



بررسی رویکردهای یادگیری ماشین در پشتیبانی و خودکارسازی تصمیمات معماری نرم افزار

امیرحسین نعیمائی کجور¹, مهدی علیرضانژاد²

1- کارشناس ارشد مهندسی فناوری اطلاعات، مؤسسه مارلیک نوشهر

2- استادیار گروه کامپیوتر، دانشگاه آزاد اسلامی واحد فیروزکوه

Amirhossein.naimaei81@gmail.com

چکیده

با افزایش پیچیدگی سامانه های نرم افزاری، تصمیمات معماری نرم افزار نقش کلیدی در تحقق ویژگی های کیفی مانند کارایی، مقیاس پذیری^۱ و نگهداری پذیری^۲ ایفا می کنند، اما رویکردهای سنتی بر پایه دانش ضمنی معماران از نظر سرعت، دقیق و تکرار پذیری با محدودیت مواجهاند. یادگیری ماشین^۳ با تحلیل داده های تاریخی، شناسایی الگوهای پنهان و پیش بینی پیامد تصمیمات، پتانسیل بالایی به عنوان سیستم تصمیم یار در معماری نرم افزار دارد. این مقاله با مرور و تحلیل پژوهش های اخیر در بازه ۱۰ سال گذشته، رویکردهای یادگیری ماشین را در خودکارسازی تصمیمات معماری مورد بررسی قرار می دهد. در این راستا، راهکارهای موجود بر اساس نوع الگوریتم (مانند یادگیری نظارت شده، بدون ناظر و عمیق) و سطح تصمیم گیری (راهبردی، تاکتیکی و عملیاتی) طبقه بندی شده اند. الگوریتم های طبقه بندی، پیش بینی، یادگیری عمیق و بدون ناظر در سطوح راهبردی، تاکتیکی و عملیاتی ارزیابی شده و ابزارهایی مانند ARCHReco، DV8 و Arcan^۴ مقایسه می گردند. نتایج حاکی از بهبود قابل توجه کیفیت تصمیمات است، اما چالش هایی نظیر کمبود داده های استاندارد، تفسیر پذیری (Explainability)^۵ و وابستگی های پیچیده مانع کاربرد صنعتی گستره ده می شوند. این پژوهش با شناسایی شکاف ها، مسیرهایی برای مدل های قابل توضی، دیتا است های مرجع و رویکردهای هیبریدی پیشنهاد می دهد.

کلمات کلیدی: معماری نرم افزار^۶، تصمیمات طراحی معماری^۷، یادگیری ماشین^۸، ارزیابی معماری

مقدمه

اهمیت معماری نرم افزار و نقش آن در کیفیت سیستم

با افزایش پیچیدگی سامانه های نرم افزاری و گسترش نیازمندی های کیفی، معماری نرم افزار به عنوان سطح بالای طراحی سیستم اهمیت ویژه ای یافته است. معماری نرم افزار چارچوبی است که ساختار کلی سیستم، اجزا، روابط میان آنها

¹ software architecture

² Scalability

³ Maintainability

⁴ Machine Learning

⁵ به مجموعه ای از روش ها و تکنیک ها اشاره دارد که هدفشان شفاف سازی و قابل درک کردن تصمیمات و پیش بینی های مدل های یادگیری ماشین برای کاربران انسانی است.

⁶ Software Architecture

⁷ Architectural Design Decisions

⁸ Automation



(Connectors) و الگوهای تعامل را مشخص می‌کند و نقش کلیدی در تحقیق ویژگی‌های کیفی نظری کارایی، مقیاس‌پذیری، قابلیت نگهداری و امنیت دارد. (Bass, Clements, & Kazman, 2012). تصمیمات نادرست در سطح معماری می‌تواند منجر به افزایش پیچیدگی، کاهش انعطاف‌پذیری و بالا رفتن هزینه‌های نگهداری شود، در حالی که تصمیمات مناسب توسعه‌پذیری و پایداری سیستم را تضمین می‌کنند.

