



INTELIGÊNCIA ARTIFICIAL NA GERAÇÃO AUTOMATIZADA DE TESTES DE SOFTWARE: Uma revisão bibliográfica

Gabriel R. RODRIGUES¹; Paulo C. SANTOS²

RESUMO

Este artigo apresenta uma Revisão Bibliográfica sobre o uso de Inteligência Artificial (IA) na geração automatizada de conteúdo para testes de software. O objetivo é mapear e analisar criticamente as principais técnicas, ferramentas e desafios descritos na literatura recente. Foram selecionados e revisados estudos publicados entre 2018 e 2024, com ênfase nas abordagens baseadas em Aprendizado de Máquina, Aprendizado Profundo e Processamento de Linguagem Natural. A análise evidencia avanços significativos no aumento da cobertura de testes e na redução de esforço humano, mas também destaca limitações, como a dependência da qualidade dos requisitos e a necessidade de validação humana. Este estudo oferece uma visão consolidada sobre o estado da arte, identificando lacunas e apontando direções para futuras pesquisas.

Palavras-chave: Geração de Testes Automatizados; Inteligência Artificial em Testes de Software; Aprendizado de Máquina; Processamento de Linguagem Natural; Engenharia de Software.

1. INTRODUÇÃO

A qualidade do software é um fator crítico no sucesso de sistemas computacionais. A etapa de testes, responsável por validar a funcionalidade e confiabilidade do código, é uma das mais exigentes em termos de tempo e esforço humano. A criação de artefatos como casos de teste, dados de entrada e scripts de automação é, tradicionalmente, realizada manualmente por testadores e desenvolvedores, o que torna o processo suscetível a erros e pouco escalável diante de sistemas complexos.

Com os avanços da Inteligência Artificial (IA), novas oportunidades têm surgido para automatizar a geração desses conteúdos de teste. Modelos como o GPT-4, BERT, e outras redes neurais profundas já demonstraram resultados promissores em tarefas que envolvem interpretação de requisitos e escrita de código. Técnicas como Aprendizado de Máquina (ML), Aprendizado Profundo (DL) e Processamento de Linguagem Natural (PLN) são utilizadas para analisar históricos de testes, código-fonte e documentação, gerando novos casos de teste com menor intervenção humana. Este artigo investiga como as técnicas de Inteligência Artificial têm sido aplicadas na geração automatizada de conteúdo para testes de software e sua eficácia

2. FUNDAMENTAÇÃO TEÓRICA

¹Bolsista PIBIC/FAPEMIG, IFSULDEMINAS – Campus Muzambinho. E-mail: gabriellaucher@gmail.com.

²Orientador, IFSULDEMINAS – Campus Muzambinho. E-mail: paulo.santos@muz.ifsuldeminas.edu.br.

A Inteligência Artificial (IA) é a área da Ciência da Computação dedicada ao desenvolvimento de sistemas que emulam capacidades humanas como raciocínio, aprendizado e resolução de problemas (RUSSELL; NORVIG, 2021). No domínio da engenharia de software, sua aplicação tem se expandido para automatizar tarefas críticas, como a identificação de falhas, a previsão de defeitos e, notadamente, a geração de casos de teste a partir de especificações técnicas (ARRIAGA; MEZA, 2022).

O Aprendizado de Máquina, um subcampo da IA, permite que sistemas aprendam com dados históricos de execução de testes e padrões de código, identificando áreas de risco e propondo cenários de validação mais eficazes. Já o Aprendizado Profundo, com suas redes neurais de múltiplas camadas, possibilita o entendimento de estruturas complexas e relações contextuais em grandes volumes de código-fonte. Segundo Chen e Liu (2023), arquiteturas como os Transformers se mostraram particularmente eficazes para modelar as estruturas sintáticas e semânticas do código, permitindo uma geração de testes mais contextualizada.

O Processamento de Linguagem Natural, por sua vez, é crucial para servir como ponte entre os requisitos do sistema, geralmente escritos em linguagem humana, e os artefatos formais de teste. O uso de modelos de linguagem avançados, como T5 e Codex, tem demonstrado uma capacidade notável de interpretar especificações funcionais e gerar automaticamente tanto os casos de teste quanto os dados de entrada correspondentes (SILVA; COSTA, 2023).

A convergência dessas técnicas deu origem a ferramentas modernas que buscam automatizar etapas críticas do ciclo de testes. Contudo, a eficácia dessas abordagens é intrinsecamente dependente da qualidade e clareza dos dados de entrada. Conforme apontam Zhang et al. (2024), requisitos ambíguos e documentação incompleta continuam sendo os principais obstáculos para uma automação totalmente confiável, tornando a validação humana uma etapa ainda indispensável no processo.

3. MATERIAL E MÉTODOS

Esta pesquisa caracteriza-se como uma Revisão Bibliográfica, conforme diretrizes da Jornada Científica do IFSULDEMINAS (JOSIF). O processo de busca e seleção da literatura seguiu as seguintes etapas:

3.1 Bases de Dados Utilizadas:

A busca foi realizada nas seguintes bases: IEEE Xplore, ACM Digital Library e Scopus.

