌های روان و منطقی را با بهبود کنترل‌های سوخت مبتنی بر مورد با حداقل اندازه بر اساس کنترل پیشنهاد کرد. نتیجه آبشار کنترل کننده منطق سیال است. قوانین مدل‌هایی که توسط خوشه‌بندی تفریقی ایجاد می‌شوند، امکان کاهش بیشتر را فراهم می‌کنند.

مدل سوم، برنامه ریزی انتشار استراتژیک (SRP). نویسندگان مقاله‌ی [۱۲۱] دریافته‌اند که نرم افزار مورد استفاده به عنوان یک افزونه برای چارچوب‌های تولیدی که اغلب استفاده می‌شود به بهبود عملکرد فرآیند کمک می‌کند. SRP یک مرحله‌ی مهم در ایجاد برنامه‌های کاربردی تکراری است. در قالب محدودیت‌های سخت و نرم، مانند زمان، تعهد، ثبات و پول، SRP شامل تخصیص ویژگی‌ها یا شرایط برای انتشار است.

مدل چهارم، مدل بلوغ. در [۱۲۲]، داده‌های بدون ساختار عظیم، که تحت چارچوب مدل بلوغ شایستگی دیجیتال (DCMM) از طریق تجزیه و تحلیل دقیق هدف، فرآیندهای مدیریتی یا عوامل تأثیرگذار این پروژه توسعه یافته‌اند. پیش بینی می‌شود که مقایسه مهارت‌های ذخیره سازی داده های مرتبط تأثیر مطلوبی داشته باشد. یک روش تثبیت شده برای بهبود عملکرد و استاندارد طلایی برای ایجاد برنامه و دستگاه برای بیش از ۲۰ سال ساخته شده است.

مدل پنجم، وزن دهی متوالی رو به جلو. نویسندگان [۱۲۳] الگوریتم‌های کارآمدی را برای تعمیم انتخاب زیرمجموعه ویژگی‌ها به یک رویکرد وزن‌دهی ویژگی عملی توسعه و ارزیابی کردند. این الگوریتم دقت را بیشتر بهبود می‌بخشد، زیرا همه ویژگی‌ها به طور یکسان در حل مسئله برای تعیین وزن به عناصر برای بهبود دقت تخمین کمک نمی‌کنند. پس از آن، آزمایش‌هایی را انجام داده‌اند که بر اساس طراحی اندازه‌گیری‌های مکرر بر روی مجموعه داده‌های دنیای واقعی برای ارزیابی این الگوریتم‌ها است.

مدل ششم، مدل تخمین استدلال مبتنی بر مورد (CBR). نویسندگان مقاله‌ی [۱۲۴] پیشنهاد کردند که سیستم‌های خوشه بندی (CBR-Cs) CBR را می‌توان برای ارائه یک برآورد هزینه دقیق ایجاد کرد. هدف رویکرد CBR-C تعیین کمیت خطاها و زمان و اجازه دادن به مدیران برای درک آسان فرآیندهای ارزیابی است. این مطالعه نشان می‌دهد که رویکرد CBR-C پیشنهادی یک ساختار برآورد هزینه پروژه طراحی نرم‌افزار جامع را ارائه می‌دهد.

مدل هفتم، مدل زمین-ماه^{۱۷}. نویسندگان مقاله‌ی [۱۲۵] یک مدل مدیریت ریسک برای پروژه‌های ماه زمین ارائه کردند که ویژگی‌های ایجاد نرم افزار را مطابق با تئوری پیاده سازی شده پروژه چرخه عمر نرم افزار در نظر گرفت. تحولات اصلی که در پارادایم کنونی مورد استفاده قرار می‌گیرد اغلب تجزیه و تحلیل و حل می‌شوند.

مدل هشتم، پیش‌بینی نقص پروژه. در مقاله‌ی [۱۲۶]، اثرات ساده سازی داده‌ها توصیف و کمی سازی شد. آزمایش‌ها انجام شده‌اند 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). طبق یک گزارش در مقاله‌ی [۱۲۷]، مقیاس، کارایی و پیچیدگی Treeboost ورودی‌های مدل هستند. مدل Treeboost با استفاده از چهار معیار خروجی MMRE، PRED، MdMRE و MSE در برابر مدل رگرسیون چند خطی و مدل موردی مورد آزمایش قرار گرفت و به عنوان یک مدل رگرسیون چند خطی ایجاد شد. مدل Treeboost را می‌توان برای کمی کردن تعهد خروجی نرم افزار استفاده کرد.

۴. بحث

این تحقیق به بررسی مطالعات ضروری در مدیریت پروژه 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 به عنوان طبقه بندی پیشنهاد شده است. در یک زمینه تحقیقاتی، ایجاد یک طبقه بندی ادبیات ممکن است چندین مزیت داشته باشد، یکی در حال تغییر. از یک سو، طبقه بندی آثار ادبی معمولاً تبلیغ می‌شود.

سوال سوم تحقیق: شکاف‌های موجود برای چشم انداز تحقیق در زمینه مدیریت پروژه نرم افزاری چیست؟

محقق جدیدی که در حال تحقیق در مورد ارزیابی پروژه نرم افزاری است ممکن است تحت تأثیر طیف گسترده اسناد در این بخش، فقدان نوع خاصی از چارچوب، و بررسی این زمینه قرار گیرد - مقالات متعددی در مورد این موضوع به روندهای نوظهور در مدیریت پروژه می‌پردازند. مطالعات خاصی امروزه مدل‌ها و پیاده‌سازی‌های ML را ایجاد کرده‌اند. طبقه بندی ادبیات به سازماندهی این آثار و حوادث متنوع به طور قابل توجهی کمک می‌کند و قابل استفاده و سازگار است. از سوی دیگر، روش شناسی طبقه بندی، بینش مفیدی را در مورد موضوع به محققان می‌دهد. در مرحله بعد، موضوعات آینده برای مطالعه را شرح می‌دهد. طبقه بندی در تجزیه و تحلیل فعلی ارزیابی نرم افزار 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 پیاده سازی فناوری‌های ML، از جمله آخرین ارزیابی پروژه نیز پوشش داده شده است.

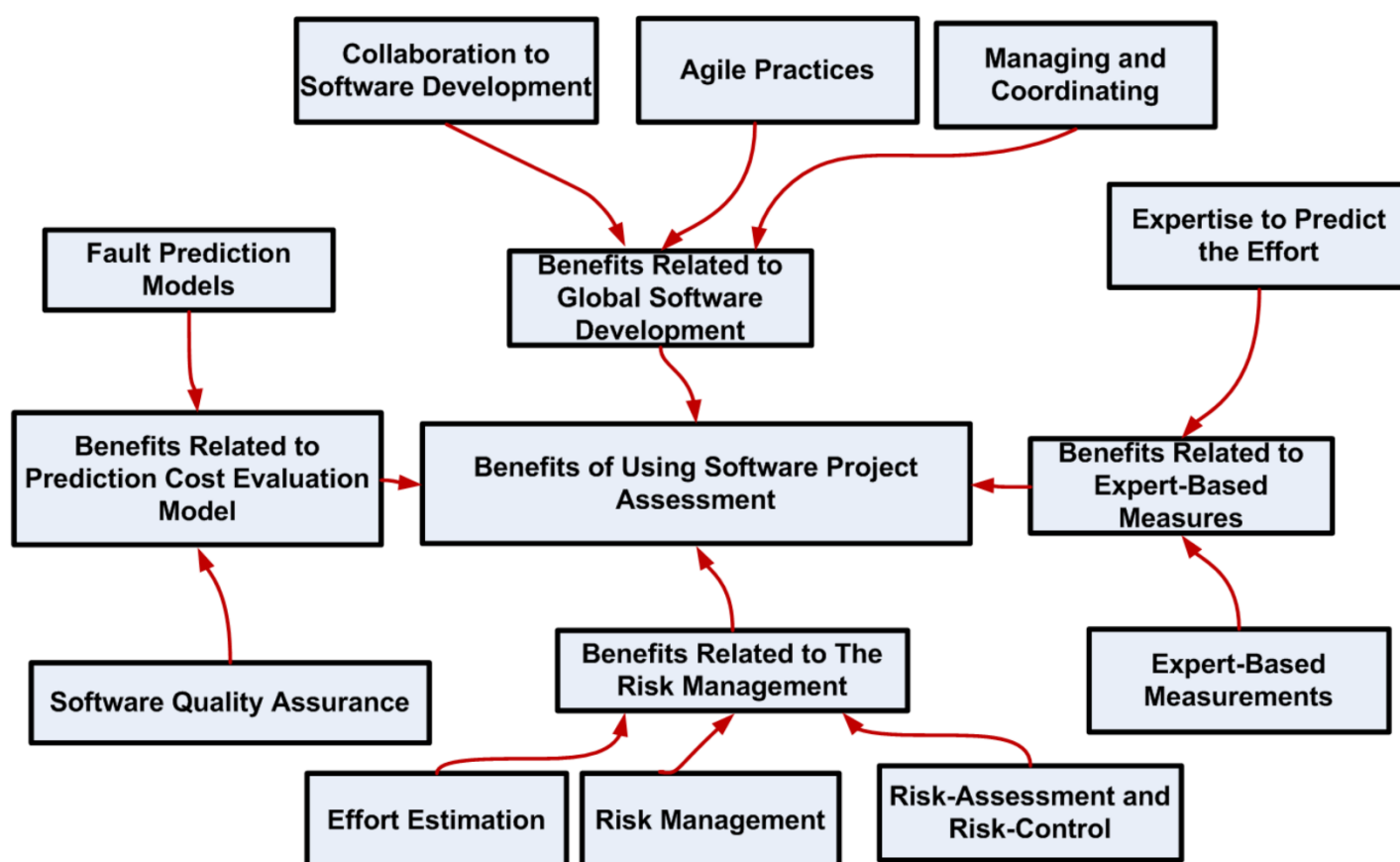
ثانیا، یک طبقه بندی می‌تواند کمبودهای مطالعه را طبقه بندی کند. نگاشت ادبیات، پوشش مطالعاتی ضعیف و قوی را در پیشنهادها بررسی پروژه در دسته‌های مختلف نشان می‌دهد. به عنوان مثال، طبقه بندی ارزش بررسی و ارزیابی گروه‌های ادعاهای فردی را به هزینه روش‌ها و ساختارهای تلفیقی و فعالیت‌های رشد نشان می‌دهد (که در فراوانی دسته‌های آنها بیان

می‌شود). طبقه بندی همچنین عدم تحقیق در مورد توسعه بازبینی پروژه را پس از یک تحقیق کافی نشان داد. ادبیات برای تحقیق ضروری است. مطالعات در این بخش با هدف بهبود و تبادل ML انجام می‌شود.