تصمیمات معماری و روش‌های سنتی ارزیابی

تصمیمات معماری نرم‌افزار عمده‌تاً بر پایه دانش ضمنی و تجربه معماران اتخاذ می‌شوند و شامل انتخاب سبک معماری، تعیین ساختار مازول‌ها و روابط بین مؤلفه‌ها، و یا انتخاب ابزارها و فناوری‌های مورد استفاده هستند (Kruchten, Capilla, & Architecture Tradeoff ATAM Dueñas, 2009). برای تحلیل این تصمیمات، روش‌های سنتی ارزیابی معماری مانند SAAM (Software Architecture Analysis Method) و (Analysis Method Kazman, Klein, & Clements, 2000; Kazman, Klein, & Bass, 1998) نظام‌مند برای بررسی پیامدهای تصمیمات و ویژگی‌های کیفی سیستم ارائه می‌کنند (Garcia, Popescu, Edwards, & Medvidovic, 2009; Lenarduzzi, Lomio, Saarimäki, & Taibi, 2020).

محدودیت‌های رویکردهای سنتی و نیاز به خودکارسازی

با پیچیده‌تر شدن سیستم‌ها و افزایش وابستگی‌های میان مؤلفه‌ها، رویکردهای سنتی محدودیت‌هایی از نظر مقیاس‌پذیری، دقیقت و تکرارپذیری نشان داده‌اند. این محدودیت‌ها نیاز به روش‌های داده‌محور و خودکار برای پشتیبانی از تصمیمات معماری را ایجاد کرده‌اند. استفاده از داده‌های معماری و تحلیل سیستماتیک می‌تواند خطاهای انسانی را کاهش دهد و کیفیت تصمیمات اتخاذ‌شده را بهبود بخشد. (Clements et al., 2010; Angelov, Trčka, & Greefhorst, 2020)

یادگیری ماشین به عنوان ابزار تصمیم‌یار در معماری نرم‌افزار

یادگیری ماشین به عنوان ابزاری توانمند در این حوزه مطرح شده است. مدل‌های ML "قادرنده الگوهای پنهان در داده‌های تاریخی را شناسایی کنند، پیامدهای تصمیمات معماری را پیش‌بینی کنند و توصیه‌های بهینه برای معماران ارائه دهند (Ahmed, Capretz, & Campbell, 2016; Guo, Karim, & Menzies, 2019). کاربردهای مهم یادگیری ماشین در معماری شامل شناسایی بوهای معماری، پیش‌بینی کیفیت سیستم، تحلیل وابستگی‌ها و ارائه پیشنهادهای تصمیم‌یار است (Bogner, Wagner, & Zimmermann, 2021). با این حال، محدودیت‌هایی نظیر کمبود داده‌های معتبر، پیچیدگی مدل‌سازی وابستگی‌ها، تفاوت دامنه‌های کاربردی و قابلیت توضیح‌پذیری مدل‌ها همچنان چالش‌برانگیز هستند (Ammar, Stamelos, & Gomaa, 2018; Angelov et al., 2020). گراف (GNN)¹ و در سال‌های بسیار اخیر، مدل‌های زبانی بزرگ (LLMs)² متمایل شده‌اند که قابلیت تحلیل داده‌های متند عصبی معماری را نیز فراهم می‌آورند.

هدف پژوهش و دامنه مطالعه

¹ بوهای معماری: الگوهای ضعیف یا مشکلات ساختاری در طراحی سیستم هستند که معمولاً نشان‌دهنده ریسک‌های توسعه، نگهداری یا کارایی در آینده‌اند مانند (God Component) یا Cyclic Dependency Component

² Machine learning

³ Graph Neural Networks

⁴ Large Language Models



پژوهش‌های کاربردی در علوم برق، کامپیوتر و مهندسی پزشکی

هدف این مقاله ارائه یک تصویر جامع و تحلیلی از کاربرد یادگیری ماشین در خودکارسازی تصمیمات معماری نرمافزار است. این پژوهش با دسته‌بندی روش‌ها، مدل‌ها و ابزارهای موجود، نقش آن‌ها را در پشتیبانی از تصمیمات معماری تحلیل کرده و با برجسته‌سازی چالش‌های فعلی، مسیرهایی را برای تحقیقات آینده ترسیم می‌کند. بدین ترتیب، چارچوبی روش برای تحقیقات آتی و توسعه ابزارهای عملی در این حوزه فراهم می‌آید.