3.2 Período de Publicação:

Foram selecionados artigos publicados entre 2018 e 2024.

3.3 Critérios de Inclusão:

- Estudos que abordam diretamente o uso de IA para gerar conteúdo de testes de software.

- Trabalhos revisados por pares (jornais acadêmicos e conferências).

3.4 Critérios de Exclusão:

- Artigos sem revisão por pares.
- Estudos fora do escopo de testes de software.

3.5 Procedimento de Seleção:

Foram utilizados descritores como "AI-driven software testing", "automated test generation", "machine learning in software testing", "geração automatizada de casos de teste" e "teste de software orientado por IA". Após leitura de títulos e resumos, os artigos elegíveis passaram por leitura completa para análise crítica.

3.5 Fluxo de Seleção dos Estudos:

Tabela 1-Fluxo de seleção dos artigos

Etapa	Quantidade
Artigos recuperados (total)	60
Excluídos por título/resumo	25
Excluídos pelos critérios de inclusão	15
Excluídos após leitura completa	8
Incluídos na análise final	12

Fonte: do autor

4. RESULTADOS E DISCUSSÃO

A revisão identificou um conjunto diversificado de abordagens para a geração automatizada de conteúdo para testes de software utilizando IA.

4.1 Técnicas Predominantes:

Arriaja e Meza (2022) apontam que o Aprendizado de Máquina tem sido largamente empregado na previsão de áreas de risco e na seleção de casos de teste prioritários. Chen e Liu (2023) destacam o uso de arquiteturas baseadas em Transformers para modelagem de código-fonte, aumentando a contextualização e precisão na geração de testes.

4.2 Uso de PLN:

Silva e Costa (2023) enfatizam o papel do PLN como mediador entre os requisitos textuais e os artefatos de teste. Modelos como T5 e Codex têm mostrado capacidade de interpretar especificações funcionais e gerar testes automatizados. No entanto, Zhang et al. (2024) alertam que requisitos ambíguos geram testes irrelevantes, reforçando a importância da validação manual.

4.3 Benefícios Observados:

A literatura convergente indica benefícios como aumento da cobertura de código, redução de esforço manual e maior padronização nos testes. Modelos como GPT-4 destacam-se pela

capacidade de gerar asserções de validação coerentes com os objetivos do teste.

4.4 Limitações Identificadas:

Além da dependência da qualidade dos dados de entrada, as abordagens atuais ainda apresentam dificuldades em lidar com requisitos vagos ou contraditórios. Como destacado por Zhang et al. (2024), mesmo os modelos mais avançados requerem intervenção humana para validação e refinamento dos testes gerados.

4.5 Tendências Futuras:

A literatura sugere que os próximos avanços devem focar na integração de dados históricos de projetos, na aplicação de técnicas de interpretação semântica sofisticadas e no desenvolvimento de pipelines híbridos que combinem heurísticas humanas com aprendizado automático.

5. CONCLUSÃO

A aplicação da Inteligência Artificial na geração automatizada de conteúdo para testes de software representa um avanço significativo na engenharia da qualidade. Esta revisão bibliográfica evidenciou que, apesar dos progressos em técnicas como ML, DL e PLN, ainda existem desafios consideráveis a serem superados, principalmente relacionados à qualidade dos requisitos de entrada e à necessidade de supervisão humana.

Contudo, o desenvolvimento de soluções mais robustas requer abordagens que combinem aprendizado de máquina com interpretação contextualizada e validação humana. A amostra final de 12 artigos representa 20% do total recuperado, demonstrando rigor na seleção e garantindo representatividade para a análise. Futuras pesquisas devem aprofundar a análise de métodos híbridos e explorar estratégias para melhorar a interpretação automática de requisitos, visando à construção de ferramentas mais confiáveis e efetivas.

REFERÊNCIAS

- ARRIAGA, A.; MEZA, J. A survey on AI-driven software testing: trends, challenges, and future directions. *IEEE Transactions on Software Engineering*, v. 48, n. 9, p. 3514–3535, 2022.
- CHEN, Z.; LIU, Y. Understanding source code with deep learning: a survey on models and applications. *ACM Computing Surveys*, v. 55, n. 8, p. 1–38, 2023.
- RUSSELL, S.; NORVIG, P. *Artificial intelligence: a modern approach*. 4. ed. Hoboken: Pearson, 2021.
- SILVA, F.; COSTA, E. From natural language to test cases: a systematic review of NLP for requirements-based testing. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE), 2023, Melbourne. *Proceedings...* Melbourne: ACM, 2023. p. 1120–1132.
- ZHANG, L. et al. The human-in-the-loop: overcoming the limits of AI in test automation. *Software Quality Journal*, v. 32, n. 1, p. 45–68, 2024.