تجزیه و تحلیل آماری در بخش‌های جداگانه طبقه‌بندی، صنایع شرکت‌کننده در روش ML را برای مقابله با تحولات در حال ظهور و بهبود زمینه‌های غیرفعال طبقه‌بندی می‌کند. این مطالعه طبقه‌بندی را ارائه می‌دهد که در آن محققان ممکن است با همکاری و تجزیه و تحلیل فناوری‌های جدید، مانند پیشرفت‌ها، مطالعات مقایسه‌ای، و ارزیابی پروژه با استفاده از فناوری ML، که مشابه طبقه‌بندی در زمینه‌های دیگر است، همکاری کنند. تجزیه و تحلیل سه بخش از ادبیات را نشان می‌دهد: عوامل پشت ظهور مدیریت پروژه خودکار با استفاده از فناوری‌های ML، چالش‌های استفاده موفق از آن روش‌ها، و توصیه‌هایی برای غلبه بر این مشکلات.

۱.۴.۱. انگیزه‌ها

مزایای استفاده از پلتفرم مدیریت پروژه ML شفاف و قانع‌کننده است. این بخش برخی از مزایای ادبیات را مورد بحث قرار می‌دهد که بر اساس مزایای منحصر به فرد طبقه‌بندی می‌شود. منابع مناسب برای بحث بیشتر ذکر شده است (شکل ۶ را ببینید).



شکل ۶. مزایای استفاده از ارزیابی پروژه نرم‌افزاری.

۱.۱.۴. 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 عبارتند از [۱۲۸]: (۱) چندین رویکرد برای

رای گیری تحت الشعاع روش های دیگر. (۲) معیارهای مکانیزم کد منبع انتخاب شده برای استفاده از معیارهای منبع دستگاه پیشنهادی، که در پروژه های نرم افزاری با درصد کلاس های خطا کمتر از مقدار آستانه توصیه شده، در مقایسه با سایر روش ها مفید هستند. و (۳) رویکرد پیش بینی شکست.

۲.۱.۴. مزایای مربوط به مدیریت ریسک

فعالیت های مختلف برنامه ریزی نرم افزار را می توان به دو روش مؤثر تقسیم کرد، یعنی ارزیابی مشارکت و کاهش ریسک [۱۲۹]. برآورد هزینه تلاش نرم افزار به بسیاری از ملاحظات هزینه بستگی دارد و کنترل ریسک نیاز به شناسایی، درمان و حذف ریسک نرم افزار قبل از نتایج غیرمنتظره دارد [۱۳۰]. تعهد پیش بینی شده به رشد نرم افزار تلاش ها را تعیین می کند. ارزیابی ریسک، فعالیت اصلی در مرحله برنامه ریزی پروژه [۱۳۱]، یک عنصر حیاتی در ارزیابی موفقیت پروژه توسعه نرم افزار است. با این حال، قضاوت و تجربه انسانی در موفقیت فعالیت های مدیریت ریسک در روش های سنتی کاهش ریسک ضروری است و ارزیابی ریسک برای پروژه نرم افزاری غیرضروری و پرهزینه در نظر گرفته می شود [۱۳۲].

۳.۱.۴. مزایای مربوط به توسعه جهانی نرم افزار (GSD)

افزایش استفاده از GSD برای به حداقل رساندن هزینه های تولید و باز بودن بسته گسترده ای از تجزیه و تحلیل حرفه ای یکی دیگر از پیشرفت های حیاتی در بازار است. سرمایه گذاری های GSD اغلب چالش های مهمی ایجاد می کنند، اما محبوب تر می شوند. اینها شامل نگرانی های ارتباطی بین شرکت کنندگان پروژه، مشکلات در ایجاد ارتباطات قابل قبول جامعه، مشکلات فرهنگی، و موانع در مدیریت و سازماندهی کار در پروژه های اجرایی است. به طور خلاصه، کار تیمی فشرده در تولید نرم افزار در یک محیط توزیع شده دشوار است [۱۳۱]. با تمرکز بر ارتباطات چهره به چهره، که برای مدیریت زمینه های GSD سخت و پیچیده است، ادغام برخی از رویکردهای چابک از همان اولین تلاش چالش برانگیز به نظر می رسد.

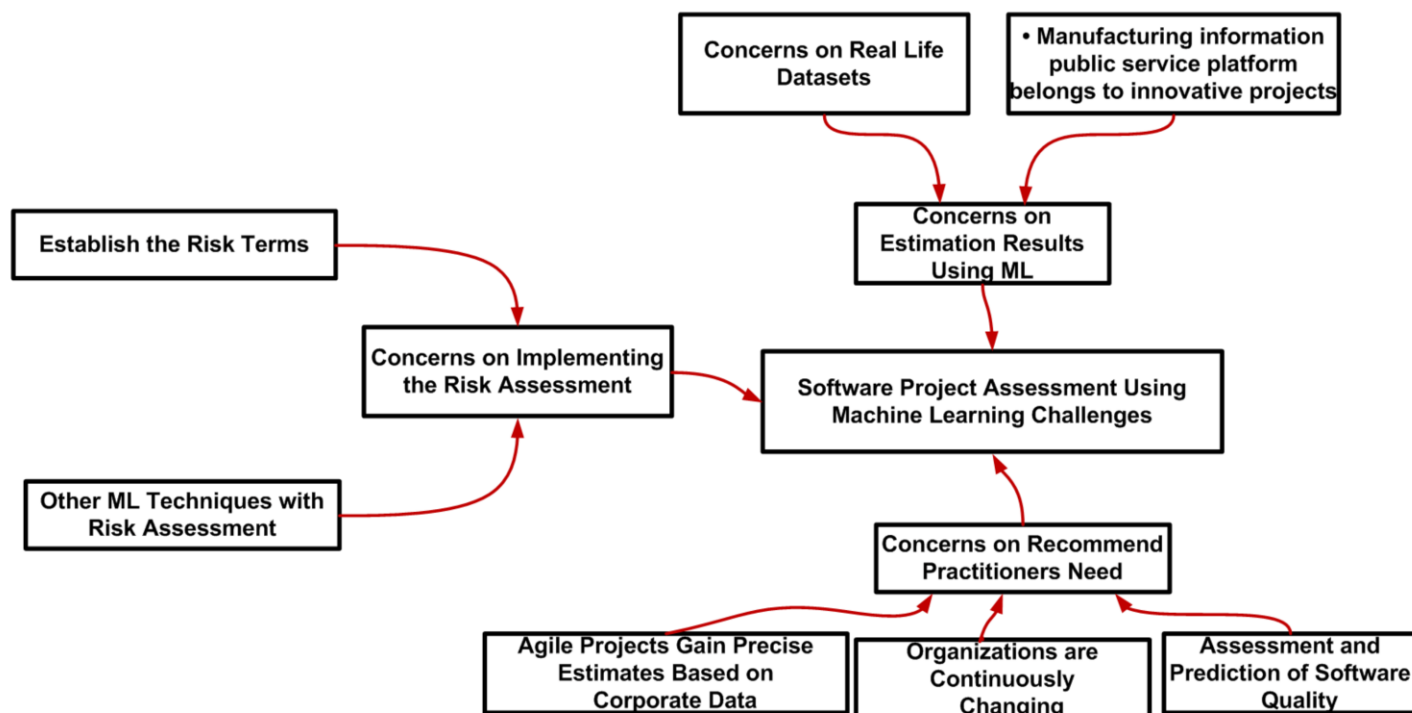
۴.۱.۴. مزایای مربوط به اقدامات مبتنی بر خبره

پیش بینی کننده از تجربه خود برای پیش بینی کار سرمایه گذاری هایی مانند کارشناسان استفاده می کند. مهارت برآوردگر به موضوع و تجربه او از اقدامات مشابه و متعارف بستگی دارد. اگر تعداد محدودی از مراحل نیاز به حذف اندازه گیری های متخصص داشته باشد، مدل متمرکز سود قابل توجهی خواهد داشت. نباید از جایگاه رفتار متخصص غافل شد. مفهوم ارزیابی ساخته شده بدون معیارهای حرفه ای حتی بدتر از مدل سازگاری و رویکرد تضمین کیفیت مورد استفاده برای جلوگیری از ارزیابی هایی است که نیاز به تحقیقات تخصصی دارند - پژوهش مدل ارزیابی محدودیت های زیادی را در جمع آوری معیارهای پیش بینی برای کار بالقوه ارائه می دهد. اشکالات بالقوه ممکن است تعداد ابزارهای خاص موجود برای ارزیابی رفتار، سازگاری و پایبندی به کاربرد یک سیستم یا دقت اندازه گیری باشد [۱۳۲].

۲.۴. چالش ها

اگرچه تکنیک های آموزشی کامپیوتری که در ارزیابی SPM استفاده می شوند، چندین مزیت را ارائه می دهند، چنین فناوری هایی به عنوان راه حل ایده آل برای پروژه های در حال تکامل در نظر گرفته نمی شوند [۱۳۳]. بررسی ها نشان می دهد که

محققان به ارزیابی پروژه‌ها و استفاده آنها از استراتژی‌های ML علاقه مند هستند. موانع اصلی در اجرای تکنیک‌های ML، علاوه بر موضوعات اضافی، در زیر ذکر شده است. مشکلات تعریف شده است (شکل ۷ را ببینید).



شکل ۷. ارزیابی پروژه نرم افزاری با استفاده از چالش‌های یادگیری ماشین.