برای ارائه تصویری جامع از حوزه یادگیری ماشین در معماری نرمافزار، این مقاله بخش‌های مختلفی را شامل می‌شود. ابتدا شاخص‌ها و معیارهای کیفیت معماری مورد بررسی قرار گرفته‌اند که شامل قابلیت نگهدارش، کارایی، قابلیت اعتماد، مازولار بودن (Modularity)، معیارهای اتصال و انسجام (Coupling & Cohesion Metrics)، احتمال انتشار تغییر (Change Propagation Probability) و شناسایی بوهای معماری هستند.

سطوح تصمیم‌گیری معماری نرمافزار و نقش یادگیری ماشین در خودکارسازی تصمیمات (1)

میزان خودکارسازی	نقش یادگیری ماشین	نوع تصمیمات معماری	سطح تصمیم‌گیری
پایین - تصمیم نهایی با معمار (Human-in-the-loop)	تحلیل تجربیات پروژه‌های مشابه، پیشنهاد گزینه‌های مناسب بر اساس ویژگی‌های کیفی و دامنه سیستم	انتخاب سیک، الگوهای کلان (Monolithic, Microservices, Event-Driven)	راهبردی
متوسط - پیشنهاد خودکار، تأیید انسانی	شناسایی بوهای معماری، پیش‌بینی پیامدهای تصمیمات، پیشنهاد اصلاحات ساختاری	طراحی ساختار مازول‌ها، تعیین وابستگی‌ها، انتخاب الگوهای	تactیکی
بالا - قابلیت خودکارسازی بیشتر	تحلیل خودکار وابستگی‌ها، پیشنهاد refactoring	Refactoring، معماری، تنظیم پارامترها، بهینه‌سازی ارتباط مؤلفه‌ها	عملیاتی
بالا - تحلیل و ارزیابی خودکار	پیش‌بینی کیفیت، امتیازدهی معماری، شناسایی ریسک‌ها	سنچش کیفیت‌هایی مانند Performance، Maintainability، Modularity	ارزیابی معماری
متوسط - پشتیبان تصمیم	استخراج الگو از تصمیمات گذشته، پیشنهاد تصمیمات مشابه	ثبت و بازیابی تصمیمات معماری و پیامدها	مستندسازی تصمیمات (ADR / AK)

کارهای مرتبط

پژوهش‌های مرتبط با تصمیمات معماری نرمافزار در ابتدا عمدتاً بر روش‌های ارزیابی معماری متتمرکز بوده‌اند. روش‌هایی نظیر SAAM (Software Architecture Analysis Method) و ATAM (Architecture Tradeoff Analysis Method) با هدف تحلیل ویژگی‌های کیفی سیستم و بررسی پیامدهای تصمیمات معماری توسعه یافته‌اند (Kazman et al., 1998; Kazman & Bass, 1994). این رویکردها اگرچه چارچوب‌های نظاممندی برای ارزیابی معماری فراهم می‌کنند، اما اجرای آن‌ها به صورت دستی انجام شده و به دانش و تجربه متخصصان وابستگی بالایی دارد.

در ادامه، با افزایش پیچیدگی سیستم‌های بزرگ‌مقیاس، رویکردهای نیمه‌خودکار مبتنی بر قوانین و مدل‌های هیورستیک معرفی شدند که تلاش داشتند بخشی از فرایند تحلیل معماری را استاندارد سازی کنند. با وجود بهبود نسبی در سرعت و

تکرار پذیری تحلیل‌ها، این روش‌ها همچنان نیازمند تعریف قواعد اولیه توسط خبرگان بوده و از انعطاف‌پذیری محدودی برخوردار بودند.

با گسترش کاربرد یادگیری ماشین در مهندسی نرم‌افزار، پژوهشگران به امکان تحلیل خودکار ساختار و کیفیت معماری توجه نشان دادند. مطالعات اولیه عمده‌ای بر شناسایی بوهای معماری و پیش‌بینی ریسک‌های ساختاری مرکز بودند (Garcia et al., 2009). در ادامه، مدل‌هایی پیشنهاد شدند که قادر به پیش‌بینی پیامد تصمیمات معماری بر ویژگی‌هایی نظری کارایی و قابلیت نگهداری بودند. در سال‌های اخیر نیز رویکردهای پیشرفته‌تری مانند سیستم‌های توصیه‌گر معماری و مدل‌های یادگیری عمیق برای تحلیل وابستگی‌های پیچیده معماری مطرح شده‌اند.

