

Aprendizado de Máquina aplicado ao Teste de Software

Vinicius Melchior L. Santos¹, Lidy Emanuelle M. Santos.¹, Odette M. Passos¹

¹Instituto de Ciência e Tecnologia – Universidade Federal do Amazonas (UFAM)
Caixa Postal 69103-128 – Manaus – AM – Brazil

{lidy.santos,vinicius.melchior, odette}@ufam.edu.br

Abstract. *Software testing is one of the main phases in guaranteeing the quality of a system, as it makes it possible not only to correct, but also to identify and prevent defects. Considering the constant evolution of technologies, there is a need to develop new ways of keeping tests effective and adaptable. In this context, Machine Learning applied to software testing has emerged, demonstrating new possibilities for validating quality. Thus, this rapid review sought to analyze relevant work in order to identify the main methods and tools, as well as the advantages and disadvantages of applying Machine Learning to software testing.*

Resumo. *Os testes de software representam uma das principais fases para a garantia da qualidade de um sistema por tornar possível não apenas a correção, mas a identificação e prevenção de defeitos. Considerando as constantes evoluções de tecnologias, observa-se a necessidade de desenvolver novas formas de manter os testes eficazes e adaptáveis. Nesse contexto, surge o Machine Learning aplicado a testes de software, que demonstram novas possibilidades de validação de qualidade. Dessa forma, por meio de uma Revisão Rápida, buscou-se analisar trabalhos relevantes para identificar os principais métodos e ferramentas, além de vantagens e desvantagens na aplicação do Machine Learning em testes de software.*

1. Introdução

Os testes possuem um papel de alta relevância para a garantia da qualidade de um software. Para Hourani et al. (2019), além de identificar defeitos, os testes devem identificar qualquer diferença entre o sistema existente e os resultados esperados, e acompanhar todo o ciclo de vida de desenvolvimento do software. Dessa forma, além de corrigir erros, os testes também desempenham um grande papel na prevenção de defeitos que possam vir a ocorrer.

Entretanto, a partir do aumento da complexidade de um software é necessário que os testes sejam capazes de acompanhar e continuar cumprindo com o objetivo de identificar e prevenir erros de forma adequada. Com isso, surgiu a necessidade de desenvolver novos métodos e ferramentas para auxiliar na realização de diferentes tipos de testes como a aplicação de Machine Learning em testes de software.

Portanto, o objetivo deste trabalho é realizar uma Revisão Rápida sobre trabalhos relevantes para o tema para verificar os principais métodos, desafios e as vantagens de utilização do Machine Learning em testes de software. Segundo Khangura (2012), uma Revisão Rápida utiliza métodos tradicionais de uma Revisão Sistemática, porém de forma mais simples, para obter informações e resultados confiáveis em menor tempo.

O trabalho está organizado nas seguintes seções: a Seção 2 apresenta os procedimentos metodológicos, na Seção 3 é apresentado o planejamento da revisão rápida, a Seção 4 apresenta a condução da revisão rápida e a Seção 5 contém o relato da revisão rápida.

2. Procedimentos Metodológicos

O presente trabalho trata-se de uma Revisão Rápida (RR), que conforme descrito por Cartaxo et al. (2020), é uma abordagem simplificada da Revisão Sistemática da Literatura (RSL). Motivada por demandas práticas que exigem a produção de informações em tempo hábil, segundo Kitchenham (2004), a RSL busca associar e avaliar evidências empíricas de um campo de estudo para responder a questões de pesquisa a partir de análises detalhadas, o autor propõe algumas etapas para o desenvolvimento da RSL, as quais foram adotadas:

- **Planejamento:** Etapa que compreendeu a definição das Questões de Pesquisa (QPs) e criação de uma string para a busca de trabalhos relevantes sobre a temática em bases de dados;
- **Condução:** Etapa que compreendeu a aplicação da string de busca em bases de dados nacionais e internacionais e aplicação de filtros para ajudar na seleção do material, bem como critérios de inclusão e exclusão;
- **Relato:** Etapa que compreendeu a análise dos trabalhos selecionados e discussão acerca dos trabalhos encontrados para auxiliar a construção da linguagem.