۱.۲.۴. نگرانی در مورد نتایج تخمین با استفاده از ML

تجزیه و تحلیل سؤالات بیشتری را در مورد مجموعه داده‌های واقعی که فاقد رویکردهای توسعه نرم افزار اساسی هستند و به معیارهای دقیق دیگری نیاز دارند که می‌تواند برای محاسبه تلاش استفاده شود، آشکار کرد. رویکردهای ارزیابی مختلفی برای تأیید یافته‌های پیش بینی تلاش برنامه در تحقیقات بیشتر مورد نیاز است. اعتبار سنجی متقاطع، رویکرد Jackknife و روش تکراری عمده اعتبارسنجی می‌شوند. علاوه بر این، روندهای مطالعاتی نشان داده‌اند که روش‌های محاسبه نیاز به بررسی و تقویت دارند. علاوه بر این، مجموعه داده‌های واقعی ممکن است معیارهای اندازه و سایر روش‌های ML، از جمله درخت‌های رگرسیون را نیز بررسی کنند. پورتال خدمات عمومی متعلق به ابتکارات خلاقانه در زمینه تولید دانش است. رویه‌های توسعه باید بیشتر توسط تیم پروژه ما به روز شوند و مهندسان نمی‌توانند پلت فرم توسعه و زیرساخت‌هایی را که تأثیر بیشتری بر کارایی محصول دارد نظارت کنند. ارزیابی پروژه نرم افزاری با استفاده از چالش‌های یادگیری ماشین.

۲.۲.۴. نگرانی در مورد اجرای ارزیابی ریسک

نگرانی در مورد اجرای ارزیابی ریسک: زمانی که پروژه شروع می‌شود. ابتدا باید تهدیدات و عوامل خطر پروژه را تعریف کنیم و شرایط ریسک اصلی را تعیین کنیم، یعنی تغییر تقاضای ریسک، خطر زیرساخت، هماهنگی پرسنل و رویکردهای حفاظت از دستگاه و غیره. شبکه ارزیابی ریسک، همراه با تخصص از کارشناسان، بر روی موارد فعلی برای ایجاد فرآیند یادگیری مورد ریسک متمرکز شده است. در پرتو دو سیستمی که بالاترین نتایج موفقیت را به دست آورده‌اند، ارزیابی درجه ریسک یک شرط نیز ضروری بود که هزینه‌های تجدیدنظر در سطح ریسک آن به مناسب‌ترین چارچوب برای طبقه‌بندی بیزی تبدیل شد. در این زمینه توصیه می‌شود، زیرا برای سایر تکنیک‌ها مانند ML، نتایج تخمین بهبود یافته را ارائه می‌دهد [۱۲]. دلیل نسبتاً قطعی

تری برای دقت بالاتر وجود دارد، زیرا داده‌ها از یک مجموعه داده موجودیت واحد به دسته‌های مختلف همگن، بسته به شرکت یا بخش استخراج می‌شوند. دقت برآورد تلاش از طریق طبقه‌بندی [۱۳۴، ۱۳۵، ۱۳۶] افزایش یافت.

۳.۲.۴. نگرانی در مورد نیاز به توصیه‌ی پزشکان

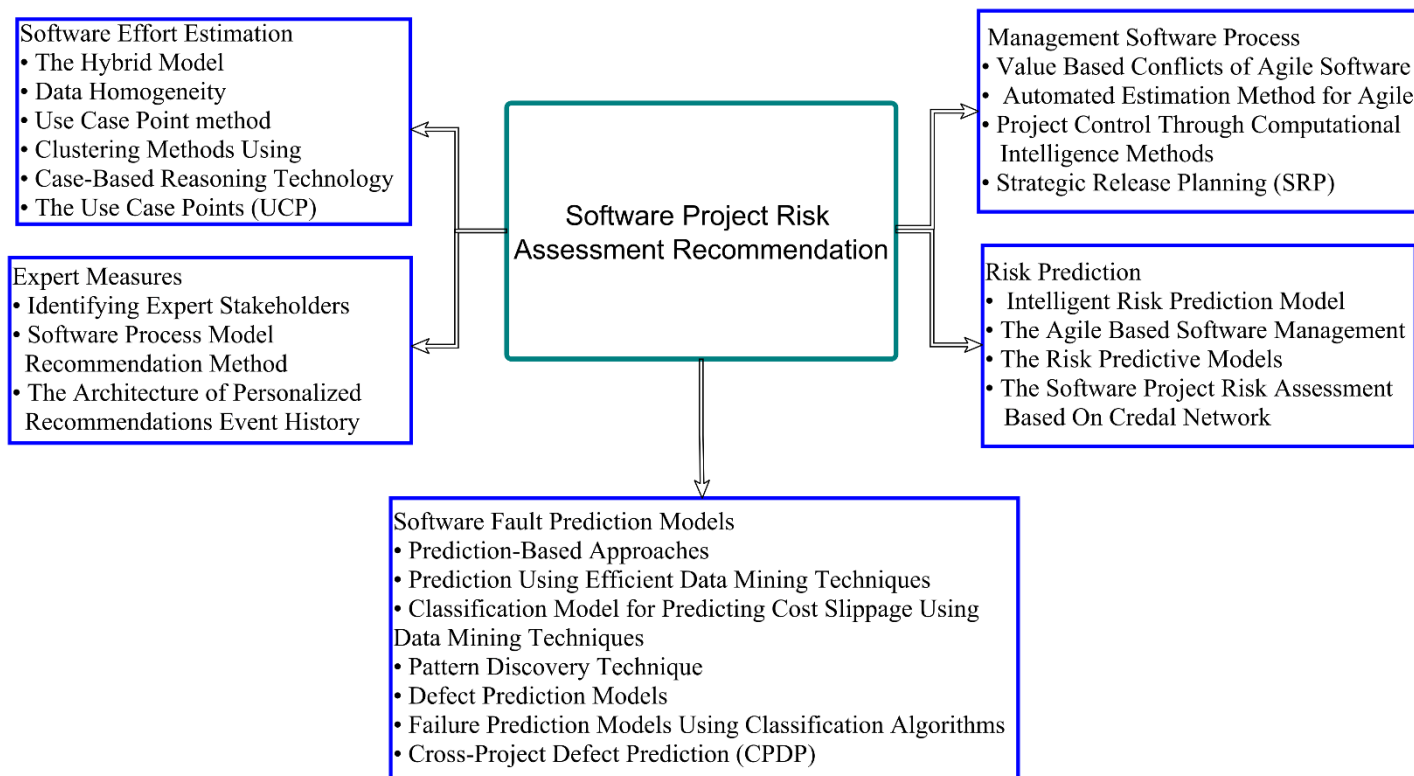
در مورد نوع روش‌های محاسبه‌ای که استفاده می‌شود و نوع مجموعه داده‌هایی که برای پروژه‌های خود استفاده می‌کنند، مراقب باشید. پروژه‌های چابک برآوردهای دقیق‌تری را بر اساس داده‌های شرکت به دست می‌آورند. در مقایسه با داده‌های ملی، کسب و کارهای خصوصی از مقدار محدودی از داده‌های پروژه داخلی سود می‌برند [۱۳۲]. همانطور که شرکت‌های رشد تدریجی مولدتر و چابک‌تری مانند XP را پیاده‌سازی می‌کند، 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 با MMRE کم نشان داده شده است، علی‌رغم حفظ همان بهره‌وری [۱۳۷].

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

با توجه به ارزیابی و پیش‌بینی کیفیت نرم‌افزار در سازمان‌های بزرگ نرم‌افزاری، پیش‌زمینه 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 ارزیابی مستند شده است. در ISO 9126 [۱۳۹]، کیفیت به عنوان «کامل بودن ویژگی‌ها و ویژگی‌های یک محصول نرم‌افزاری که قادر به برآوردن نیازهای اعلام شده یا ضمنی است، تعریف می‌شود. در ISO 25000 [۱۴۰]، ظرفیت محصولات نرم‌افزاری برای برآوردن مشخصات مشخص شده، تحت برخی شرایط» رویکرد سازگاری اتخاذ شده است. شناسایی و تخصیص منابع در جایی که بیشتر نرم‌افزار برای ارزیابی کیفیت نرم‌افزار در مراحل اولیه توسعه ضروری است [۱۴۱].

۳.۴. توصیه‌ها

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



شکل ۸. دسته بندی توصیه‌ها برای استفاده از ارزیابی پروژه نرم افزاری.

۱.۳.۴. توصیه‌هایی برای تخمین تلاش نرم افزار

اندازه گیری مشارکت نرم افزار برای برنامه های نرم افزاری موفق ضروری است. اگر بخواهیم در کسب و کارهای نرم افزاری به دنبال بودجه خاصی باشیم، آماده سازی و هزینه پروژه به درستی انجام می‌شود، تخمین تلاش محاسباتی دقیق ضروری است. با توجه به تغییرات قیمت، می‌توان سودهای تجاری بیش از حد برآورد شده را از دست داد. از طرف دیگر، دست کم گرفتن برنامه و هزینه‌ها را تحت الشعاع قرار می‌دهد و برای کسب و کار یک تن سرمایه هزینه می‌کند. از آنجایی که هزینه تلاش منعکس کننده هزینه بالا است، ادبیات، در عوض، از محاسبه تلاش نرم افزاری و اصطلاحات محاسبه هزینه نرم افزار استفاده می‌کند که به ارزش تقریبی اشاره دارد.

RQ4. معیارهای پیش‌بینی و سطح دقت فعلی آن‌ها که توسط تکنیک‌های تخمین مختلف اثبات می‌شود، چیست؟

مدل ترکیبی، برای مدت طولانی، تخمین تکمیل فعالیت های برنامه، یک زمینه مطالعاتی ضروری است. بنابراین، ما چندین مدل مستقل با عملکرد خوب را برای بهبود دقت و قابلیت اطمینان اثرات پیش‌بینی می‌خواستیم. یک نمونه اولیه ترکیبی شامل سه مجموعه مجزا از ویژگی‌ها است: (۱) یکی یک ویژگی مبتنی بر متن است که به عنوان جمع شناخته می‌شود. (۲) یکی شامل ویژگی های مبتنی بر متن است که به عنوان تعریف شناخته می‌شوند. و، (۳) و دارای انواع ویژگی های مبتنی بر فراداده است. در مقایسه با مدل‌های مدل قبلی در [۱۳۷]، این مدل همچنین 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.

