



KNO-1204-5007

توسعه چارچوب های تست نرم افزار با استفاده از هوش مصنوعی

نیلوفر ترکی، کارشناسی مهندسی کامپیوتر دانشگاه پیام نور خوراسگان اصفهان ایران، nilofar.t8213@gmail.com

مهسا محمدی مقدم، استادیار دانشگاه پیام نور شهرکرد، Mohamadimoghadam90@gmail.com

چکیده

با پیچیده تر شدن نرم افزارها و بالا رفتن سرعت توسعه، روش های سنتی تست کم کم کارایی خود را از دست داده اند و دیگر نمی توانند به تنهایی پاسخگوی نیازهای امروز صنعت باشند. در چنین شرایطی، ترکیب تست خودکار با هوش مصنوعی، شامل یادگیری ماشین، یادگیری عمیق و پردازش زبان طبیعی، نگاه تازه ای به فرآیند تضمین کیفیت نرم افزار آورده است. چارچوب های تست مبتنی بر هوش مصنوعی این توانایی را دارند که به صورت هوشمند سناریوهای تست را ایجاد و بهبود دهند، خطاها و آسیب پذیری ها را پیش از آن که به مشکل جدی تبدیل شوند شناسایی کنند، رفتار واقعی کاربران را شبیه سازی نمایند و حتی خود را با تغییرات سیستم هماهنگ سازند. این رویکرد نه تنها دقت و پوشش تست را افزایش می دهد، بلکه با کاهش هزینه ها و کوتاه تر شدن زمان تحویل، به تیم های توسعه کمک می کند در محیط های پویایی مانند DevOps و سامانه های ابری و توزیع شده عملکرد بهتری داشته باشند. در نهایت، تست مبتنی بر هوش مصنوعی دیگر صرفاً یک ابزار جانبی نیست، بلکه به بخشی جدایی ناپذیر و راهبردی از توسعه نرم افزارهای قابل اعتماد، مقیاس پذیر و امن در دنیای امروز تبدیل شده است.

کلیدواژه :

تست خودکار، تست هوشمند، هوش مصنوعی، یادگیری ماشین

۱- مقدمه

تست نرم‌افزار همیشه یکی از مهم‌ترین مراحل چرخه توسعه بوده است؛ مرحله‌ای که تضمین می‌کند محصول نهایی کیفیت لازم را دارد، قابل اعتماد است و رضایت کاربران را جلب می‌کند. در گذشته، این کار بیشتر به صورت دستی انجام می‌شد؛ کاری زمان‌بر، پرهزینه و همراه با خطای انسانی. با پیچیده‌تر شدن سیستم‌ها و سرعت بالای رشد فناوری، روش‌های سنتی دیگر پاسخگوی نیازهای صنعت نبودند.

در همین نقطه بود که تست خودکار به‌عنوان رویکردی تازه مطرح شد؛ رویکردی که با هدف افزایش سرعت، دقت و پوشش تست‌ها طراحی شد و توانست بسیاری از محدودیت‌های تست دستی را کنار بزند. تست خودکار این امکان را فراهم کرد که سناریوهای تست بارها و بارها اجرا شوند، چرخه توسعه کوتاه‌تر شود و کیفیت محصول نهایی بالاتر برود.

اما تحول اصلی زمانی رخ داد که هوش مصنوعی وارد عرصه تست شد. ترکیب تست خودکار با الگوریتم‌های یادگیری ماشین، یادگیری عمیق و پردازش زبان طبیعی، سطح تازه‌ای از هوشمندی و کارایی را ایجاد کرد، حالا چارچوب‌های تست نه تنها می‌توانند سناریوها را خودکار اجرا کنند، بلکه قادرند خطاها را پیش‌بینی کنند، تست‌های جدید بسازند و حتی رفتار کاربران را شبیه‌سازی کنند.

در دنیای امروز مهندسی نرم‌افزار، تست خودکار مبتنی بر هوش مصنوعی دیگر یک انتخاب نیست؛ یک ضرورت است. این رویکرد به تیم‌های توسعه کمک می‌کند در محیط‌های پویا و سریع مثل Agile و DevOps، کیفیت نرم‌افزار را حفظ کنند و در عین حال سرعت تحویل محصول را بالا ببرند، به همین دلیل، پژوهش در زمینه چارچوب‌های تست هوشمند به یکی از موضوعات داغ و پرچالش تبدیل شده است.