3. Planejamento da Revisão Rápida

3.1. Definição das Questões de Pesquisa

A estrutura inicial de planejamento é delineada na Tabela 1, no qual as Questões de Pesquisa (QPs) e suas motivações correspondentes estão apresentadas.

Tabela 1. Questões de pesquisa x motivação

QP	Questões de Pesquisa	Motivação
QP1	Quais as principais técnicas de aprendizado de máquina têm sido aplicadas em testes de software?	Investigar e compreender as principais técnicas de aprendizado de máquina utilizadas no processo de testes de software
QP2	Quais os benefícios e malefícios de utilizar aprendizado de máquina em testes de software?	Explorar as aplicações do aprendizado de máquina em testes de software e extrair os benefícios e malefícios dessas aplicações
QP3	Quais as limitações de utilizar aprendizado de máquina em testes de software?	Explorar as limitações do aprendizado de máquina no processo de testes de software

3.2. Definição de Palavras-Chaves

As palavras-chave utilizadas para o levantamento foram: aprendizado de máquina, testes de software, técnicas, benefícios, malefícios e limitações. A palavra-chave “aprendizado de máquina” foi escolhida devido à centralidade do tema nas questões de pesquisa, explorando sua aplicação no contexto de testes de software. O termo “testes de software” delimita o campo de estudo, “técnicas” foi incluída para abranger as metodologias específicas utilizadas no processo, enquanto “benefícios” e “malefícios” foram escolhidos

para analisar as vantagens e desvantagens dessas aplicações. Por fim, “limitações” explora as restrições e desafios associados ao uso de aprendizado de máquina no domínio especificado. A Tabela 2 mostra os termos utilizados em inglês e as variações.

Tabela 2. Palavras-Chaves

Palavras-Chaves	Termo em Inglês/Variações
Aprendizado de Máquina	Machine Learning
Testes de Software	Software Testing
Técnicas	Methods e Methodologies
Benefícios	Advantages e Gains
Malefícios	Disadvantages e Drawbacks
Limitações	Constraints, Challenges e Barriers

3.3. Definição das Fontes de Busca

Inicialmente a pesquisa foi realizada na base de dados dos eventos: (i) SAST: Simpósio Brasileiro de Testes de Software Sistemático e Automatizado e (ii) SBES: Simpósio Brasileiro de Engenharia de Software. Entretanto, devido à baixa quantidade de trabalhos identificados, foram exploradas outras bases de dados: IEEE Xplore e ScienceDirect, utilizando-se dos campos de conhecimento abrangidos pela engenharia de software para teste de software. A Tabela 3 apresenta as fontes utilizadas na busca por artigos científicos.

Tabela 3. Fontes de busca

Fontes Utilizadas	Link
SAST	https://sol.sbc.org.br/index.php/sast
SBES	https://sol.sbc.org.br/index.php/sbes
IEEE Xplore	https://ieeexplore.ieee.org/Xplore/
ScienceDirect	https://www.sciencedirect.com/

3.4. Definição do Recorte Temporal e Idioma

A delimitação para o escopo de tempo exigiu artigos que tenham sua publicação igual ou inferior a 5 anos. Assim, torna-se possível a identificação de lacunas e oportunidades de pesquisa. Ademais, o período pode ser aumentado para 7 anos, de forma que permita maior aquisição de conceitos consolidados relevantes à proposta. Para a delimitação do idioma houve uma restrição para que fossem utilizados artigos escritos em português ou inglês.

3.5. Definição da Expressão de Busca

Seguindo a lógica booleana (OR, AND), foi estabelecida a seguinte expressão de busca: Aprendizado de Máquina AND Testes de Software AND (Técnicas OR Métodos OR Metodologias) AND (Benefícios OR Vantagens OR Ganhos) AND (Malefícios OR Desvantagens OR Dificuldades) AND (Limitações OR Restrições OR Barreiras), com a possibilidade de modificá-las conforme as variações descritas na Tabela 1 e quebrar em substrings de busca na lógica AND. Além disso, foi utilizado a expressão de busca no idioma inglês.

