



بررسی رویکردهای یادگیری ماشین در پشتیبانی و خودکار سازی تصمیمات معماری نرم افزار

امیرحسین نعیمائی کجور¹، مهدی علیرضائزاد²

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، دقت و تکرار پذیری با محدودیت مواجه اند. 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 خودکار سازی 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 DV8، Arcan و ARCHReco مقایسه می گردند. نتایج حاکی از بهبود قابل توجه کیفیت تصمیمات است، اما چالش هایی نظیر کمبود داده های استاندارد، تفسیر پذیری (Explainability)^۴ و وابستگی های پیچیده مانع کاربرد صنعتی گسترده می شوند. این پژوهش با شناسایی شکاف ها، مسیرهایی برای مدل های قابل توضیح، دیتاست های مرجع و رویکردهای هیبریدی پیشنهاد می دهد.

کلمات کلیدی: معماری نرم افزار^۵، تصمیمات طراحی معماری^۶، یادگیری ماشین، خودکار سازی^۷، 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 می کند و نقش کلیدی در تحقق ویژگی های کیفی نظیر کارایی ، مقیاس پذیری ، قابلیت نگهداشت و امنیت دارد. (Bass, Clements, & Kazman, 2012) تصمیمات نادرست در سطح معماری می تواند منجر به افزایش پیچیدگی، کاهش انعطاف پذیری و بالا رفتن هزینه های نگهداشت شود، در حالی که تصمیمات مناسب توسعه پذیری و پایداری سیستم را تضمین می کنند.

تصمیمات معماری و روش های سنتی ارزیابی

تصمیمات معماری نرم افزار عمدتاً بر پایه دانش ضمنی و تجربه معماران اتخاذ می شوند و شامل انتخاب سبک معماری ، تعیین ساختار ماژول ها و روابط بین مؤلفه ها، و یا انتخاب ابزارها و فناوری های مورد استفاده هستند (Kruchten, Capilla, & Dueñas, 2009). برای تحلیل این تصمیمات، روش های سنتی ارزیابی معماری مانند (ATAM Architecture Tradeoff Analysis Method) و (SAAM Software Architecture Analysis Method) توسعه یافته اند که چارچوب هایی نظام مند برای بررسی پیامدهای تصمیمات و ویژگی های کیفی سیستم ارائه می کنند (Kazman, Klein, & Clements, 1998; Kazman, Klein, & Bass, 2000). با این حال، اجرای این روش ها به تجربه متخصصان وابسته بوده و قابلیت مقیاس پذیری و تکرارپذیری محدودی دارند. علاوه بر این، مشکلات ساختاری شناخته شده به عنوان بوهای معماری (Architectural Smells¹) می توانند کیفیت سیستم را کاهش دهند و شناسایی آن ها به صورت دستی چالش برانگیز است (Garcia, Popescu, Edwards, & Medvidovic, 2009; Lenarduzzi, Lomio, Saarimäki, & Taibi, 2020).

محدودیت های رویکردهای سنتی و نیاز به خودکارسازی

با پیچیده تر شدن سیستم ها و افزایش وابستگی های میان مؤلفه ها، رویکردهای سنتی محدودیت هایی از نظر مقیاس پذیری، دقت و تکرارپذیری نشان داده اند. این محدودیت ها نیاز به روش های داده محور و خودکار برای پشتیبانی از تصمیمات معماری را ایجاد کرده اند. استفاده از داده های معماری و تحلیل سیستماتیک می تواند خطاهای انسانی را کاهش دهد و کیفیت تصمیمات اتخاذ شده را بهبود بخشد. (Clements et al., 2010; Angelov, Trčka, & Greefhorst, 2020)

یادگیری ماشین به عنوان ابزار تصمیم یار در معماری نرم افزار

یادگیری ماشین به عنوان ابزاری توانمند در این حوزه مطرح شده است. مدل های ML² قادرند الگوهای پنهان در داده های تاریخی را شناسایی کنند، پیامدهای تصمیمات معماری را پیش بینی کنند و توصیه های بهینه برای معماران ارائه دهند (Ahmed, Capretz, & Campbell, 2016; Guo, Karim, & Menzies, 2019). معماری شامل شناسایی بوهای معماری، پیش بینی کیفیت سیستم، تحلیل وابستگی ها و ارائه پیشنهادهای تصمیم یار است (Bogner, Wagner, & Zimmermann, 2021). مدل سازی وابستگی ها، تفاوت دامنه های کاربردی و قابلیت توضیح پذیری مدل ها همچنان چالش برانگیز هستند (Ammar, Stamelos, & Gomaa, 2018; Angelov et al., 2020). گراف (GNN³) و در سال های بسیار اخیر، مدل های زبانی بزرگ (LLMs⁴) متمایل شده اند که قابلیت تحلیل داده های متنی معماری را نیز فراهم می آورند.