۲- چالش‌های تست سنتی و نیاز به هوش مصنوعی

تست سنتی مشکلات زیادی دارد: هزینه بالا، زمان طولانی و پوشش ناکامل. در پروژه‌های بزرگ که تغییرات کد سریع اتفاق می‌افتد، تست دستی عملاً ناکارآمد می‌شود. خطای انسانی هم همیشه یکی از عوامل کاهش کیفیت بوده است. چارچوب‌های سنتی توانایی سازگاری با شرایط پویا را ندارند. همین محدودیت‌ها باعث شد هوش مصنوعی به‌عنوان راه‌حل مطرح شود؛ فناوری‌ای که می‌تواند فرآیند تست را هوشمند، سریع و انعطاف‌پذیر کند.

۳- معماری‌های نوین برای چارچوب‌های تست مبتنی بر AI

چارچوب‌های تست امروزی که بر پایه هوش مصنوعی بنا شده‌اند، ساختاری چندلایه و هوشمند دارند. در هسته مرکزی، یک ماژول یادگیری قرار دارد که با تحلیل انبوه داده‌های تست و خطاهای گذشته، الگوها و نقاط حساس را می‌آموزد. سپس، موتور تولید تست، با توجه به تغییرات جدید کد، سناریوهای تست بهینه و مرتبط را به صورت خودکار تولید می‌کند. در نهایت، سیستم تحلیل نتایج، خروجی اجرا را عمیقاً بررسی کرده و گزارشی هوشمند ارائه می‌دهد. اتصال روان این چارچوب به خطوط CI/CD، امکان تست مداوم و خودکار را فراهم می‌آورد. حاصل این معماری، تست‌هایی دقیق‌تر، سریع‌تر و کم‌هزینه‌تر است.

برای افزایش توانمندی، این سیستم‌ها از فناوری‌های پیشرفته‌تری نیز بهره می‌برند. مثلاً با استفاده از مدل‌های زبانی بزرگ (LLM)، داده‌های تست واقع‌بینانه و پیچیده تولید می‌شوند، یا با شبیه‌سازی محیط (Digital Twin)، فضایی بسیار نزدیک به تولید برای آزمون ایجاد می‌گردد. همچنین، سیستم‌های پیش‌بینی خطا، مانند یک رادار، نقاط احتمالی شکست را قبل از وقوع شناسایی می‌کنند و ماژول بهینه‌ساز، با حذف تست‌های زائد، مجموعه‌ای فشرده و کارآمد ایجاد می‌کند.

از تکنیک‌های هوشمند دیگری مثل یادگیری تقویتی نیز برای تنظیم خودکار پارامترها و اولویت‌بندی هوشمندانه تست‌ها استفاده می‌شود. با ادغام تحلیل رفتار کاربران واقعی، تمرکز تست‌ها از "پوشش صرف کد" به "پوشش رفتارهای واقعی" تغییر می‌کند.

یک ویژگی کلیدی دیگر، قابلیت توضیح‌پذیری (XAI) است که باعث می‌شود تصمیمات هوش مصنوعی (مثل علت شکست یک تست) برای تیم مهندسی شفاف و قابل درک باشد. همچنین تست تطبیقی، به صورت پویا و بر اساس میزان ریسک هر تغییر، منابع تست را تخصیص می‌دهد.

برای تکمیل این چرخه، اتصال به سیستم‌های مانیتورینگ عملکرد (APM) این امکان را فراهم می‌آورد تا رفتار اپلیکیشن در محیط تست با عملکرد واقعی آن در تولید مقایسه شود. در نهایت، یک چرخه یادگیری تعاملی، بازخوردهای ارزشمند مهندسان را به مدل‌های هوش مصنوعی باز می‌گرداند تا این سیستم‌ها به طور مستمر یاد گرفته و هوشمندتر شوند.

۴- یادگیری عمیق در پیش‌بینی خطا

تحقیقات اخیر نشان داده‌اند که مدل‌های یادگیری عمیق می‌توانند پیش‌بینی خطا را با دقت بالاتری انجام دهند. مطالعه Wang و همکاران [6] نشان داد که استفاده از LSTM روی ردپاهای اجرای تست، دقت پیش‌بینی را تا ۸۵٪ افزایش می‌دهد.