همگنی داده‌ها همگنی هزینه دامنه کاربرد نتایج. این سوال قبلاً در ادبیات تخمین ماشین مورد بحث قرار نگرفته بود. تأثیر مقیاس داده‌های آموزشی موفقیت پیش‌بینی و موضوعی حیاتی که در تحقیق دیگری مورد بررسی قرار گرفته است را بررسی

کنید. مطالعات انجام شده برای رسیدگی به این مشکل هنوز تأیید نکرده است که این موضوع به دلیل تفاوت در کیفیت داده‌ها و الگوریتم‌های پیش‌بینی، یک مسئله بی‌پاسخ است. با این حال، آخرین یافته‌های تجربی مدیران پروژه را هدایت می‌کند تا تعیین کنند چه مقدار داده برای آموزش الگوریتم لازم است [۱۰۹]. از روش‌های مختلف ML برای اندازه‌گیری هزینه‌های برنامه و تجزیه و تحلیل 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، به طوری که کیفیت داده‌های ویژگی‌ها بهتر شود و خروجی پیش‌بینی آنها افزایش یابد. این ممکن است با تجزیه و تحلیل داده‌ها در حوزه دستگاه تعبیه شده به دست آمده باشد.

استفاده از روش Case Point مدل Treeboost، با تمرکز بر سه متغیر مستقل مقیاس نرم‌افزار، کارایی و پیچیدگی، تعهد نرم‌افزار را پیش‌بینی می‌کند. مدل Treeboost پیش‌بینی عملکرد برنامه را بر اساس فرآیند نقطه استفاده پیشنهاد کرد. از مدل Treeboost استفاده شد. اندازه نرم‌افزار، بهره‌وری و پیچیدگی شامل ورودی‌های مدل می‌شود. یک مدل رگرسیون چند خطی ایجاد شد و مدل Treeboost با استفاده از چهار معیار عملکرد مورد ارزیابی قرار گرفت: MRE، PRED، MdmRE، و MSE در برابر مدل رگرسیون چند خطی و مدل نقطه‌موردی. 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، به عنوان مثال، سلسله‌مراتب خوشه‌ها، K-means، و حداکثر کردن ترجیحات [۱۴۴، ۱۴۵، ۱۴۶]. Dycom Clustering مشابه Dycom اصلی است که بر اساس چهار زیرمجموعه CC از چهار جدول SEE با اندازه‌های مختلف است. برای مثال، این تحقیق شامل طرح‌بندی توالی است. خوشه‌بندی K-mean Dycom به جداسازی برنامه‌های CC کمک می‌کند، با تحویل برابر با Dycom یا کارایی پیش‌بینی بهتری دارد. با این حال، تعداد زیرمجموعه‌های CC نیز نیاز به تعریف قبلی دارد و انتخاب اشتباه ممکن است تأثیر منفی بر نتایج پیش‌بینی داشته باشد.

با استفاده از فناوری استدلال مبتنی بر مورد (CBR)، چنین عواملی باعث می‌شود که یک CBR تعاملی برای رفع نیازهای مدیران پیشنهاد شود. این روش به دنبال ارتقای درک مکانیسم برآورد هزینه و ارائه دانش کافی به مدیران است. درک چرخه CBR ساده است، زیرا ذخیره و بازیابی اطلاعات در حافظه انسان را شبیه‌سازی می‌کند. مدیران با اقداماتی که انجام داده‌اند بهتر می‌توانند درک کنند که محاسبه چگونه انجام می‌شود. روش ANGEL با استراتژی CBR-C [۱۴۷] بهتر عمل کرد. توضیح اصلی برای این امر این است که رویکرد CBR-C فعلی دارای یک تابع نمایه‌سازی 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. قابل مقایسه‌ترین پروژه‌های گذشته برای تخمین اینکه چگونه هزینه پروژه فعلی می‌تواند از معیارهای مقایسه استفاده کند انتخاب می‌شوند [۱۴۳]. ۱. آیا شکلی از الگوریتم وزن دهی تابع برای ارزیابی تلاش استدلال گاه به گاه وجود دارد که روش‌های فعلی انتخاب تابع را بهبود بخشد؟ ۲. مجموعه داده‌ها چقدر بر این نتایج متکی است؟ ۳. آیا می‌توان دقت مجموعه داده‌ها را با جستجوی راه‌هایی برای نشان دادن اینکه یافته‌های ما دقیق یا بدون نویز هستند اندازه‌گیری کرد؟ ۴. اندازه فاز بررسی وزن با ویژگی‌های یک مجموعه داده را چگونه توصیف کنیم؟

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

۲.۳.۴. توصیه‌هایی برای اقدامات مبتنی بر خبره

دانش، تخصص، تجربه و شهود متخصص بر اساس دانش گروه‌های پروژه در ۱۰ زمینه مانند قیمت‌ها، زمان، مسافت، بهره‌وری و مدیریت منابع است (اغلب به برآوردهای بیش از حد خوش بینانه کمک می‌کند). این نشان دهنده مجموعه‌ای از فرآیندها، سیاست‌ها و شیوه‌هایی است که دانش مورد نیاز برای اجرای یک پروژه را ایجاد و جمع‌آوری می‌کند. دو لایه دانش - خرد و کلان - ایجاد شد.

آنها ذینفعان خبره و روشی را که برای ارزیابی ورودی‌های ذینفعان مورد استفاده قرار گرفت شناسایی کردند و موضوعات بازار را شناسایی کردند که نمایه‌هایی را برای بازتاب مشارکت ذینفعان در هر موضوع ایجاد می‌کند. متعاقباً، با در نظر گرفتن عوامل اساسی کمک‌کننده ذینفعان خاص، اغلب راه‌حل‌ها را در طیف وسیع‌تری از ابتکارات بررسی می‌کنند تا اینکه هر کدام را به طور مساوی مشاهده کنیم. رویکردهای تعریف شده در [۱۵۰] تقریباً به طور کامل بر روی محتویات اشیاء اختصاص داده شده به هر موضوع متمرکز هستند. در موارد خاص، استفاده تصادفی از کلمات مستلزم توزیع مقادیر سری کوچک برای موضوعات مفهومی مشابه است. به عنوان مثال، موضوعات حمل و نقل و نظارت، حتی اگر از نظر فنی به هم 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 توصیه مدل فرآیند نرم‌افزاری، به مدیران پروژه توصیه می‌شود که از بهترین مدل چرخه نرم‌افزاری برای پروژه فعلی بر اساس اثبات علم داده موجود در مراحل اولیه اجرا استفاده کنند. در رویکرد مدیریت پروژه، چارچوب جامعه را با توصیه برای ساختار پروتکل به چالش می‌کشد. ثانیاً، تفاوت‌ها در الگوریتم‌های طبقه‌بندی و انتخاب جایگزین را تجزیه و تحلیل می‌کند، که با یک مدل پیشنهادی شامل ارقام ایجاد نرم‌افزار تاریخی برای تخمین یک مدل فرآیند نرم‌افزار مدرن تنها با برخی جزئیات همراه است [۱۵۱]. استفاده از این شکل از اطلاعات به طور کامل: (۱) به مدیران پروژه در انتخاب سیستم نرم‌افزاری مناسب برای پروژه فعلی در یک نقطه دقیق در مرحله ایجاد کمک می‌کند. (۲) اثر متقابل بین مدل‌های فاز و اشکال مختلف پروژه فاکتور را ارزیابی می‌کند تا به مدیران پروژه اجازه دهد تا مناسب‌ترین مدل فرآیند را انتخاب کنند. و (۳) از بهترین رویکرد استفاده می‌کند. یک چارچوب توصیه خودکار متمرکز بر مدل فرآیند نرم‌افزار ML استفاده می‌شود تا مدیریت پروژه را قادر سازد تا با توجه به نتایج مهندسی نرم‌افزار تاریخی، تصمیم بگیرد که کدام مدل فرآیند نرم‌افزار در مرحله توسعه اولیه برای یک پروژه جدید مناسب‌تر است [۱۵۲].

معماری توصیه‌های شخصی‌شده، رابطی را ارائه می‌دهد که به تیم‌های تحقیقاتی داده کمک می‌کند تا به طور مؤثر در طرح‌های ML مشارکت کنند. خروجی، بازخورد سفارشی میلیون‌ها متقاضی را می‌دهد، در کمتر از یک ثانیه به پرسش‌ها پاسخ می‌دهد و

دانش جدیدی به ارمغان می‌آورد. بنابراین، بر روی پیاده‌سازی منبع باز Antelope اجرا می‌شود و ایده تاریخچه پرونده، یک پلت فرم مهندسی اطلاعات انعطاف‌پذیر را ادغام می‌کند. همچنین ممکن است از طیف وسیعی از ابزارهای ML همراه با فناوری استفاده کند که در اینجا توسعه یافته است و می‌تواند امکان ادغام عمیق‌تر را فراهم کند، حتی برخی که فقط با سیستم‌های مدیریت داده سنتی رابط دارند [۱۵۳].

۳.۳.۴. توصیه‌هایی برای فرآیند نرم افزار مدیریت

Agile در تلاش است تا ابتدا با ایجاد مهمترین ویژگی‌ها، تأثیر سازگاری نازک را به حداقل برساند. در مقایسه، پروژه‌های معماری در محیط‌های پروژه بزرگ‌تر نیز با مشکلات خروجی مواجه می‌شوند که مشخصات کلی سیستم را منصفانه می‌دانند. این می‌تواند در مورد جوامع چابکی که با محصولات غیرچابکی برخورد می‌کنند که باعث تعارض بین سطوح چابک و غیرچابک می‌شود، اعمال شود [۱۵۴]. هنگام پیروی از رویکرد چابک، گرایش‌ها ممکن است برای ارائه راه حل کوتاه‌مدت با بهبودهای کوتاه‌مدت و طرح حل تعارض ارزشی بلندمدت، اختلاف را بهتر شناسایی کنند. اکنون فرهنگ‌های مختلف در حال همگرایی و تعامل هستند و دیگر نظام‌های مذهبی با هم تلاقی می‌کنند. این می‌تواند شامل روابط مالیاتی، فناوری، عملکردی، شرکتی یا ارتباطی باشد [۱۵۵].