هدف پژوهش و دامنه مطالعه

¹ بوهای معماری: الگوهای ضعیف یا مشکلات ساختاری در طراحی سیستم هستند که معمولاً نشان دهنده ریسک های توسعه، نگهداری یا کارایی در آینده اند مانند (God Component یا Cyclic Dependency)

² Machine learning

³ Graph Neural Networks

⁴ Large Language Models



هدف این مقاله ارائه یک تصویر جامع و تحلیلی از کاربرد یادگیری ماشین در خودکارسازی تصمیمات معماری نرم افزار است. این پژوهش با دسته بندی روش ها، مدل ها و ابزارهای موجود، نقش آن ها را در پشتیبانی از تصمیمات معماری تحلیل کرده و با برجسته سازی چالش های فعلی، مسیرهایی را برای تحقیقات آینده ترسیم می کند. بدین ترتیب، چارچوبی روشن برای تحقیقات آتی و توسعه ابزارهای عملی در این حوزه فراهم می آید.

برای ارائه تصویری جامع از حوزه یادگیری ماشین در معماری نرم افزار، این مقاله بخش های مختلفی را شامل می شود. ابتدا شاخص ها و معیارهای کیفیت معماری مورد بررسی قرار گرفته اند که شامل قابلیت نگهداشت ، کارایی، قابلیت اعتماد، ماژولار بودن (Modularity) ، معیارهای اتصال و انسجام (Coupling & Cohesion Metrics)، احتمال انتشار تغییر (Change Propagation Probability) و شناسایی بوهای معماری هستند.

سطوح تصمیم گیری معماری نرم افزار و نقش یادگیری ماشین در خودکارسازی تصمیمات (1)

سطح تصمیم گیری	نوع تصمیمات معماری	نقش یادگیری ماشین	میزان خودکارسازی
راهبردی	انتخاب سبک ، الگوهای کلان (Monolithic, Microservices, Event-Driven)	تحلیل تجربیات پروژه های مشابه، پیشنهاد گزینه های مناسب بر اساس ویژگی های کیفی و دامنه سیستم	پایین - تصمیم نهایی با معمار (Human-in-the-loop)
تاکتیکی	طراحی ساختار ماژول ها، تعیین وابستگی ها، انتخاب الگوهای	شناسایی بوهای معماری ، پیش بینی پیامدهای تصمیمات، پیشنهاد اصلاحات ساختاری	متوسط - پیشنهاد خودکار، تأیید انسانی
عملیاتی	Refactoring معماری، تنظیم پارامترها، بهینه سازی ارتباط مؤلفه ها	تحلیل خودکار وابستگی ها، پیشنهاد refactoring، پیش بینی تأثیر تغییرات	بالا - قابلیت خودکارسازی بیشتر
ارزیابی معماری	سنجش کیفیت هایی مانند Performance, Maintainability, Modularity	پیش بینی کیفیت، امتیازدهی معماری، شناسایی ریسک ها	بالا - تحلیل و ارزیابی خودکار
مستندسازی تصمیمات (ADR / AK)	ثبت و بازیابی تصمیمات معماری و پیامدها	استخراج الگو از تصمیمات گذشته، پیشنهاد تصمیمات مشابه	متوسط - پشتیبان تصمیم

کارهای مرتبط

پژوهش های مرتبط با تصمیمات معماری نرم افزار در ابتدا عمدتاً بر روش های ارزیابی معماری متمرکز بوده اند. روش هایی نظیر ATAM (Architecture Tradeoff Analysis Method) و SAAM (Software Architecture Analysis Method) با هدف تحلیل ویژگی های کیفی سیستم و بررسی پیامدهای تصمیمات معماری توسعه یافته اند (Kazman et al., 1998; Kazman & Bass, 1994). این رویکردها اگرچه چارچوب های نظام مندی برای ارزیابی معماری فراهم می کنند، اما اجرای آن ها به صورت دستی انجام شده و به دانش و تجربه متخصصان وابستگی بالایی دارد.