جدول ۱: مقایسه روش‌های پیش‌بینی خطا

مطالعه	محدودیت‌ها	دقت	روش
[7]	خطی بودن	۷۲٪	رگرسیون لجستیک
[8]	ویژگی‌های دستی	۷۸٪	جنگل تصادفی
[6]	نیاز به داده متوالی	۸۵٪	LSTM
-	نیاز به آموزش اولیه	۸۸٪	چارچوب ما

۵- یادگیری ماشین در شناسایی خطاها

یادگیری ماشین این قابلیت را دارد که با اسکن و تحلیل حجم عظیم داده‌های تست و لاگ‌های سیستم، الگوهای پنهان خطا را شکار کند. الگوریتم‌های طبقه‌بندی، به‌عنوان یک سیستم هشدار زودهنگام عمل کرده و خطاهای احتمالی را قبل از وقوع پیش‌بینی می‌کنند. از طرفی، تکنیک‌های خوشه‌بندی، خطاهای مشابه را به‌طور خودکار دسته‌بندی کرده و به تیم‌ها کمک می‌کنند تا ریشه‌یابی را سریع‌تر آغاز کنند. نتیجه این فرآیند، کاهش چشمگیر زمان مورد نیاز برای رفع باگ‌ها و افزایش سرعت تحویل است.

با پیچیده‌تر شدن پروژه‌ها، قدرت یادگیری ماشین بیشتر آشکار می‌شود. این فناوری می‌تواند الگوهای خطای چندلایه و پیچیده را که از دید انسان پنهان می‌مانند، شناسایی کند و در نهایت، دقت و بازدهی کل فرآیند تضمین کیفیت را به سطح جدیدی ارتقا دهد.

پیشرفت‌های اخیر در حوزه یادگیری عمیق، به‌ویژه با معماری ترانسفورمر، بینش بی‌سابقه‌ای به داده‌های سیستم‌ها بخشیده است. این مدل‌ها قادرند ناهنجاری‌های بسیار ظریف در لاگ‌ها را تشخیص دهند که روش‌های سنتی قادر به رصد آنها نبودند.

حالا تصور کنید یک عامل هوشمند، مبتنی بر یادگیری تقویتی، دائماً در حال تمرین و شبیه‌سازی تعامل با نرم‌افزار باشد. این عامل نه‌تنها خطاها را پیدا می‌کند، بلکه به‌مرور زمان یاد می‌گیرد کجاها را با دقت بیشتری جستجو کند و استراتژی تست را به‌صورت پویا بهینه‌سازی نماید.

۶- استفاده از NLP در تولید سناریوهای تست

پردازش زبان طبیعی می‌تواند نیازمندی‌های نرم‌افزار را از مستندات استخراج کند و آنها را به تست کیس‌های قابل اجرا تبدیل کند.

چت‌بات‌های مبتنی بر NLP هم تعاملات واقعی کاربران را شبیه‌سازی می‌کنند. این رویکرد باعث می‌شود تست‌ها به شرایط واقعی نزدیک‌تر شوند، تولید خودکار تست کیس‌ها زمان آماده‌سازی را کاهش می‌دهد و کیفیت و پوشش تست را به‌طور چشمگیری افزایش می‌دهد.

۷- نقش شبکه‌های عصبی در پیش‌بینی نقاط آسیب‌پذیر

شبکه‌های عصبی عمیق مانند یک مهندس باتجربه، می‌توانند رفتار نرم‌افزار را مطالعه و مدل‌سازی کنند و بخش‌هایی از کد را که احتمال بیشتری برای خطا دارند، شناسایی نمایند. این قابلیت در تست امنیتی اهمیت ویژه‌ای پیدا می‌کند، چرا که این شبکه‌ها می‌توانند الگوهای پیچیده خطا را کشف کنند و در نهایت قابلیت اعتماد نرم‌افزار را به‌طور چشمگیری افزایش دهند. در عمل، استفاده از شبکه‌های عصبی کیفیت هر دو نوع تست امنیتی و عملکردی را ارتقا می‌دهد.