تضادهای مبتنی بر ارزش سیستم‌های چابک نرم افزار چابک مستقل هستند. در ادبیات وجود دارد که یک سازمان یک عامل مهم در توسعه چابکی است. ادبیات نشان می‌دهد که اگر یک پروژه چابک محیط پروژه خود را تحت تأثیر قرار ندهد یا آن را تغییر ندهد، از انجام کارهایی که باید به دست آید، حمایت بسیار کمی وجود دارد. هیچ راهنمایی در مورد اینکه کدام اختلافات و مداخلات می‌توانند نظارت و مدیریت شوند ارائه نشده است. این رویکرد شامل توسعه محصول چابک، تضادهای مبتنی بر ارزش، و ابتکارات نهادی توسط [۱۱۳] است که مشکلات و راه حل‌ها را تشخیص می‌دهد. تجربه من از یک تیم توسعه نرم افزار Agile است که در فضایی غیر پروژه‌ای کار می‌کند و ممکن است تأثیر و/یا تغییر قابل توجهی نداشته باشد. آرمان‌های افراد و نهادها بخشی از محیط است و بنابراین مفاهیم چابک باید مورد ارزیابی قرار گیرند.

روش تخمین خودکار Agile به طور موثر روش تخمین کارت خودکار را برای داده‌های انسانی پیش‌بینی شده تاریخی با آخرین الگوریتم‌های ML اعمال می‌کند [۳۱]. رویکرد "برآورد خودکار" محبوب‌ترین شکل آماده سازی دستی پوکر را در محیط‌های چابک تقویت می‌کند [۱۵۴]. خود تخمینی از ویژگی‌های کارت داستان در یک محیط چابک استفاده می‌کند: (الف) دقت تخمین را با کاهش تأثیر تخمین‌های اشتباه به حداکثر می‌رساند. (ب) نشان می‌دهد که تخمین خودکار آمادگی پوکر را در قسمت آخر پروژه افزایش می‌دهد. و (ج) ارزش نوشتن کارتهای داستانی با طراحی مناسب را تعیین کنید.

کنترل پروژه از طریق روش‌های هوش محاسباتی با مدیریت داده‌های عددی و زبانی، نویز خطای محاسباتی، درک انسانی و اصول تصمیم‌گیری مبهم مرتبط است. همچنین 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 پیش‌بینی شده برای ایجاد مدل‌های کنترل پروژه مدرن و منابع فناوری اطلاعات اشاره دارد که شامل چارچوب‌های مبتنی بر ML و مراقبت از عدم دقت اطلاعات، ابهام یا ابهام توسط معیارهای اصلی موفقیت است که به کل زمینه‌های دانش مرتبط است. معرفی کتابخانه‌های ارزیابی یادگیری مدرن و چارچوب‌های توسعه منبع باز برای مدیریت پروژه، حوزه‌ای از مطالعه را باز می‌کند که به همگرایی فنی منابع فناوری اطلاعات مرتبط است [۱۵۶، ۱۵۷].

برنامه ریزی انتشار استراتژیک (SRP): یک گام مهم در رشد نرم افزارهای تکراری. SRP شامل تحویل، مانند ترکیب، کنترل های نرم، از جمله زمان، منابع، قیمت، یا پول، ویژگی ها یا الزامات انتشار است. SRP-Plugin نشان می دهد که برنامه های کاربردی مورد استفاده با برنامه مشترک به بهبود بهره وری فرآیند توسعه کمک می کنند [۱۲۱]. این افزونه دارای یک اکوسیستم فضای بصری غنی با قابلیت های آماده سازی نسخه پیشرفته است که ظرفیت آماده سازی برای پرتاب ها، افزایش دوام و تقویت همکاری بین سهامداران پروژه را افزایش می دهد. SRP-Plugin با رویکردی قوی، دقیق و سازمان یافته و توانایی Release Planner برای تولید طرح های انتشار پیچیده، Visual Lab را بهبود می بخشد. با این حال، آماده سازی زمان بندی های انتشار فقط اولین مرحله در پیچیده کردن ایده هایی است که راهنمایی گسترده تری برای تصمیم گیری های برنامه ریزی انتشار استراتژیک ارائه می دهد.

۴.۳.۴. توصیه هایی برای پیش بینی ریسک

اگر پروژه های مرتبط با نرم افزار می خواهند کارایی خود را مستقل از حوزه کسب و کارشان بهبود بخشند، مدیریت ریسک ضروری است. انتظارات مصرف کننده هنوز در سیستم های اثبات شده که تحت نظارت دقیق قرار دارند مورد توجه قرار نمی گیرند [۱۵۸]. تعیین خطرات برای ابتکار یک ملاحظه اساسی در ارزیابی موفقیت پروژه یا رگرسیون است. تقریباً هر سازمانی از ابزارهای پیچیده برای طبقه بندی، به حداقل رساندن و حذف کلی آسیب استفاده می کند.

مدل پیش بینی ریسک هوشمند: آیا یک ابتکار با ریسک بالا به موقع شناسایی می شود؟ با این حال، مدل های فعلی در درجه اول بر این فرض متمرکز هستند که تمام هزینه های طبقه بندی خطا معادل هستند و تخمین احتمال در پروژه نرم افزاری است. هزینه پیش بینی یک پروژه شکست خورده به عنوان پروژه ای که احتمالاً به موفقیت می رسد با پیش بینی پروژه ای که احتمالاً به عنوان یک پروژه شکست خورده موفق می شود متفاوت است. تا آنجا که ما درک می کنیم، در حالی که معمولاً در چندین زمینه استفاده می شود [۱۵۹]، رویکرد یادگیری حساس به هزینه هنوز در زمینه مدیریت ریسک پروژه نرم افزار برون سپاری اعمال نمی شود. در منطقه مورد مطالعه مدل پیش بینی ریسک پروژه نرم افزاری، دو حفره تحقیقاتی عمده وجود دارد. اولاً، مدل های پیش بینی ریسک که منحصر به یک پروژه نرم افزاری برون سپاری هستند، به ندرت مورد بررسی قرار می گیرند. دوم، اگرچه مطالعات پیش بینی ریسک پروژه نرم افزاری جامع هستند، هیچ محقق روش های یادگیری حساس به هزینه را در پیش بینی ریسک پروژه نرم افزاری اعمال نکرده است.

مدیریت نرم افزار مبتنی بر Agile برای بخش عمده ای از عملکرد پروژه در زمان های اخیر سازنده بوده است. تهدیدهای مرتبط با زمان، که بر انتشار محصولات قابل تحویل تأثیر می گذارند، تهدیدهای مبتنی بر زمان بندی را نشان می دهند. این به این دلیل است که منابع مالی، پیش بینی های غیرقابل اعتماد آن زمان و اقدامات مثبت مدیر پروژه به درستی توزیع نشده است. تهدیدات بودجه منعکس کننده مخاطرات مالی است که ممکن است از انبوه وجوه رخ دهد. چنین عواملی ممکن است به گسترش دسترسی ناخواسته پروژه، استفاده کم از خروجی های موجود و مدیریت ضعیف نسبت داده شود [۱۶۰].

اشکال خطر عملیاتی با رویه های منظم پروژه همراه است. رویه های اشتباه، برنامه ریزی ناکافی و قدرت تیمی از دلایل این تهدیدات هستند. در مورد ارزش روش های چابک و کاربرد چارچوب های مدرن در کنترل ریسک توسط منابع کافی بحث کنید. در آینده، تیم را قادر می سازد تا با ارزیابی پارامترهای ریسک، اثرات تهدیدات را تعیین کند. همچنین در صورت استفاده از چنین معیارهایی، احتمال قوی خروجی صدا وجود دارد. مدل های ریسک پیش بینی کننده: یک تخمین ۵۰ درصدی از نوع تاخیرهای مخاطره آمیز نرم افزار، تعداد درخواست های اسکن کلمه کلیدی را کاهش می دهد. دقت مدل های بیزی دریافتی با استفاده از

چندین مقیاس طبقه‌بندی اندازه‌گیری و مقایسه می‌شود. یک معماری شبکه بهینه شده برای درخت عملکرد آزمایشی موفقیت آمیز را برای همه مجموعه داده‌ها نشان می‌دهد. رابطه بین متغیرهای به‌دست‌آمده 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 توسعه نرم افزار شناسایی، اندازه‌گیری، آماده سازی و واکنش به ریسک های بالقوه برای جلوگیری از تاثیر آنها بر پروژه نرم افزاری است.

ارزیابی ریسک پروژه نرم‌افزاری مبتنی بر شبکه Credal است، یک روش مدرن که برای ارزیابی آسیب‌پذیری یک برنامه پروژه متمرکز بر شبکه Credal معرفی شده است. با استفاده از مجموعه اعتباری برای اندازه‌گیری تأثیر عوامل خطر [۱۱۶]، با نظرات متناقض کارشناسان و تفاوت های آنها مقابله خواهد کرد.

استنتاج شبکه اعتباری، پیش‌بینی ریسک و تشخیص ریسک را انجام داد. نتایج موردی نشان می‌دهد که اصل روش صحیح است و ارزیابی پروژه نرم‌افزاری به خوبی پیش‌بینی شده است. هدف مدل‌سازی تهدیدها برای پیش‌بینی ریسک‌ها و پیامدهای خطرات و تعریف عوامل خطر اصلی است که استراتژی ریسک و مدیریت ریسک را ارتقا می‌دهد. با این حال، یک دوره توسعه طولانی، پیچیدگی بالای محصول، و بی‌ثباتی فوق‌العاده روش، پیش‌بینی و ارزیابی خطر پروژه نرم‌افزار را غیرممکن می‌کند. فلسفه ارزیابی ریسک پروژه نرم‌افزاری کنونی اغلب ریسک‌ها را با در نظر گرفتن ویژگی‌های سیستم‌های نرم‌افزاری و روش‌های توسعه اطلاعات، از مدیریت کلی پروژه تحلیل می‌کند.

۵.۳.۴. توصیه‌هایی برای مدل‌های پیش‌بینی خطای نرم افزار

پیش‌بینی تلاش و مدت زمان توسعه نرم‌افزار، وظیفه حیاتی برای SPM فعال است. دقت و قابلیت اطمینان مکانیسم های پیش‌بینی نیز مهم است. چندین الگوریتم ML برای پیش‌بینی مدت زمان نرم افزار استفاده شد.