با وجود پیشرفته‌های انجام‌شده، کاربرد یادگیری ماشین در تصمیم‌سازی معماری نرم‌افزار همچنان با چالش‌هایی نظری کمبود داده‌های معماری معتبر، تنوع دامنه‌های کاربردی و مسئله قابلیت اعتماد و تفسیر پذیری نتایج مواجه است. این شکاف پژوهشی نشان می‌دهد که اگرچه رویکردهای مبتنی بر یادگیری ماشین پتانسیل بالایی در پشتیبانی از تصمیمات معماری دارند، اما نیازمند تحلیل‌های جامع‌تر و چارچوب‌های منسجم‌تری هستند؛ موضوعی که این مقاله بر آن تمرکز دارد.

این شکاف پژوهشی نشان می‌دهد که نیازمند یک چارچوب منسجم برای تحلیل یکپارچه سطوح تصمیم‌گیری و رویکردهای یادگیری ماشین هستیم، موضوعی که در بخش بعدی تحت عنوان «چارچوب بررسی و دسته‌بندی» ارائه شده است.

چارچوب بررسی و دسته‌بندی

به منظور ارائه تحلیلی جامع و ساختاریافته از وضعیت کنونی کاربرد یادگیری ماشین در مهندسی نرم‌افزار، در این پژوهش ادبیات موضوع بر اساس یک چارچوب سه وجهی مورد بررسی قرار گرفته است: «سطح تصمیم‌گیری»، «تکنیک‌های یادگیری ماشین» و «شاخص‌های کیفیت». این دسته‌بندی به درک دقیق‌تر رابطه میان الگوریتم‌ها و نیازهای معماری کمک می‌کند.

مقایسه ابزارهای خودکارسازی تصمیمات معماری مبتنی بر یادگیری ماشین(2)

محدودیت	نقاط قوت	نوع داده ورودی	نوع تحلیل	ابزار
محدود به تحلیل ساختار کد	تشخیص بوهای معماری	وابستگی کد	Shallow ML	Arcan
عدم پشتیبانی از دامنه‌های متعدد	تحلیل معماری دقیق	گراف ساختاری	Rules + ML	DV8
نیاز به داده زیاد برای آموزش	ارائه پیشنهادهای معماری	متريک‌ها + گراف	ML-Based	ARCHReco

سطح تصمیم‌گیری معماری

تصمیمات معماری نرم‌افزار دارای ماهیت یکسانی نیستند و در سطوح مختلفی از انتزاع رخ می‌دهند. طبق بررسی‌های انجام شده، این تصمیمات به سه سطح اصلی تقسیم می‌شوند:



1. سطح راهبردی (Strategic Level): شامل تصمیمات کلان و بنیادین است که تأثیر طولانی مدت بر سیستم دارند، مانند انتخاب سبک معماری) مثلاً گذار از یکپارچه به میکروسرویس ها. در این سطح، خودکارسازی کامل دشوار است و یادگیری ماشین عمدها نقش پشتیبان تصمیم (Decision Support) را ایفا می کند.
2. سطح تاکتیکی (Tactical Level): بر طراحی ساختار داخلی ماژولها و الگوهای طراحی تمرکز دارد. سیستم های هوشمند در این سطح می توانند وابستگی های بین ماژول ها را تحلیل کرده و ناهنجاری های طراحی را شناسایی کنند.
3. سطح عملیاتی (Operational Level): شامل فعالیت های نگهداری و بهینه سازی مداوم است، نظیر بازسازی کد و تنظیم پارامتر های سیستم. این سطح بالاترین پتانسیل را برای خودکارسازی کامل دارد.