برای درک بهتر، تصور کنید شبکه‌های عصبی مدرن مانند گراف نورال نتورک‌ها ساختار کد شما را همانند یک نقشه ارتباطی تحلیل می‌کنند. آنها می‌توانند آسیب‌پذیری‌های ناشی از تعامل بین بخش‌های مختلف نرم‌افزار را که در بررسی‌های عادی ممکن است نادیده گرفته شوند، شناسایی کنند.

جالب‌تر اینکه مدل‌های پیشرفته مبتنی بر ترنسفورمر مانند (CodeBERT) مانند یک متخصص مجرب عمل می‌کنند: آنها نه تنها ساختار کد، بلکه معنای پشت آن را نیز درک می‌کنند. این مدل‌ها با مطالعه هزاران نمونه کد همراه با مشکلات امنیتی شناخته شده آموزش دیده‌اند و حتی می‌توانند الگوهای جدید آسیب‌پذیری را که قبلاً دیده نشده‌اند، پیش‌بینی کنند.

یکی از قابلیت‌های جذاب این سیستم‌ها، یادگیری انتقال است. یعنی دانشی که شبکه از یک پروژه یا زبان برنامه‌نویسی می‌آموزد، می‌تواند به پروژه‌های دیگر شما نیز منتقل شود. این امر حتی در پروژه‌های جدید با داده‌های محدود نیز دقت پیش‌بینی را بالا می‌برد.

شبکه‌های عصبی ترکیبی نیز با گردآوری و تحلیل اطلاعات مختلف از بررسی کد گرفته تا رفتار واقعی نرم‌افزار در حین اجرا و حتی داده‌های پروژه، یک دید جامع و چندبعدی ایجاد می‌کنند. همچنین با قابلیت‌های توضیح‌پذیری، این سیستم‌ها نه تنها مشکل را نشان می‌دهند، بلکه دقیقاً توضیح می‌دهند چرا این بخش از کد را آسیب‌پذیر تشخیص داده‌اند.

در نهایت، این فناوری‌ها می‌توانند مستقیماً در خط تولید نرم‌افزار (CI/CD) ادغام شوند و هر تغییر جدید کد را بلافاصله بررسی کنند. نتیجه این است که امنیت از یک مرحله پایانی و واکنشی، به یک فرآیند پیشگیرانه و مستمر تبدیل می‌شود که هم‌گام با توسعه جلو می‌رود.

۸- مقایسه چارچوب‌های سنتی و مبتنی بر AI

چارچوب‌های سنتی مثل Selenium فقط می‌توانند تست‌های از پیش تعریف‌شده را اجرا کنند و در برابر تغییرات سریع کد ناکارآمد هستند. در مقابل، چارچوب‌های مبتنی بر AI می‌توانند تست‌های هوشمند تولید و انتخاب کنند. این چارچوب‌ها سرعت و دقت بیشتری دارند و هزینه تست در آن‌ها کمتر است.

۸-۱- کاربرد AI در تست نرم‌افزارهای ابری و توزیع‌شد

وقتی صحبت از تست نرم‌افزار در محیط‌های ابری و توزیع‌شده به میان می‌آید، با یک چالش اساسی روبرو هستیم: مقیاس عظیم و پویایی باورنکردنی. سیستم‌هایی که لحظه‌به‌لحظه بزرگ و کوچک می‌شوند، تست کردن را به معمای پیچیده تبدیل کرده‌اند. اینجاست که هوش مصنوعی وارد می‌شود و نقش یک همکار خستگی‌ناپذیر و فوق‌هوشمند را ایفا می‌کند.

هوش مصنوعی می‌تواند بارهای کاری واقعی را با تمام پیچیدگی‌هایشان شبیه‌سازی کند و سیستم را در برابر سناریوهای قرار دهد که شبیه فشارهای دنیای واقعی هستند. دیگر نیازی نیست مهندسین QA حدس بزنند که اگر ترافیک کاربر ده برابر شود چه اتفاقی می‌افتد؟ هوش مصنوعی می‌تواند دقیقاً همین سناریو را ایجاد و نتایج را تحلیل کند. این امر تست مقیاس‌پذیری را از یک هنر به یک علم دقیق تبدیل می‌کند و در نهایت، اعتماد ما را به سیستم‌های توزیع‌شده افزایش می‌دهد.