3.6. Definição dos Critérios para Excluir/Incluir Artigos

Foram adotados os seguintes critérios para exclusão: duplicidade (trabalhos duplicados nas bases), diferentes idiomas do requerido, estudos que não tratem do tema mesmo que

pertencam aos resultados, não possuir completude (Resumos e Resumos expandidos) e trabalhos que não estão disponíveis gratuitamente. Dessa forma, todos os trabalhos que não cumprirem os critérios de exclusão estarão aptos a serem incluídos.

3.7. Definição da Extração de Dados

Foram extraídos os seguintes dados das publicações selecionadas: título, autores, fonte de publicação, ano de publicação, resumo, técnicas de Machine Learning, benefícios e malefícios de utilizar aprendizado de máquina, limitações de utilizar aprendizado e outras informações relevantes para a pesquisa, conforme mostra a Tabela 4.

Tabela 4. Dados extraídos

Dados da Publicação	
Título	Indica o título do trabalho
Autor(es)	Nome dos autores
Fonte de Publicação	Local de publicação
Ano da Publicação	Ano de publicação
Resumo	Texto contendo uma descrição do resumo informado pelos autores
Dados Derivados das QPs	
Técnicas	Técnicas de machine learning presentes no trabalho
Benefícios/Malefícios	Benefícios e malefícios de utilizar aprendizado de máquina
Limitações	Limitações de utilizar aprendizado em testes de software
Dados Adicionais	
Outras Informações	Descrição de outras informações consideradas importantes

4. Condução da Revisão Rápida

Concluída a etapa de planejamento, as expressões de buscas foram executadas nas fontes de busca, resultando em um total de 778 publicações. A seleção ocorreu em duas etapas, onde na 1ª seleção foi realizada a leitura do título, palavras-chaves e resumo e na 2ª seleção foi realizada a leitura completa dos trabalhos selecionados na primeira fase.

O corpus final de estudos incluiu 13 artigos aprovados pelos critérios estabelecidos no planejamento. Destaca-se que todos os artigos selecionados se encontram no idioma inglês, devido a serem provenientes principalmente das bases de dados IEEE Xplore e ScienceDirect, conforme mostra a Tabela 5.

Tabela 5. Quantitativo de publicações

Fonte de Busca	Publicações (aprox.)	1ª Seleção	2ª Seleção
SAST	60	2	1
SBES	368	2	2
IEEE Xplore	167	10	7
ScienceDirect	183	3	3
Total	778	17	13

Os 13 artigos selecionados abordam tópicos relevantes para a aplicação de técnicas de aprendizado de máquina no contexto de teste de software. São exploradas áreas como geração automática de casos de teste, otimização de suítes de teste, predição de defeitos, priorização de casos de teste e validação de modelos de aprendizado de máquina.

A partir dos conteúdos adquiridos por meio desses trabalhos, além de responder as questões de pesquisa, foi realizado uma breve análise sobre a quantidade e ano de publicação. Na Tabela 6, estão descritos todos os artigos e seus respectivos autores, assim

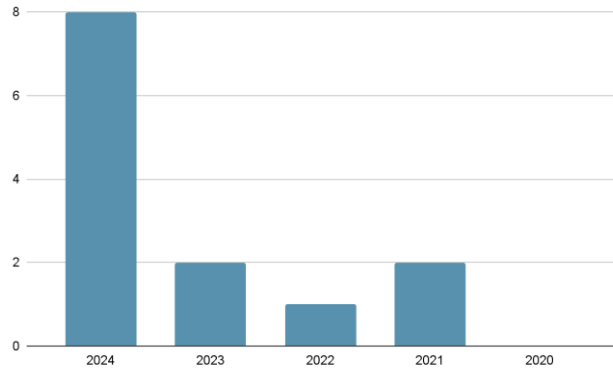
como o ano de publicação e o local de busca onde foram publicadas. Na Figura 1 encontra-se a relação de ano x quantidade de artigos selecionados, onde observa-se que 2024 foi o ano com maior quantidade de publicações relacionadas ao tema.