رویکردهای مبتنی بر پیش‌بینی به یک ویژگی پیش‌بینی نیاز دارند که تعهد و دوره بالقوه پروژه را با توجه به داده‌های فعلی/گذشته پروژه پیش‌بینی می‌کند. با این وجود، الگوریتم های متعدد ML اغلب با وجود تعداد زیادی از الگوریتم های ML ارزیابی نمی‌شوند. برای مدل ماشین ساخت، بر اساس چندین جزئیات پروژه، الگوریتم‌های مختلف ML استفاده می‌شود [۱۶۱]. مدل ML شبکه رگرسیون عصبی جهانی (GRNN) نام گرفت تا کارایی متخصصان فناوری را پیش‌بینی کند. GRNN می‌تواند برای پیش‌بینی بهره‌وری پزشکان برای خطوط جدید و اصلاح‌شده کد، کدها و برنامه‌نویسان استفاده شود. تجربیات به عنوان متغیرهای مستقل دقت پیش‌بینی GRNN بهتر از رگرسیون آماری زمانی که از دو مدل برای پیش‌بینی بهره‌وری متخصصان نرم‌افزار اختصاص داده شده به‌صورت جداگانه استفاده می‌شود، استفاده می‌شود.

پیش‌بینی با استفاده از تکنیک‌های داده‌کاوی کارآمد برای این بخش از تجزیه و تحلیل، موفقیت/شکست پروژه‌ها را با یک محاسبه فاز به فاز، به جای سبک ارزیابی منظم از جنبه کل پروژه، اندازه‌گیری می‌کند. روش داده‌کاوی برای خوشه‌بندی و گروه‌بندی روش‌ها با جمع‌آوری داده‌ها از ابتکارات مختلف از طریق چندین بخش محاسباتی استفاده می‌شود. آنها همچنین پیشنهاد کردند که تکمیل پروژه در حال افزایش است و پروژه شکست خورده موفقیت آمیز باشد. برای پیش‌بینی شدت یک خطای برنامه، ناگوانی و بانسالی یک مدل رابط کاربری گرافیکی مدرن معرفی کردند [۱۶۲]. رویکرد خوشه‌بندی برای ساخت یک خوشه شدت مخزن اشکالات نرم‌افزاری با طول رفع مشکل استفاده شد. مدل پیشنهادی با استفاده از کد منبع باز، که اغلب توسط مخزن مسائل نرم افزار باز MySQL ارائه می‌شود، اعمال می‌شود [۱۶۳].

مدل طبقه‌بندی برای پیش‌بینی لغزش هزینه با استفاده از تکنیک‌های داده کاوی از بودجه و زمان‌بندی برای برنامه‌ریزی اولیه یک پروژه ICT استفاده می‌کند و سپس یک لغزش هزینه را در دسته پروژه پیش‌بینی می‌کند. سه دسته سقوط وجود دارد که طبیعی تلقی می‌شوند، لغزش متوسط و سقوط زیاد که نیاز به اقدام دارند [۱۱۵، ۱۶۴]. هدف توضیح این است که چگونه یک مدل طبقه‌بندی با استفاده از تکنیک‌های داده کاوی برای پیش‌بینی تلفات هزینه ساخته می‌شود. مدل پیشنهادی از ورودی (به عنوان مثال، بودجه اولیه و زمان‌بندی) تعداد محدودی از پارامترهای پروژه استفاده می‌کند و پروژه را به یکی از سه دسته (عادی، متوسط و بزرگ) تقسیم می‌کند.

تکنیک کشف الگو، اجرای آزمایشی در زیست‌شناسی محاسباتی با استراتژی کشف الگو است که به طور موثر اجرا شده است. این فناوری عادات رابطه‌ای را که از سوابق به ارث رسیده‌اند، آشکار می‌کند، و شاغلین کسب‌وکار را قادر می‌سازد تا دانش معناداری را به دست آورند و اعتماد را در تصمیم‌گیری بهبود بخشند. روندهای آماری مرتبط با نتایج درجه خوب برای مجموعه داده‌های آزمایش شده تولید شد [۱۶۵]. همچنین تأثیرات روی نتایج تکنیک‌های بودجه‌ای متعدد را نشان می‌دهد. اولین تحقیق از روش منحصر به فرد کاوی الگوها برای شناسایی نرم افزار معیوب در مهندسی نرم افزار استفاده می‌کند. یافته‌ها توانایی چنین استراتژی را برای ارائه عملکرد رتبه‌بندی مثبت و دانش مفید برای تصمیم‌گیرندگان نشان داده‌اند.

مدل‌های پیش‌بینی عیب ایجاد مدل‌های پیش‌بینی پروژه نرم‌افزاری برای افزایش تلاش برای شناسایی عیوب و توسعه یک مدل پیش‌بینی نقص برای یک شرکت بزرگ نرم‌افزاری صنعتی ارزشمند است. سیستم و اندازه‌گیری روش برای ایجاد مدل نشان می‌دهد که، حتی اگر چهار درصد از برنامه دارای نقص است، هدف این است که یک عامل ناکارآمد را در فرآیند توسعه مستمر پروژه برنامه‌ای عظیم وارد کرده و ابتکار ارزیابی را کاهش دهد. مطالعه تجربه آخرین استراتژی را ارائه می‌دهد [۱۶۶]. این مدل یک پیش‌بینی خطای بسیار قابل اعتماد برای یک برنامه معیوب چهار درصد 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 یک چارچوب پیاده‌سازی شده با استفاده از الگوریتم‌های طبقه‌بندی ML به طور خودکار مدل‌های پیش‌بینی شکست را می‌سازند و عملکرد تکنیک‌های مختلف را برای پروژه‌های فایرفاکس و Netbeans مقایسه می‌کنند. این محاسبه بر اساس مدل هزینه-فایده برای تعیین اهمیت تحقیقات اولیه اضافی است. اهمیت تحقیقات بیشتر در مراحل اولیه در این مدل بر اساس عملکرد احتمالی آن در جلوگیری از خرابی در هزینه نسبی خطاهایی است که با هزینه‌های آن مرتبط هستند [۱۶۷]. پیش‌بینی‌های منطقی دو پارامتر پیش‌بینی عددی کاربرد بهتری را برای برخی اشکال و برنامه‌های خطا نسبت به مجموعه‌ای از خطوط مبنا می‌دهد. این نشان می‌دهد که پیش‌بینی خودکار شکست ممکن است راه‌حلی سودمند برای فعالیت‌های توسعه نیازمندی‌های راهنمایی در محیط‌های آنلاین در طول ایجاد نیازهای آنلاین باشد.

پیش‌بینی نقص بین پروژه (CPDP) حوزه تحقیقاتی است که در آن داده‌های سایر برنامه‌ها می‌توانند توسط یک پروژه نرم‌افزاری با داده‌های محلی ناکافی برای ساخت پیش‌بینی‌کننده‌های خطا استفاده شوند. جزئیات پروژه باید به دقت بررسی شود تا قبل از اجرای محلی به CPDP کمک کند. چندین فیلتر افزایش کارایی CPDP خاص توسط محققان توسعه و معرفی شده است [۱۶۸]. با این حال، تکنیک فیلتر داده، به طور کلی، و به ویژه در CPDP، هنوز نامشخص است. این نشان می‌دهد که تکنیک فیلتر داده به طور چشمگیری کارایی پیش‌بینی خطای بین پروژه‌ای را افزایش می‌دهد و فیلتر انتخابی سلسله مراتبی حتی حیاتی‌تر است. علاوه بر این، پیش‌بینی‌کننده نقص وابسته به داده‌های بین پروژه‌ای ممکن است با استفاده از تکنیک فیلتر داده

صحیح، پیش‌بینی‌کننده آموزش‌دیده با استفاده از داده‌های پروژه داخلی را دور بزند. در واقع، CPDP مورد نیاز است، زیرا از داده‌های منبع/پروژه برچسب‌گذاری شده برای ساخت یک مدل و پیش‌بینی خطا برای یک پروژه هدف استفاده می‌کند [۸۴].

RQ5. کدام الگوریتم یادگیری ماشینی تمایل به دست کم گرفتن و کدام الگوریتم دست کم گرفتن دارد؟

پیش‌بینی خطای نرم‌افزار دقت مبتنی بر: پیش‌بینی اولیه خطاها در نرم‌افزار با استفاده از یک تکنیک پیش‌بینی خاص ممکن است هزینه و تلاش را به حداقل برساند. تکنیک‌های مختلف ML برای پیش‌بینی خطا استفاده شده است، و ثابت شده است که مفید هستند. جدول ۲ دقت پیش‌بینی را که توسط مطالعات اولیه گزارش شده است، ادغام می‌کند. ذکر این نکته دلگرم‌کننده است که از ۱۱۱ مطالعه اولیه تحت پوشش دوربین‌های SLR، نزدیک به ۲۲ مورد دقت مدل را گزارش می‌دهند. SVR و KNN پرکاربردترین معیارهایی هستند که تقریباً ۹۵٪ از مطالعات دقت آنها را گزارش کرده‌اند. زمانی که داده‌ها بر اساس نوع سازمان یا نوع صنعت، به جای محدود شدن به یک مجموعه داده شرکت، در بین گروه‌های همگن مختلف تفکیک می‌شوند، شواهد نسبتاً بیشتری در مورد دقت بهتر وجود دارد. طبقه‌بندی دقت برآورد تلاش را بهبود بخشیده است. ما نتایج تخمین زده شده را در نرم‌افزارهای مختلف در بین الگوریتم‌ها مقایسه کرده ایم. این الگوریتم‌ها می‌توانند در مراحل اولیه چرخه عمر نرم‌افزار مورد استفاده قرار گیرند و می‌توانند به SPM کمک کنند تا قبل از شروع پروژه، تخمین تلاش را به طور موثر انجام دهد. این کار از دست کم گرفتن و دست کم گرفتن کار، از جمله مزایای دیگر، جلوگیری می‌کند. اندازه نرم‌افزار، بهره‌وری، پیچیدگی و پایداری نیاز، عوامل ورودی این مدل‌ها هستند.