دسته بندی الگوریتم های یادگیری ماشین

رویکردهای فنی مورد استفاده در پژوهش های اخیر، بر اساس ماهیت الگوریتم و نوع مسئله به چهار دسته کلی تقسیم شده اند:

دسته بندی الگوریتم های یادگیری ماشین و کاربرد آن ها در تصمیمات معماری (3)

هدف کاربردی	مثال ها	نوع الگوریتم
دسته بندی بوهای معماری، تشخیص	Random Forest, SVM, Decision Trees	طبقه بندی
پیش بینی کیفیت، پیش بینی ریسک	Linear Regression, Gradient Boosting	پیش بینی
تحلیل پیچیده ساختار، مدل سازی تعاملات مؤلفه ها	LSTM, GNN (Graph Neural Networks)	یادگیری عمیق
کشف الگوهای ناشناخته در معماری	K-Means, DBSCAN	بدون ناظر

1. الگوریتم های طبقه بندی: شامل روش هایی نظیر Random Forest، SVM و درخت تصمیم است. کاربرد اصلی این دسته، شناسایی مشکلات ساختاری یا همان «بوهای معماری» است که در آن مدل با دریافت ویژگی های کد، وجود یا عدم وجود مشکل را تعیین می کند.
2. الگوریتم های پیش بینی: از روش هایی مانند رگرسیون خطی و Gradient Boosting برای تخمین مقادیر پیوسته استفاده می شود. پیش بینی شاخص های کیفی نظری میزان نگهداری پذیری یا زمان پاسخ دهی سیستم در آینده، در این دسته قرار می گیرد.
3. یادگیری عمیق (Deep Learning): با توجه به ساختار گراف گونه و غیرخطی معماری نرم افزار، مدل های پیشرفته تر مانند شبکه های عصبی گراف و LSTM برای تحلیل وابستگی های پیچیده و تعاملات میان مؤلفه ها به کار گرفته شده اند.
4. یادگیری نظارت نشده: روش هایی مانند K-Means و DBSCAN که برای کشف الگوهای پنهان و خوشه بندی ماژول ها بدون نیاز به داده های بر جسب دار استفاده می شوند.



شاخص های ارزیابی کیفیت

نهایتاً، اثربخشی این رویکردها بر اساس بهبود شاخص های کیفی استاندارد سنجیده می شود. تمرکز اصلی این پژوهش بر شاخص هایی نظیر قابلیت نگهداری، کارایی، مازولار بودن و قابلیت اطمینان است که مستقیماً تحت تأثیر تصمیمات معماری قرار دارند.

طبقه بندی رویکردهای تصمیم سازی معماری (4)

رویکرد	توضیح	سطح خودکارسازی	وابستگی به خبره
روش های دستی ارزیابی معماری	ارزیابی تصمیمات معماری بر اساس سناریوها و قضاؤت انسانی	بسیار کم	بسیار زیاد
روش های مبتنی بر	استفاده از قوانین از پیش تعریف شده برای تحلیل معماری	کم تا متوسط	زیاد
مدل های یادگیری ماشین کلاسیک	یادگیری الگوها از داده های معماری برای پیش بینی پیامد تصمیمات	متوسط	متوسط
مدل های یادگیری عمیق	تحلیل وابستگی های پیچیده معماری با شبکه های عصبی عمیق	متوجه تا زیاد	کم
سیستم های توصیه گر معماری	پیشنهاد تصمیمات معماری بر اساس داده و تجربه پیشین	زیاد	کم

شاخص های کیفیت معماری مورد استفاده در پژوهش های مبتنی بر یادگیری ماشین (5)

شاخص	توضیح
نگهداری پذیری	توانایی سیستم برای نگهداری، اصلاح و افزودن قابلیت ها.
کارایی	توانایی سیستم در پاسخ دهی سریع و مصرف بهینه منابع
قابلیت اطمینان	میزان پایداری سیستم و مقاومت آن در برابر خطای
مازو لار بودن	سطح تفکیک و استقلال مؤلفه ها
وابستگی و انسجام	شاخص های وابستگی بین مؤلفه ها و انسجام داخلی آن ها
احتمال گسترش تغییرات	احتمال تأثیر تغییرات یک مؤلفه بر سایر مؤلفه ها
نشان های مشکل زا (Architecture Smells)	God Component و Cyclic Dependency.