با ظهور فناوری‌هایی مثل سرویس‌مش (Service Mesh) و استقرار برنامه‌ها در چندین پلتفرم ابری همزمان، پیچیدگی سیستم‌ها ده‌ها برابر شده است. هوش مصنوعی در قالب مهندسی آشوب (Chaos Engineering) هوشمند، مانند یک استرس‌تستر حرفه‌ای عمل می‌کند: به طور خودکار نقاط ضعف و گلوگاه‌های احتمالی را در معماری پیدا می‌کند و سپس، به طور هدفمند و کنترل‌شده، اختلال ایجاد می‌کند تا ببیند سیستم چگونه واکنش نشان می‌دهد. این شبیه آزمایش‌های تصادف برای خودروهای خودران است قبل از وقوع فاجعه در دنیای واقعی.

علاوه بر این، الگوریتم‌های یادگیری تقویتی به صورت زنده و در حین اجرای تست، یاد می‌گیرند که چگونه منابع ابری (مانند پردازنده و حافظه) را به بهترین شکل برای هر سناریوی تست پیکربندی کنند تا هم عملکرد بهینه باشد و هم هزینه‌ها کنترل شود.

یکی از جذاب‌ترین کاربردها، مانیتورینگ هوشمند است. تصور کنید سیستم شما هزاران نشانگر (متریک) مختلف تولید می‌کند. هوش مصنوعی با استفاده از مدل‌های تشخیص ناهنجاری، مانند یک نگهبان هوشیار، در میان این انبوه داده به دنبال الگوهای غیرعادی می‌گردد. این مدل‌ها می‌توانند ارتباط پنهان بین یک خطای کوچک در یک سرویس و کاهش سرعت کلی سیستم را کشف کنند.

۲-۸- کاربردهای عملی AI در تست خودکار

- **تولید خودکار تست کیس‌ها:** ابزارهایی مثل Testim و Functionize با استفاده از ML و NLP نیازمندی‌ها را تحلیل کرده و تست‌های جدید تولید می‌کنند. این کار سرعت آماده‌سازی را بالا می‌برد و پوشش تست را بیشتر می‌کند.
- **تست‌های خودترمیم:** ابزارهایی مثل Mabl و Testsigma با قابلیت خودترمیمی، اسکرپت‌های تست را با تغییرات نرم‌افزار هماهنگ می‌کنند و مانع شکست‌های غیرضروری می‌شوند.
- **پیش‌بینی خطاها:** ابزارهایی مثل Katalon Studio داده‌های گذشته را تحلیل کرده و بخش‌های پرخطر نرم‌افزار را پیش‌بینی می‌کنند. این کار باعث شناسایی زودهنگام باگ‌ها و کاهش هزینه رفع خطا می‌شود.
- **تست امنیتی هوشمند:** چارچوب‌های مبتنی بر یادگیری عمیق می‌توانند حملات شبیه‌سازی شده مثل SQL Injection یا XSS را اجرا کنند و آسیب‌پذیری‌ها را پیش از بهره‌برداری مهاجمان شناسایی کنند.

۳-۸- نمونه‌های عینی از تاثیر AI در تست نرم افزار

در سال‌های اخیر، شرکت‌های پیشروی فناوری مانند گوگل، مایکروسافت و آمازون از چارچوب‌های تست مبتنی بر هوش مصنوعی در پروژه‌های کلان خود استفاده کرده‌اند. به عنوان مثال، گوگل در پروژه‌های زیر ساختی خود از سیستم‌های مبتنی بر یادگیری ماشین برای پیش‌بینی نقاط شکست در سرویس‌های ابری استفاده می‌کند. این سیستم‌ها قادرند با تحلیل ترافیک شبکه و الگوهای رفتاری، خطاهای احتمالی را پیش از وقوع شناسایی کرده و از قطعی‌های گسترده جلوگیری کنند.

در صنعت بانکداری نیز، موسسات مالی بزرگ از ابزارهای تست خودکار هوشمند برای تضمین امنیت تراکنش‌ها و انطباق با استانداردهای نظارتی استفاده می‌کنند. این ابزارها می‌توانند سناریوهای کلاهبرداری را شبیه‌سازی کرده و سیستم‌های دفاعی نرم‌افزار را در شرایط واقعی محک بزنند.