| محدودیت‌ها | مزایا | دستیابی به پیش‌بینی | مدل | دیتاست | نوع یادگیری ماشین | مرجع |
|---|--|---------------------|--|--|----------------------------|------|
| در مجموعه داده‌های این مطالعه، عکس‌های فوری تاریخی نداشتند تا مطمئن شوند که ارزش نهایی ویژگی‌های گنجانده شده برای همه وظایف با مقدار آنها قبل از تخصیص به یک توسعه‌دهنده برابر است. | برآورد تلاش خودکار برای تعداد بیشتری از کارها | دقت 88% | مدل ترکیبی از سه مجموعه ویژگی مستقل (۱) ویژگی‌های مبتنی بر ابرداده اولیه، (۲) عنوان و (۳) شرح وظایف نرم‌افزار را استفاده می‌کند. | IBM commercial projects called RQM and RTC | kNN | [73] |
| بررسی حساسیت Dycom به مقادیر پارامترها، یادگیرندگان پایه، ویژگی‌های ورودی و تکنیک‌های تقسیم پروژه‌های CC به بخش‌های مختلف | بهترین استفاده از داده‌های CC، به طوری که می‌تواند مقدار داده‌های WC را کاهش دهد و در عین حال عملکرد را در مقایسه با مدل‌های WC SEE حفظ یا بهبود بخشد. | دقت 66% | DYCOM | KitchenMax CocNasaCoc81 ISBSG2000 ISBSG2001 ISBSG | Logistic linear regression | [48] |
| کارایی سایر تکنیک‌های ML مانند SVM، یادگیری درخت تصمیم و غیره را می‌توان برای تخمین تلاش استفاده کرد. | بر اساس تکنیک‌های ML برای داده‌های غیر کمی و دو فاز انجام می‌شود | دقت 87% | Software Effort Estimation | Data sets University Student Projects developed in 2005) (USP05-FT) and USP05-RQ | Naïve Bayes | [72] |
| بررسی سایر ماشین‌های یادگیری و مجموعه داده‌ها؛ انواع دیگر اندازه اثر، به ویژه موارد غیر پارامتری؛ و سایر اندازه‌های پنجره برای ارزیابی روند یادگیری آنلاین | بررسی کنید که تنظیمات پارامتر تا چه حد بر عملکرد ML در SEE تأثیر می‌گذارد و چه ماشین‌های یادگیری به پارامترهای خود حساس‌تر هستند. | دقت 92% | Software effort estimation | PROMISE Repository | K-NN | [47] |
| نیاز به تحقیقات بیشتر در آینده برای ارزیابی کارایی تکنیک‌های محاسبات نرم در مقایسه با روش‌های رایج رگرسیون آماری، به ویژه در زمینه تخمین تلاش نرم‌افزاری. | مقایسه ای بین تکنیک‌های محاسبات نرم و رگرسیون آماری از نظر مشکل رگرسیون تخمین توسعه نرم‌افزار انجام دهید. | دقت 95% | Software Effort Estimation | NASA93 dataset | SVR | [60] |
| پیاده‌سازی مدلی برای تخمین مقدار نهایی تلاش مورد نیاز در پروژه‌های | بررسی تأثیر طبقه‌بندی در تخمین میزان تلاش مورد نیاز | دقت 95% | Experiments Models | NASA 93 | ANN | [81] |

| محدودیت‌ها | مزایا | دستیابی به پیش بینی | مدل | دیتاست | نوع یادگیری ماشین | مرجع |
|--|--|---------------------------|---|--|----------------------------------|-------|
| جدید، برای برآورد تلاش جزئی در مراحل مختلف فرآیند توسعه پروژه. | در پروژه های توسعه نرم افزار | | | | | |
| مجموعه وسیع‌تری از انواع ویژگی‌ها، مانند ویژگی‌های طبقه‌بندی، را بررسی کنید 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. | به مسئله برآورد هزینه نرم افزار می پردازد و یک رویکرد جایگزین را پیشنهاد می کند که ساختارهای درخت تصمیم گیری قوی را با منطق فازی ترکیب می کند. | دقت 97% | HYBRID Models | ISBSG, COCOMO and DESHARNAIS datasets | Fuzzy logic | [37] |
| فرآیند جمع‌آوری داده‌ها در حوزه سیستم‌های جاسازی شده ممکن است بر جستجوی ویژگی‌های خاص دامنه متمرکز شود، به طوری که محتوای اطلاعاتی ویژگی‌ها غنی‌تر شده و در نتیجه عملکرد پیش‌بینی الگوریتم بهبود می‌یابد. | بررسی همگنی داده های هزینه بر حسب حوزه های کاربردی و تمرکز بر دامنه تعبیه شده | دقت 98% | Data homogeneity | International Software Benchmarking Standards Group (ISBSG) repository | SVR | [109] |
| نسبت به الگوریتم های مقایسه ای عملکرد خوبی ندارد و دلیل آن می تواند عدم وجود داده های منسجم باشد. | روش‌های مبتنی بر مدل از یک فرمول واحد و مقادیر ثابت استفاده می‌کنند و این روش‌ها پاسخگوی پیشرفت‌های روزافزون در زمینه مهندسی نرم‌افزار نیستند. | دقت 91% | Software Cost Estimation (SCE) models | KEMERER, MAXWELL, MIYAZAKI 1, NASA 60, NASA 63, NASA93 | KNN | [107] |
| تمرکز بر تأیید رویکرد پیشنهادی از طریق اثبات مفهوم با سازمان‌های مختلف برای اعتبارسنجی دقت مدل و تنظیم چارچوب استقرار و نگهداری | با استفاده از یافته‌های تحقیقاتی و بهترین شیوه‌های صنعت، شکاف بین نتایج تحقیقات به‌روز و پیاپی‌سازی‌ها در سازمان‌ها را با پیشنهاد رویکردهای استقرار و نگهداری موثر و عملی ML کاهش دهید. | دقت 72% | Software project estimation | ISBSG dataset | SVR | [89] |
| از نظر پارامترهای آموزشی و قوانین ترکیبی که می توان از آنها استفاده کرد. دوم، مطالعات تجربی کاربرد MIAMI در مجموعه داده‌های سایر حوزه‌های داده کاوی باید برای ارزیابی عملکرد آن در یک زمینه کلی‌تر انجام شود. | بهبود دقت پیش‌بینی تلاش نرم‌افزار با تولید مجموعه با استفاده از دو روش انتساب به عنوان عناصر | دقت 92% | Software effort prediction | Kemerer Bank Test equipment DSI Moser, Desharnais Finnish, ISBSG CCCS, Company X | Decision tree | [46] |
| ارائه ابزاری برای پیش‌بینی شکاف‌های شایستگی در پرسنل مدیریت کلیدی با پیش‌بینی سطوح شایستگی برنامه‌ریزی و زمان‌بندی با تمرکز بر گنجاندن انواع دیگر پروژه‌ها به منظور اثبات اینکه چارچوب پیشنهادی می‌تواند هنگام پیش‌بینی شکاف‌های شایستگی در پروژه‌های مختلف تطبیق داده شود. | ارائه ابزاری برای پیش‌بینی شکاف‌های شایستگی در پرسنل مدیریت کلیدی با پیش‌بینی برنامه‌ریزی و زمان‌بندی سطوح شایستگی | دقت 93% | I-Competere | Historical data | Neural networks | [92] |
| مدل پیشنهادی در مجموعه داده‌های جدیدی که برای آزمایش‌ها و تجزیه و تحلیل ما در دسترس قرار می‌گیرند استفاده خواهد شد | در ارتباط با تبدیل ویژگی، انتخاب ویژگی و تکنیک‌های تنظیم پارامتر برای تخمین دقیق تلاش توسعه مورد بررسی قرار گرفت و مدلی به عنوان بخشی از یک سیستم خبره پیشنهاد شد. | دقت 97% | Software development effort estimation | ISBSG datasets | ANN | [94] |
| مجموعه داده‌های بیشتری را از همان دامنه و با استفاده از سایر الگوریتم‌های ماشینی با مقایسه نتایج آنها بگنجانید | ساخت یک مدل پیش‌بینی با استفاده از یادگیری انتقال مبتنی بر آنی از طریق روش اهرم داده | دقت 95% | Source + target | Cross-Project Software Fault Prediction Using Data-Leveraging Technique to Improve Software Quality | Logistic linear regression | [166] |

| محدودیت‌ها | مزایا | دستیابی به پیش بینی | مدل | دیتاست | نوع یادگیری ماشین | مرجع |
|---|--|---------------------|-----------------------------------|---|-------------------|-------|
| پیاده سازی مدل به عنوان یک الگوریتم آنلاین، که با هر نسخه یاد می گیرد | ساخت یک مدل پیش بینی نقص برای یک پروژه بزرگ نرم افزار صنعتی | دقت 90% | Defect Prediction | Real data | Random Forest | [101] |
| نشان دهید که در پروژه هایی که هزینه طبقه بندی اشتباه دقیق ناشناخته است، یک سناریوی محتمل در عمل، مدل های حساس به هزینه با نسبت های هزینه طبقه بندی اشتباه مشابه احتمالاً عملکردی را نشان می دهند که تفاوت قابل توجهی ندارد. | تجزیه و تحلیل مزایای تکنیک هایی که هزینه های طبقه بندی اشتباه را در توسعه مدل های پیش بینی خطای نرم افزار گنجانده اند. | دقت 95% | Misclassification cost-sensitive | 13 data sets | Random forest | [55] |
| تعیین اثربخشی آن در برآورد سایر معیارهای مهم نرم افزار، در مجموعه داده های توسعه نرم افزار خصوصی و عمومی | بکارگیری یک الگوریتم تکاملی برای تولید درخت تصمیم متناسب با مجموعه داده های تلاش نرم افزاری ارائه شده توسط یک شرکت بزرگ فناوری اطلاعات در سراسر جهان | دقت 64% | Evolutionary-based Decision Trees | Company effort data set | Decision tree | [108] |
| کاهش ویژگی های ویژگی با استفاده از چارچوب پیشنهادی | برای اعتبارسنجی معیارهای کد متبوع و انتخاب مجموعه مناسب از معیارها با هدف بهبود عملکرد مدل پیش بینی خطا | دقت 98% | Fault prediction model | Experiments on 45 open source project dataset | ANN | [83] |
| مطالعه اهمیت ویژگی ها برای افزایش عملکرد طبقه بندی کننده ها و دقت مدل SFP | بهینه سازی شعله پروانه دودویی (EBMFO) با نمونه گیری مصنوعی تطبیقی (ADASYN) برای پیش بینی خطاهای نرم افزار | دقت 89% | EBMFO | Several dataset | KNN | [42] |
| بررسی جنبه های اضافی فرآیندهای نرم افزاری و سایر تکنیک های ML برای توسعه راه حل های پیشرفته تر | تمرکز بر روی جنبه رویه فرآیندهای نرم افزار، و فرمول بندی مسئله به عنوان یک کار طبقه بندی دنیای واقعی، که با استفاده از ML حل می شود. | دقت 97% | Defect management (DM) | Quanxi Mi data set | SVM | [86] |
| فرآیند برآورد پروژه و از این پس باعث بهبود کیفیت، بهره وری و پایداری شرکت در فضای صنعتی می شود. | بررسی انتخاب مناسب تکنیک های داده کاوی به منظور برآورد دقیق میزان موفقیت و شکست پروژه ها بر اساس نقص به عنوان یکی از عوامل تعدیل کننده. | دقت 99% | Software Effort Estimation | NASA namely CM1, PC1 and JM1 | Random Forest | [77] |