در ادامه، با افزایش پیچیدگی سیستم های بزرگ مقیاس، رویکردهای نیمه خودکار مبتنی بر قوانین و مدل های هیورستیک معرفی شدند که تلاش داشتند بخشی از فرایند تحلیل معماری را استانداردسازی کنند. با وجود بهبود نسبی در سرعت و



تکرارپذیری تحلیل ها، این روش ها همچنان نیازمند تعریف قواعد اولیه توسط خبرگان بوده و از انعطاف پذیری محدودی برخوردار بودند.

با گسترش کاربرد یادگیری ماشین در مهندسی نرم افزار، پژوهشگران به امکان تحلیل خودکار ساختار و کیفیت معماری توجه نشان دادند. مطالعات اولیه عمدتاً بر شناسایی بوهای معماری و پیش بینی ریسک های ساختاری متمرکز بودند (Garcia et al., 2009). در ادامه، مدل هایی پیشنهاد شدند که قادر به پیش بینی پیامد تصمیمات معماری بر ویژگی های نظیر کارایی و قابلیت نگهداشت بودند. در سال های اخیر نیز رویکردهای پیشرفته تری مانند سیستم های توصیه گر معماری و مدل های یادگیری عمیق برای تحلیل وابستگی های پیچیده معماری مطرح شده اند.

با وجود پیشرفت های انجام شده، کاربرد یادگیری ماشین در تصمیم سازی معماری نرم افزار همچنان با چالش هایی نظیر کمبود داده های معماری معتبر، تنوع دامنه های کاربردی و مسئله قابلیت اعتماد و تفسیرپذیری نتایج مواجه است. این شکاف پژوهشی نشان می دهد که اگرچه رویکردهای مبتنی بر یادگیری ماشین پتانسیل بالایی در پشتیبانی از تصمیمات معماری دارند، اما نیازمند تحلیل های جامع تر و چارچوب های منسجم تری هستند؛ موضوعی که این مقاله بر آن تمرکز دارد.

این شکاف پژوهشی نشان می دهد که نیازمند یک چارچوب منسجم برای تحلیل یکپارچه سطوح تصمیم گیری و رویکردهای یادگیری ماشین هستیم، موضوعی که در بخش بعدی تحت عنوان "چارچوب بررسی و دسته بندی" ارائه شده است

چارچوب بررسی و دسته بندی

به منظور ارائه تحلیلی جامع و ساختاریافته از وضعیت کنونی کاربرد یادگیری ماشین در مهندسی نرم افزار، در این پژوهش ادبیات موضوع بر اساس یک چارچوب سه وجهی مورد بررسی قرار گرفته است: «سطوح تصمیم گیری»، «تکنیک های یادگیری ماشین» و «شاخص های کیفیت». این دسته بندی به درک دقیق تر رابطه میان الگوریتم ها و نیازهای معماری کمک می کند.

مقایسه ابزارهای خودکار سازی تصمیمات معماری مبتنی بر یادگیری ماشین (2)

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

سطوح تصمیم گیری معماری

تصمیمات معماری نرم افزار دارای ماهیت یکسانی نیستند و در سطوح مختلفی از انتزاع رخ می دهند. طبق بررسی های انجام شده، این تصمیمات به سه سطح اصلی تقسیم می شوند:



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 عمدتاً نقش پشتیبان تصمیم (Decision Support) را ایفا می کند.
2. سطح تاکتیکی (Tactical Level): بر طراحی ساختار داخلی ماژول ها 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 (Refactoring) و تنظیم پارامترهای 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)

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

1. الگوریتم های طبقه بندی: شامل روش هایی نظیر SVM ، Random Forest 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 نظیر 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 برچسب 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 ، ماژولار بودن و قابلیت اطمینان است که مستقیماً تحت تأثیر تصمیمات معماری قرار دارند.