Tabela 6. Artigos selecionados

ID	Título	Autores	Ano	Local de Busca
P1	AutomTest 3.0: An automated test-case generation tool from User Story processing powered with LLMs	Joanderson Gonçalves Santos, Rita Suzana Pitangueira Maciel	2024	SBES
P2	Comparative Analysis of Large Language Model Tools for Automated Test Data Generation from BDD	Isela Mendoza, Fernando Silva Filho, Gustavo Medeiros, Aline Paes, Vânia O. Neves	2024	SBES
P3	Property-based Testing for Machine Learning Models	Vinicius H. S. Durelli, Ricardo Monteiro, Rafael S. Durelli, Andre T. Endo, Fabiano C. Ferrari, Simone R. S. Souza	2024	SAST
P4	A Literature Review of Using Machine Learning in Software Development Life Cycle Stages	Saad Shafiq; Atif Mashkoor; Christoph Mayr-Dorn; Alexander Egyed	2021	IEEE
P5	Trend Application of Machine Learning in Test Case Prioritization: A Review on Techniques	Muhammad Khatibsyarbini; Mohd Adham Isa; Dayang N. A. Jawawi; Muhammad Luqman Mohd Shafie; Wan Mohd Nasir Wan-Kadir; Haza Nuzly Abdull Hamed	2021	IEEE
P6	Test Suite Optimization Using Machine Learning Techniques: A Comprehensive Study	Abid Mehmood; Qazi Mudassar Ilyas; Muneer Ahmad; Zhongliang Shi	2024	IEEE
P7	Validating Unsupervised Machine Learning Techniques for Software Defect Prediction With Generic Metamorphic Testing	Pak Yuen Patrick Chan; Jacky Keung	2024	IEEE
P8	A Novel Approach to Improve Software Defect Prediction Accuracy Using Machine Learning	Iqra Mehmood; Sidra Shahid; Hameed Hussain; Inayat Khan; Shafiq Ahmad; Shahid Rahman	2023	IEEE
P9	Systematic Mapping: Artificial Intelligence Techniques in Software Engineering	Hazrina Sofian; Nur Arzilawati Md Yunus; Rodina Ahmad	2022	IEEE
P10	Generating Software Tests for Mobile Applications Using Fine-Tuned Large Language Models	Jacob Hoffmann; Demian Frister	2024	IEEE
P11	The role of Reinforcement Learning in software testing	Amr eleneen, Ahammed Palliyali, Cagatay Catal	2023	ScienceDirect
P12	An optimized neural network for prediction of security threats on software testing	Suman, Raees Ahmad Khan	2024	ScienceDirect
P13	A systematic review of machine learning methods in software testing	Sedighe Ajorloo, Amirhossein Jamarani,	2024	ScienceDirect

		Mehdi Kashfi, Mostafa Haghi Kashani, Abbas Najafizadeh		
--	--	--	--	--

Figura 1. Ano x Quantidade



5. Relato da Revisão Rápida

A análise das 13 publicações selecionadas foi realizada de forma sistemática, seguindo o protocolo e fluxo pré-definido no planejamento da RR. Através da codificação e categorização dos dados extraídos dos estudos, foi possível responder às questões de pesquisa e construir um quadro abrangente do estado da arte sobre Machine Learning aplicada aos processos de Teste de Software.

5.1. Com relação à Primeira Questão da Pesquisa: Quais principais técnicas de aprendizado de máquina têm sido aplicadas em testes de software?

A análise das 13 publicações revelou uma diversidade de técnicas de Machine Learning aplicadas em Testes de Software, como pode-se observar na Tabela 7, com destaque para redes neurais artificiais, árvores de decisão e KNN. Observa-se uma tendência crescente no uso de Deep Learning para tarefas complexas, como geração automática de casos de teste e detecção de anomalias. No entanto, a maioria dos estudos se concentra em técnicas supervisionadas, indicando uma necessidade de explorar mais aprofundadamente o potencial do aprendizado não supervisionado e por reforço.