۵. نتیجه گیری ها

تجزیه و تحلیل ادبی نتیجه گرفت که مطالعه گسترده ای در مدیریت پروژه نرم افزاری بر روی روش های ML انجام شده است. گسترش مشاغل در طول سالیان پیوسته بوده است. الگوریتم های ANN، منطق فازی، ژنتیک و رگرسیون روش های حیاتی ML برای تخمین تلاش خودکار هستند. محاسبه دقیق تلاش یکی از روش های پیشرو در توسعه نرم افزار است. این نرم افزار به طور خاص تحت تأثیر زمان و سختی قرار گرفت. مضامین اساسی ممکن است از کارهای مختلف ML در مدیریت پروژه نرم افزاری استخراج شود. این سرمایه گذاری ها تقریباً به چهار گروه طبقه بندی می شوند: گروه اول شامل بررسی ها و بررسی های مرتبط با مدیریت پروژه نرم افزاری است. گروه دوم مقالاتی را پوشش می دهد که بر مطالعات موردی روش های مدیریت پروژه نرم افزار تمرکز دارند. دسته سوم شامل انتشارات تجربی است که در مدیریت ML، یک نوع ساختار یا مدل معماری استفاده شده است. و گروه نهایی مطالعه مشارکت پژوهشی، تجزیه و تحلیل یک پروژه، فرم یک سازه، یا مدل معماری است. بررسی عمیق این

مقالات به مدیریت پروژه نرم افزار کمک می کند تا رویکردهای ML را برای تعریف و توضیح تهدیدها، مزایا و توصیه ها بررسی کند. با این حال، به دلیل حجم زیاد الگوریتم های ML، الگوریتم های مختلف مطالعه ماشین تحلیل نشده باقی می ماند. سپس دلایل استفاده از SPM خودکار، مشکلات ارزیابی آماده سازی پروژه و فناوری های مهندسی ML بر اساس یافته های ادبیات بررسی می شوند. اگرچه ادبیات مربوط به SPM عملکرد پروژه ها و ضرر را توضیح می دهد، اما سنت طولانی اختلاف نظر در مورد اینکه آیا پیشرفت پروژه باید محاسبه شود وجود دارد. در مورد اینکه چه چیزی توسعه یک پروژه را منعکس می کند و چگونه برآورد می شود، اختلاف نظر وجود دارد. این دستورالعمل ها مشکلات پیش روی یک پروژه نرم افزاری در روش های ML را برطرف می کند و فرصت های کاری را در این بخش باز می کند. تحقیقات هنوز باید تخمین تلاش را بر اساس رویکردهای ML که بر ارزیابی ریسک تمرکز دارد را بررسی کند. عامل دیگر استفاده از روش های فیلتر استاندارد برای به حداقل رساندن مشکل با ایجاد مناطق با ذینفعان مشابه و پیش بینی اینکه آیا ذینفع از موضوع آگاه است یا خیر. این بررسی ادبیات، پاسخ های اولیه ای به سؤالات اساسی در مورد برآورد مدیریت پروژه نرم افزاری که بر اساس ML است، ارائه می دهد.

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. [Google Scholar] [CrossRef]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar] [CrossRef]
5. Hughes, S.W.; Tippet, D.D.; Thomas, W.K. Measuring project success in the construction industry. *Eng. Manag. J.* 2004, 16, 31–37. [Google Scholar] [CrossRef]
6. Project Management Institute. *Guide to the Project Management Body of Knowledge (Pmbok Guide)*; Project Management Institute: Newtown Square, PA, USA, 2013. [Google Scholar]
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. [Google Scholar]
8. Aladwani, A.M. IT project uncertainty, planning and success. *Inf. Technol. People* 2002, 210–226. [Google Scholar] [CrossRef]
9. Cates, G.R.; Mollaghasemi, M. The project assessment by simulation technique. *Eng. Manag. J.* 2007, 19, 3–10. [Google Scholar] [CrossRef]
10. Parsons, V.S. Project performance: How to assess the early stages. *Eng. Manag. J.* 2006, 18, 11–15. [Google Scholar] [CrossRef]
11. Rosenfeld, Y. Root-cause analysis of construction-cost overruns. *J. Constr. Eng. Manag.* 2014, 140, 04013039. [Google Scholar] [CrossRef]
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. [Google Scholar]
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. [Google Scholar] [CrossRef]
14. Tiwana, A.; Keil, M. The one-minute risk assessment tool. *Commun. ACM* 2004, 47, 73–77. [Google Scholar] [CrossRef]
15. Sommerville, I. *Software Engineering*, 9th ed.; Pearson: London, UK, 2011; ISBN 0137035152. [Google Scholar]

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. [Google Scholar] [CrossRef]
17. Malhotra, R.; Chug, A. Software maintainability: Systematic literature review and current trends. *Int. J. Softw. Eng. Knowl. Eng.* 2016, 26, 1221–1253. [Google Scholar] [CrossRef]
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. [Google Scholar]
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. [Google Scholar]
20. Alsolai, H.; Roper, M. A systematic literature review of machine learning techniques for software maintainability prediction. *Inf. Softw. Technol.* 2020, 119, 106214. [Google Scholar] [CrossRef]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar] [CrossRef][Green Version]
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. [Google Scholar]
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 Conference on Software Engineering, Cape Town, South Africa, 2–8 May 2010; Volume 1, pp. 155–164. [Google Scholar]
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. [Google Scholar]
 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. [Google Scholar]
 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. [Google Scholar]
 31. Moharrerri, 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. [Google Scholar]
 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. [Google Scholar]
 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. [Google Scholar]
 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. [Google Scholar] [CrossRef]
 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. [Google Scholar]
 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. [Google Scholar]
 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. [Google Scholar] [CrossRef]
 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. [Google Scholar]

39. Twala, B. Reasoning with Noisy Software Effort Data. *Appl. Artif. Intell.* 2014, 28, 533–554. [Google Scholar]
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. [Google Scholar]
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 (TENSYP)*, Kolkata, India, 7–9 June 2019; pp. 137–142. [Google Scholar]
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. [Google Scholar] [CrossRef]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
46. Twala, B.; Cartwright, M. Ensemble missing data techniques for software effort prediction. *Intell. Data Anal.* 2010, 14, 299–331. [Google Scholar] [CrossRef]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
51. Benala, T.R.; Bandrupalli, 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. [Google Scholar]

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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar] [CrossRef]
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. [Google Scholar]

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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
70. Minku, L.L.; Yao, X. Software effort estimation as a multiobjective learning problem. *ACM Trans. Softw. Eng. Methodol. (TOSEM)* 2013, 22, 1–32. [Google Scholar] [CrossRef][Green Version]
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. [Google Scholar]
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. [Google Scholar]
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 Computing, Data Science & Engineering (ACIT-CSII-BCD)*, Las Vegas, NV, USA, 14 December 2016; pp. 265–270. [Google Scholar]
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. [Google Scholar]

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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar] [CrossRef]
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. [Google Scholar] [CrossRef]
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. [Google Scholar]

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. [Google Scholar] [CrossRef]
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. [Google Scholar] [CrossRef]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar] [CrossRef]
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. [Google Scholar] [CrossRef]
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. [Google Scholar] [CrossRef][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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar] [CrossRef]

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. [Google Scholar] [CrossRef]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar] [CrossRef][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. [Google Scholar]
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. [Google Scholar] [CrossRef]
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. [Google Scholar]
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. [Google Scholar] [CrossRef]
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. [Google Scholar]
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. [Google Scholar] [CrossRef][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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
120. Nassif, A.B.; Azzeh, M.; Idri, A.; Abran, A. Software development effort estimation using regression fuzzy models. *Comput. Intell. Neurosci.* 2019, 2019. [Google Scholar] [CrossRef]
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 Plugins, Honolulu, HI, USA, 28 May 2011; pp. 36–39. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]

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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
129. Pressman, R.S. *Software Engineering: A Practitioner’s Approach*; Palgrave Macmillan: London, UK, 2005. [Google Scholar]
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. [Google Scholar] [CrossRef][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. [Google Scholar] [CrossRef][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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar] [CrossRef]
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. [Google Scholar]
136. Jorgensen, M.; Shepperd, M. A systematic review of software development cost estimation studies. *IEEE Trans. Softw. Eng.* 2006, 33, 33–53. [Google Scholar] [CrossRef][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. [Google Scholar] [CrossRef][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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar] [CrossRef]
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. [Google Scholar]
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. [Google Scholar] [CrossRef]
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. [Google Scholar]
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 Measurement and the 6th International Conference on Software Process and Product Measurement*, Nara, Japan, 4 November 2011; pp. 188–195. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar] [CrossRef]
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. [Google Scholar]

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. [Google Scholar]
150. Larsson, S.; Jansson, M.; Boholm, Å. Expert stakeholders' perception of nanotechnology: Risk, benefit, knowledge, and regulation. *J. Nanoparticle Res.* 2019, 21, 57. [Google Scholar] [CrossRef][Green Version]
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. [Google Scholar] [CrossRef][Green Version]
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. [Google Scholar] [CrossRef]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar] [CrossRef]
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. [Google Scholar]
158. Kappen, T.H.; Vergouwe, Y.; Wolfswinkel, L.V.; Kalkman, C.; Moons, K.; Klei, W.V. Impact of adding therapeutic recommendations to risk assessments from a prediction model for postoperative nausea and vomiting. *Br. J. Anaesth.* 2015, 114, 252–260. [Google Scholar] [CrossRef][Green Version]
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. [Google Scholar] [CrossRef]
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. [Google Scholar] [CrossRef]
161. Yang, M.Q.; Elnitski, L.L. Prediction-based approaches to characterize bidirectional promoters in the mammalian genome. *BMC Genom.* 2008, 9, S2. [Google Scholar] [CrossRef] [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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar]
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. [Google Scholar] [CrossRef]