طبقه بندی رویکردهای تصمیم سازی معماری (4)

رویکرد	توضیح	سطح خودکار سازی	وابستگی به خبره
روش های دستی ارزیابی معماری	ارزیابی تصمیمات معماری بر اساس سناریوها و قضاوت 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	متوسط 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 خطا
ماژولار 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 مشکلات ساختاری مانند



مقایسه روش ها از منظر یادگیری ماشین (6)

نوع روش	نوع مدل یادگیری ماشین	کاربرد اصلی	مزایا	محدودیت ها
شناسایی بدبوه های معماری	SVM (Classification) (Decision Tree)	کشف مشکلات ساختاری	ساده، قابل تفسیر	نیاز به داده برچسب خورده
پیش بینی ویژگی های کیفی	Regression Models	پیش بینی کارایی و نگهداشت	تحلیل پیامد تصمیمات	دقت وابسته به کیفیت داده
تحلیل وابستگی معماری	Deep Learning (ANN, GNN)	مدل سازی روابط پیچیده	دقت بالا در سیستم های بزرگ	تفسیر پذیری پایین
سیستم های توصیه گر	Recommendation Systems	پیشنهاد تصمیم معماری	پشتیبانی تصمیم	تعمیم پذیری محدود
تحلیل تکاملی معماری	Hybrid ML + Optimization	بهینه سازی ساختار معماری	یافتن راهکار های بهینه	پیچیدگی محاسباتی

شکاف های پژوهشی و چالش ها (7)

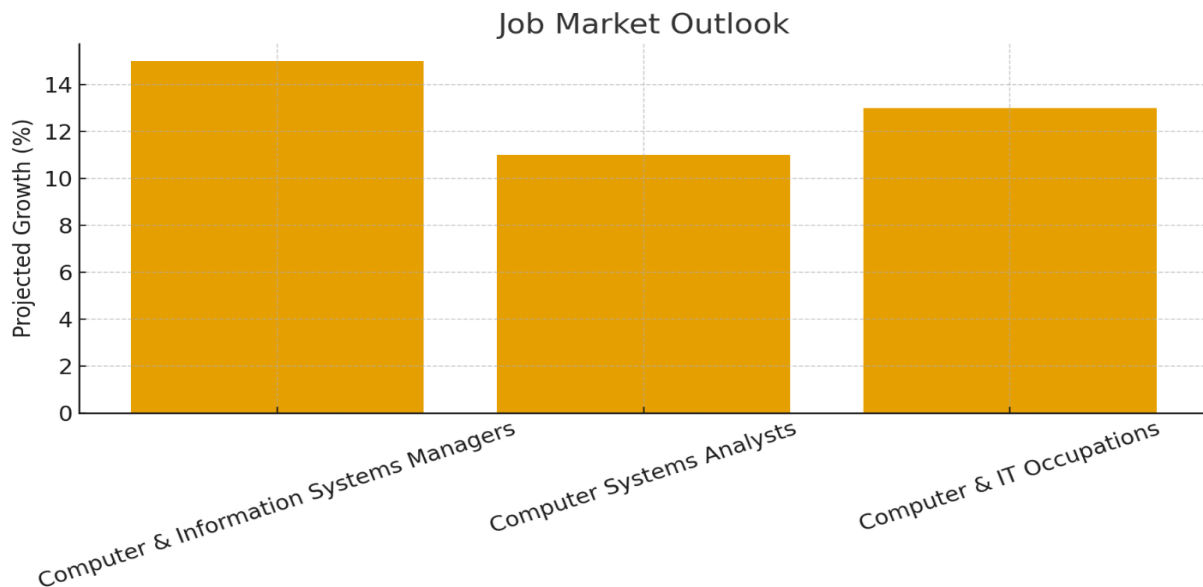
چالش	توضیح	پیامد	فرصت پژوهشی
کمبود داده های معماری	نبود دیتاست های استاندارد معماری	کاهش دقت مدل ها	تولید دیتاست های مرجع
تنوع دامنه ها	تفاوت معماری در حوزه های مختلف	ضعف تعمیم پذیری	مدل های Domain-Aware
وابستگی های پیچیده	روابط غیر خطی میان اجزا	مدل سازی دشوار	استفاده از شبکه های عصبی گراف
تفسیر پذیری مدل ها	خروجی های غیر شفاف یادگیری ماشین	کاهش اعتماد معماران	Explainable AI (XAI)
تغییر پذیری در زمان اجرا	پویایی سیستم ها	افت دقت پیش بینی	Runtime-aware Models