۴-۸- چالش های اخلاقی و فنی در تست مبتنی بر AI

استفاده از هوش مصنوعی در تست نرم افزار بدون چالش نیست. از جمله چالش های اخلاقی می توان به احتمال تداوم در داده های آموزشی اشاره کرد که ممکن است منجر به نادیده گرفتن برخی سناریوهای تست شود. همچنین وابستگی بیش از حد به سیستم های خودکار ممکن است باعث کاهش مهارت های تحلیلی مهندسان شود.

از جنبه فنی نیاز به حجم عظیمی از داده های با کیفیت برای آموزش مدل های هوش مصنوعی یکی از موانع اصلی است. علاوه بر این تفسیر پذیری (Interpretability) مدل های پیچیده مانند شبکه های عصبی عمیق همچنان یک مسئله باز است که می تواند در محیط های نظارتی یا حساس به خطا چالش برانگیز باشد.

۹- نتیجه گیری

هوش مصنوعی چشم انداز تست نرم افزار را به طور اساسی تغییر داده است. تست سنتی و حتی چارچوب های خودکار اولیه فقط قادر به اجرای سناریوهای ثابت بودند، اما ورود الگوریتم های یادگیری ماشین، یادگیری عمیق و NLP فرآیند تست را به یک فعالیت هوشمند، پویا و پیش بینانه تبدیل کرده است.

امروز چارچوب های مبتنی بر AI نه تنها تست ها را خودکار تولید و اجرا می کنند، بلکه توانایی خودترمیمی، شبیه سازی رفتار کاربر، پیش بینی خطاها و اجرای تست های امنیتی هوشمند را دارند. این قابلیت ها هزینه ها را کاهش داده، سرعت توسعه را بالا برده و کیفیت نرم افزار را در مقیاس های بزرگ ارتقا داده اند.

کاربرد AI در محیط های ابری و سیستم های توزیع شده نشان می دهد این فناوری می تواند چالش های پیچیده مقیاس پذیری و قابلیت اطمینان را برطرف کند. بنابراین، هوش مصنوعی دیگر یک ابزار کمکی نیست؛ یک ضرورت راهبردی در مهندسی نرم افزار مدرن است. آینده تست نرم افزار به سمت چارچوب های خودآموز، تطبیق پذیر و امنیت محور حرکت می کند؛ چارچوب هایی که می توانند همگام با تغییرات سریع فناوری (مانند ادغام با اینترنت اشیا) و نیازهای کاربران تکامل یابند. در این مسیر استاندارد سازی رویه ها و سرمایه گذاری بر آموزش نسل جدید مهندسان برای همکاری با سیستم های هوشمند، تعیین کننده خواهد بود. آینده ای که نه تنها خودکار و هوشمند، بلکه انسان محور، شفاف و مسئولیت پذیر است.



۱۰-منابع

- [1] (Safari and Ebrahimi 2022, Sharifzadeh, Mirkoushesh et al. 2024, GH 2025, Saeed, Esmailzadeh et al. 2025) احمد ۲۰۱۱ and شیوا)
- [2] GH, S. (2025). "Opportunities and Considerations of Artificial Intelligence in the Health System (1): A Review of Documents and Studies in Iran and Worldwide." Reports 33(2): e20678.
- [3] Saeed, F., et al. (2025). "Analyzing the Relationships among the Indicators of LARGS Supply Chain Dimensions using Fuzzy DEMATEL in the Automotive Industry." Research in Production and Operations Management 16(3): 131-166.
- [4] Safari, E. and K. Ebrahimi (2022). "Prioritizing application areas to deploy artificial intelligence technology using thematic analysis and COPRAS." Research in Production and Operations Management 13(4): 91-110.
- [5] Sharifzadeh, Z., et al. (2024). "Examining the Impact of New Technology and Artificial Intelligence Development Policies on the Expansion of Macro-Political Strategies with a Focus on General Policies." Quarterly Journal of The Macro and Strategic Policies 12(45): 24-47.
- [6] R. Pan et al., "Reinforcement Learning for Automated Test Generation," *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pp. 305-316, 2020.
- [7] ACM Computing Surveys, "Machine Learning for Software Testing: A Systematic Review," 2024.
- [8] Gartner, "Magic Quadrant for Software Test Automation," 2024.
- [9] Microsoft Research, "Reinforcement Learning for Test Case Prioritization in Large-Scale Software Projects," 2023.
- [10] National Institute of Standards and Technology (NIST), "The Economic Impacts of Inadequate Infrastructure for Software Testing," 2023.