مقایسه روش ها از منظر یادگیری ماشین (6)

محدودیت ها	مزایا	کاربرد اصلی	نوع مدل یادگیری ماشین	نوع روش
نیاز به داده برچسب خورده	ساده، قابل تفسیر	کشف مشکلات ساختاری	'Classification (SVM) (Decision Tree)	شناسایی بیوهوای معماری
دقت وابسته به کیفیت داده	تحلیل پیامد تصمیمات	پیش‌بینی کارایی و نگهداشت	Regression Models	پیش‌بینی ویژگی‌های کیفی
تفسیرپذیری پایین	دقیق بالا در سیستم‌های بزرگ	مدل‌سازی روابط پیچیده	Deep Learning (ANN, GNN)	تحلیل وابستگی معماری
تعیین‌پذیری محدود	پشتیبانی تصمیم	پیشنهاد تصمیم معماری	Recommendation Systems	سیستم‌های توصیه‌گر
پیچیدگی محاسباتی	یافتن راهکارهای بهینه	بهینه‌سازی ساختار معماری	Hybrid ML + Optimization	تحلیل تکاملی معماری

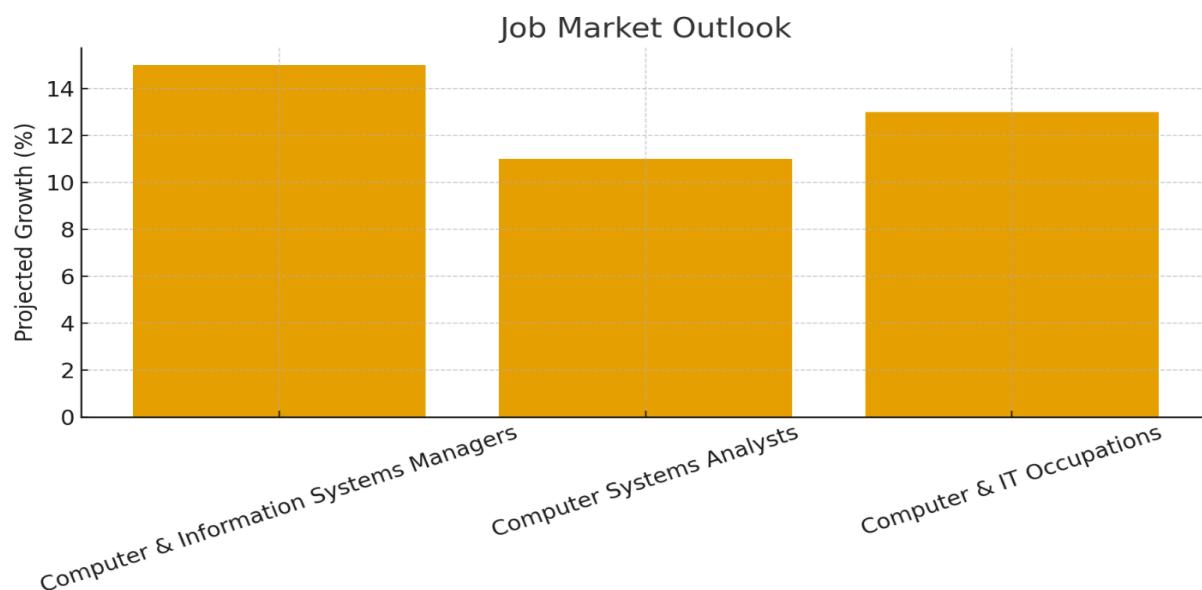
(7) شکاف‌های پژوهشی و چالش‌ها

فرصت پژوهشی	پیامد	توضیح	چالش
تولید دیتاست‌های مرجع	کاهش دقت مدل‌ها	نیوود دیتاست‌های استاندارد معماری	کمبود داده‌های معماری
Domain-Aware مدل‌های	ضعف تعیین‌پذیری	تفاوت معماری در حوزه‌های مختلف	تنوع دامنه‌ها
استفاده از شبکه‌های عصبی گراف	مدل‌سازی دشوار	روابط غیرخطی میان اجزا	وابستگی‌های پیچیده
Explainable AI (XAI)	کاهش اعتماد معماران	خروجی‌های غیرشفاف یادگیری ماشین	تفسیرپذیری مدل‌ها
Runtime-aware Models	افت دقت پیش‌بینی	پویایی سیستم‌ها	تغییرپذیری در زمان اجرا