Tabela 7. Principais técnicas

ID	Técnica	Publicações Relacionadas	Citação
T1	K-Vizinhos Mais Próximos (KNN)	P5, P6, P7, P8, P13	5
T2	Máquinas de Vetores de Suporte (SVM)	P4, P6, P8, P13	4
T3	Floresta Aleatória	P4, P7, P8, P13	4
T4	Árvore de Decisão	P4, P5, P6, P8	4
T5	Q-Learning	P5, P6, P11, P13	4
T6	Algoritmos de Clustering K-Means	P5, P6, P9, P13	4
T7	Redes Neurais Artificiais	P4, P5, P8	3
T8	LLMs	P1, P2, P3	3
T9	Redes Neurais Convolucionais	P4, P13	2
T10	Redes Neurais Recorrentes	P4, P13	2
T11	Perceptron Multicamadas	P8, P13	2
T12	Naive Bayes	P4, P6	2
T13	Regressão Linear	P6, P9	2

T14	Rede Bayesiana	P5, P8	2
T15	Algoritmos Fuzzy	P5, P6	2
T16	Regressão Logística	P8	1
T17	Regra ZeroR	P8	1
T18	Inteligência de Enxame	P5	1
T19	Redes Neurais Feedforward	P13	1
T20	Redes Neurais Profundas (DQNs)	P5	1
T21	Toco de Decisão	P8	1
T22	HDBSCAN	P6	1
T23	Modelo de Mistura Gaussiana (GMM)	P7	1
T24	Deep Deterministic Policy Gradient (DDPG)	P11	1
T25	Deep Q-Network (DQN)	P11	1
T26	Advantage Actor-Critic (A2C)	P11	1
T27	Asynchronous Advantage Actor-Critic (A3C)	P11	1
T28	Soft Actor-Critic (SAC)	P11	1
T29	Dove Swarm-based deep neural method (DSbDNM)	P12	1

Com relação à principal questão de pesquisa, as técnicas mais frequentemente aplicadas são K-Vizinhos Mais Próximos (KNN), Máquinas de Vetores de Suporte (SVM), Floresta Aleatória e Árvores de Decisão, cada uma sendo citada em pelo menos quatro estudos. Essas técnicas têm se mostrado particularmente úteis em tarefas como classificação de casos de teste e priorização de defeitos, como o demonstrado nos trabalhos de Shafiq et al. (2021) e Khatibsyarhini et al. (2021), que em suas publicações abordam aplicações de diferentes modelos de aprendizado de máquina em processos que compõem a etapa de teste de software. Além dos principais modelos, apresentam também outros modelos como: Redes Neurais Artificiais, incluindo arquiteturas mais avançadas como Redes Neurais Convolucionais e Redes Neurais Recorrentes, destacando-as em aplicações complexas, como a geração automática de casos de teste e detecção de anomalias.

Outros artigos como o de Mehmood et al.(2024) destacam técnicas de aprendizado de máquina como o K-Means utilizando-os em processos de teste de software que envolvem agrupamentos e classificações. Porém não se limitando a esta técnica, foram utilizadas outras, de aprendizado por reforço, como as apresentadas por Amr Abo-eleneen et al. (2024) que mescla técnicas clássicas de aprendizado de máquina com atuais como a Deep Q-Networks (DQN) e Advantage Actor-Critic (A2C), entretanto, estas abordagens são dependentes de uma função de valor de “ação ótima” ou políticas que precisam ser extremamente detalhadas para que a aplicação alcance resultados satisfatórios.

Nota-se que dentre os artigos analisados, poucos citam técnicas de aprendizado não supervisionado. Isso indica que a intervenção humana ainda é necessária nos processos de teste de software. Visando a qualidade final do produto de software, porém, é destacado por Iqra Mehmood et al. (2023) que técnicas como as Redes Neurais Feedforward e Perceptron Multicamadas demonstraram seu valor, mas a complexidade e a falta de interoperabilidade continuam sendo barreiras para a adoção em larga escala, especialmente em contextos críticos.