اهمیت کاربردی فناوری های تصمیم یار و یادگیری ماشین در بازار کار

بررسی داده های بازار کار جهانی نشان می دهد که متخصصان حوزه فناوری های تصمیم یار، یادگیری ماشین و تحلیل داده در سازمان ها با رشد تقاضای قابل توجه مواجه هستند (U.S. Bureau of Labor Statistics, 2025). برای مثال، مدیران سیستم های کامپیوتری و اطلاعاتی، تحلیل گران سیستم ها و تحلیل گران داده ها، رشد پیش بینی شده ای بین ۱۰ تا ۱۵ درصد دارند. این روند، اهمیت یادگیری تکنیک ها یادگیری ماشین و درخت تصمیم برای معماران نرم افزار و کارشناسان سیستم های اطلاعاتی را برجسته می کند و نشان می دهد پژوهش های این حوزه، علاوه بر ارزش علمی، دارای کاربرد صنعتی واقعی نیز هستند.



روند رشد پیش بینی شده مشاغل مرتبط با فناوری های تصمیم یار در سطح جهانی :



جمع بندی

نتایج این مطالعه مروری نشان می دهد که ادغام یادگیری ماشین با تصمیم سازی معماری نرم افزار می تواند به شکل چشم گیری فرآیند تحلیل و ارزیابی معماری را داده محور، سریع و تکرار پذیر کند. مدل های یادگیری ماشین در حوزه هایی مانند شناسایی بوهای معماری ، پیش بینی شاخص های کیفیت (مانند کارایی و نگهداری پذیری) و پیشنهاد اصلاحات ساختاری، توانسته اند معماران نرم افزار را در اتخاذ تصمیمات دقیق تر و آگاهانه تر پشتیبانی کنند. در عین حال، مشخص شد که سطح خودکارسازی در سطوح مختلف تصمیم گیری معماری (راهبردی، تاکتیکی و عملیاتی) یکسان نیست و در تصمیمات عملیاتی و ارزیابی معماری، پتانسیل خودکارسازی بالاتر است.

با وجود این پیشرفت ها، این حوزه هنوز به بلوغ کامل نرسیده است. چالش هایی مانند کمبود مجموعه داده های استاندارد معماری، تفاوت دامنه های کاربردی، پیچیدگی وابستگی های چند لایه بین مؤلفه ها و مسئله تفسیر پذیری مدل ها باعث شده که پذیرش صنعتی این راهکارها محدود باقی بماند. بنابراین، یادگیری ماشین در معماری نرم افزار در حال حاضر بیش از آن که جایگزین کامل تصمیم گیری انسانی باشد، نقش یک تصمیم یار هوشمند را ایفا می کند و نیازمند ترکیب با دانش تخصصی معماران است.

پیشنهاد برای تحقیقات آینده

بر اساس شکاف ها و چالش های شناسایی شده، چند محور کلیدی برای تحقیقات آینده قابل طرح است:

- طراحی و انتشار مجموعه داده های مرجع معماری که شامل مدل های معماری، گراف های وابستگی، شاخص های کیفیت و برچسب های مرتبط با بوهای معماری باشند، تا آموزش و ارزیابی مدل های یادگیری ماشین با دقت بیشتری انجام شود.
- توسعه مدل های قابل توضیح که بتوانند علاوه بر ارائه پیش بینی یا پیشنهاد، دلیل تصمیم خود را به زبان قابل درک برای معماران نرم افزار بیان کنند و اعتماد پذیری سیستم های تصمیم یار 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 ، Serverless و Event-Driven و محیط های مقیاس پذیر ابری.
 - بهره گیری از مدل های گراف محور 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 زبانی 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 ساختاریافته (مانند گراف وابستگی یا متغیرهای عددی) را تحلیل کنند، مدل های زبانی بزرگ می توانند متن را پردازش و تولید کنند.
1. تولید و نگهداری خودکار 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 خودکار، 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 زیرساختی 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.