اهمیت کاربردی فناوری‌های تصمیم‌گیری ماشین در بازار کار

بررسی داده‌های بازار کار جهانی نشان می‌دهد که متخصصان حوزه فناوری‌های تصمیم‌گیری ماشین و تحلیل داده در سازمان‌ها با رشد تقاضای قابل توجه مواجه هستند (U.S. Bureau of Labor Statistics, 2025). برای مثال، مدیران سیستم‌های کامپیوتری و اطلاعاتی، تحلیل‌گران سیستم‌ها و تحلیل‌گران داده‌ها، رشد پیش‌بینی شده‌ای بین ۱۰ تا ۱۵ درصد دارند. این روند، اهمیت یادگیری تکنیک‌ها یادگیری ماشین و درخت تصمیم برای مهندسان نرم‌افزار و کارشناسان سیستم‌های اطلاعاتی را برجسته می‌کند و نشان می‌دهد پژوهش‌های این حوزه، علاوه بر ارزش علمی، دارای کاربرد صنعتی واقعی نیز هستند.

روند رشد پیش‌بینی شده مشاغل مرتبط با فناوری‌های تصمیم‌یار در سطح جهانی :



جمع‌بندی

نتایج این مطالعه مربوطی نشان می‌دهد که ادغام یادگیری ماشین با تصمیم‌سازی معماري نرم‌افزار می‌تواند به شکل چشم‌گیری فرآیند تحلیل و ارزیابی معماري را داده محور، سریع و تکرارپذیر کند. مدل‌های یادگیری ماشین در حوزه‌هایی مانند شناسایی بوهای معماري ، پیش‌بینی شاخص‌های کیفیت (مانند کارایی و نگهداری‌پذیری) و پیشنهاد اصلاحات ساختاري، توانسته‌اند معمaran نرم‌افزار را در اتخاذ تصمیمات دقیق‌تر و آگاهانه‌تر پشتیبانی کنند. در عین حال، مشخص شد که سطح خودکارسازی در سطوح مختلف تصمیم‌گیری معماري (راهبردي، تاكتيكي و عملياتي) يكسان نیست و در تصمیمات عملياتي و ارزیابي معماري، پتانسیل خودکارسازی بالاتر است.

با وجود اين پیشرفت‌ها، اين حوزه هنوز به بلوغ كامل نرسیده است. چالش‌هایی مانند کمبود مجموعه‌داده‌های استاندارد معماري، تفاوت دامنه‌های کاربردی، پیچیدگی وابستگی‌های چندلايه بین مؤلفه‌ها و مسئله تفسیرپذیری مدل‌ها باعث شده که پذيرش صنعتی اين راهكارها محدود باقی بماند. بنابراين، یادگیری ماشین در معماري نرم‌افزار در حال حاضر بيش از آن که جايگزين كامل تصمیم‌گیری انساني باشد، نقش يك تصمیم‌یار هوشمند را ايقا می‌کند و نيازمند ترکيب با دانش تخصصي معمaran است.

پیشنهاد برای تحقیقات آینده

بر اساس شکاف‌ها و چالش‌های شناسایي شده، چند محور کلیدی برای تحقیقات آینده قابل طرح است:

- طراحی و انتشار مجموعه‌داده‌های مرجع معماري که شامل مدل‌های معماري، گراف‌های وابستگی، شاخص‌های کیفیت و برچسب‌های مرتبط با بوهای معماري باشند، تا آموزش و ارزیابي مدل‌های یادگیری ماشین با دقت بيشتری انجام شود.

- توسيعه مدل‌های قابل توضیح که بتوانند علاوه بر ارائه پیش‌بینی یا پیشنهاد، دلیل تصمیم خود را به زبان قابل درک برای معمaran نرم‌افزار بيان کنند و اعتمادپذیری سیستم‌های تصمیم‌یار را افزایش دهند.



- ترکیب دانش دامنه و قوانین معماری با روش های داده محور در قالب رویکردهای هیبریدی (Hybrid Approaches) تا نقاط قوت هر دو نوع روش در کنار هم استفاده شود.
- گسترش سیستم های توصیه گر معماری برای سناریوهای واقعی شامل معماری های مبتنی بر میکروسرویس ها ، Event-Driven و Serverless و محیط های مقیاس پذیر ابری.
 - بهره گیری از مدل های گراف ممحور مانند (GNNs) برای مدل سازی دقیق تر ساختار معماری و تحلیل تعاملات پیچیده بین مؤلفه ها در سطح گراف وابستگی (Dependency Graph).
 - استفاده از هوش مصنوعی مولد و مدل های بزرگ در معماری: جدیدترین و هیجان انگیزترین مسیر تحقیقاتی، بهره گیری از مدل های زبانی بزرگ برای پشتیبانی از تصمیمات معماری است. برخلاف الگوریتم های سنتی یادگیری ماشین که تنها می توانند داده های ساختار یافته (مانند گراف وابستگی یا متغیر های عددی) را تحلیل کنند، مدل های زبانی بزرگ می توانند متن را پردازش و تولید کنند.
 - 1. تولید و نگهداری خود کار مستندات معماری: مدل های زبانی بزرگ می توانند شرح تصمیمات معماری را از طریق تحلیل کامنت های کد، درخواست های و مستندات موجود استخراج کرده و به صورت خود کار، مستندات به روز را تولید کنند.
 - 2. تولید طرح های پیشنهادی اولیه : بر اساس نیازهای متنی سیستم (مانند User Stories یا Stakeholder Requirements)، مدل های زبانی بزرگ می توانند ساختار اولیه معماری، لیست سرویس های مورد نیاز میکروسرویس ها یا حتی کدهای زیر ساختی را پیشنهاد دهند.
 - 3. ترجمه چالش های کیفی: این مدل ها می توانند چالش های کیفی و مبهم معمار (مانند «سیستم باید منعطف باشد») را به نیازمندی های فنی و کمی تبدیل کنند تا سایر ابزارهای یادگیری ماشین بتوانند آن ها را تحلیل نمایند.

مراجع

1. Ahmed, F., Capretz, L. F., & Campbell, P. (2016). Software architecture decision-making: A machine learning perspective. IEEE International Conference on Software Engineering.
2. Ammar, H., Stamelos, I., & Gomaa, H. (2018). Software engineering using artificial intelligence techniques: Current state and open problems. ACM Computing Surveys, 51(4), 1–42.
3. Angelov, S., Trčka, N., & Greefhorst, D. (2020). A classification of software architecture optimization approaches. Journal of Systems and Software, 160, 110453.
4. Bass, L., Clements, P., & Kazman, R. (2012). Software architecture in practice (3rd ed.). Addison-Wesley.
5. Chen, L., Ali Babar, M., & Zhang, H. (2010). Towards an evidence-based understanding of evolving software architectures with architecture smells. In ICSE Workshops.
6. Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Nord, R., & Stafford, J. (2010). Documenting software architectures: Views and beyond (2nd ed.). Addison-Wesley.
7. Clements, P., Kazman, R., & Klein, M. (2002). Evaluating software architectures: Methods and case studies. Addison-Wesley.



8. Garcia, J., Popescu, D., Edwards, G., & Medvidovic, N. (2009). Identifying architectural bad smells. In Proceedings of the 13th European Conference on Software Maintenance and Reengineering (pp. 255–258). IEEE.
9. Guo, J., Karim, R., & Menzies, T. (2019). Learning software performance models using AI-based approaches. ACM Journal on Autonomous and Adaptive Systems.
10. Hassan, A. E. (2009). Predicting change propagation in software architectures. IEEE Software.
11. Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In 5th International Conference on Learning Representations (ICLR 2017).
12. Kruchten, P., Capilla, R., & Dueñas, J. C. (2009). The decision view's role in software architecture practice. IEEE Software, 26(2), 36–42.
13. Kazman, R., Klein, M., & Clements, P. (2000). ATAM: Method for architecture evaluation. In Proceedings of the 26th International Conference on Software Engineering (pp. 143–152). IEEE.
14. Lenarduzzi, V., Lomio, F., Saarimäki, N., & Taibi, D. (2020). Software architecture smells: A systematic literature review. Journal of Systems and Software, 164.
15. Macia, I., García, F., & Piattini, M. (2012). Architectural smells in software systems. Journal of Systems and Software, 85(1), 139–156.
16. Bogner, J., Wagner, S., & Zimmermann, O. (2021). Automatically extracting architectural smells using machine learning. Empirical Software Engineering.
17. Xu, J., Liang, P., & Avgeriou, P. (2019). Machine learning for architecture recovery: A review. Information and Software Technology.
18. Zimmermann, O., Zdun, U., Gschwind, T., & Leymann, F. (2010). Reusable architectural decision models for enterprise application development. Springer Software and Systems Modeling.