5.2. Com relação à Segunda Questão de Pesquisa: Quais os benefícios e malefícios de utilizar aprendizado de máquina em testes de software?

Para verificar as principais vantagens apresentadas pelos trabalhos selecionados, foram criadas algumas classificações para realizar o agrupamento de relatos semelhantes descritos de diferentes formas e proporcionar uma melhor análise dos resultados. As

categorias foram criadas com o objetivo de demonstrar um panorama geral das vantagens mais recorrentes em trabalhos de aplicação de machine learning em testes de software.

Na Tabela 8, observa-se a relação de benefícios relatados pelos trabalhos selecionados para o desenvolvimento da RR. As categorias criadas como forma de sintetizar dados foram: Facilita a automação de processos de teste de software, Automatização de tarefas humanas, Redução de custos em testes de software, Aumento de precisão e Otimização de tempo de execução.

Tabela 8. Benefícios identificados

ID	Benefícios	Publicações Relacionadas	Citação
B1	Facilita a automação de processos de teste de software	P1, P2, P5, P6, P9, P10, P11	7
B2	Automatização de tarefas humanas	P1, P4, P5, P7, P9, P10, P11	7
B3	Redução de custos em testes de software	P1, P2, P5, P6, P8, P9, P13	7
B4	Aumento da precisão	P4, P5, P6, P8, P13	5
B5	Otimização de tempo de execução	P5, P6, P8, P13	4
B6	Adaptatividade	P12, P13	2
B7	Otimização de esforço de teste	P5, P13	2
B8	Aumento de eficiência	P6, P13	2
B9	Alta escalabilidade	P3, P6	2
B10	Redução de erros	P4, P7	2
B11	Exploração dos limites	P3	1
B12	Geração de várias entradas com uma única especificação	P3	1
B13	Aplicações em grandes conjuntos de dados	P4	1
B14	Validação de modelos	P7	1
B15	Não necessidade de dados bem rotulados	P7	1
B16	Simplicidade	P12	1
B17	Baixa sobrecarga	P13	1

Para Santos et al. (2024), Mendoza et al. (2024) e Mehmood (2024) a automação dos processos relacionados aos testes de software é uma das principais vantagens da aplicação do Machine Learning. Esses processos podem envolver desde a criação dos cenários até a realização dos testes, além disso, o fator automação influencia na diminuição da necessidade de trabalho manual, vantagem citada em outros trabalhos como Shafiq (2021), Khatibsyarbini et al. (2021), Chan e Keung (2024), Sofian et al. (2022), Hoffmann e Frister (2024) e Abo-eleneen et al. (2024).

A redução de custos relacionados aos Testes de Software também é apresentada como um ponto importante em trabalhos como Mehmood et al. (2023) e Ajorloo et al. (2023). A redução de custos engloba questões como a melhor alocação de recursos e redução de custos de execução. Outra observação recorrente é a maior precisão, onde Shafiq (2021) destaca a precisão em classificação e em previsões de erros. Já Khatibsyarbini et al. (2021) e Mehmood (2024) demonstram a maior precisão sob a perspectiva da priorização e seleção de casos de testes.

Quanto aos malefícios (veja a Tabela 9) referentes a utilização de Machine Learning em Testes de Software, um dos mais citados, como no trabalho de Chan e Keung (2024), Mehmood et al. (2023) e Hoffmann e Frister (2024), foi a sensibilidade aos dados, que se refere a capacidade de lidar com variações na base de dados o que pode levar a diferentes resultados, e até mesmo inconsistentes. Hoffmann e Frister (2024) junto a

Mehmood (2024) e Khatibsyarhini et al. (2021), também, apontaram que a adaptação é um aspecto que pode apresentar problemas, pois existe a possibilidade de não responder positivamente a mudança de ambientes, ferramentas ou complexidade de dados.

Tabela 9. Malefícios identificados

ID	Malefícios	Publicações Relacionadas	Citação
M1	Sensibilidade ao tipo de dados	P7, P8, P10	3
M2	Problemas de adaptação	P5, P6, P10	3
M3	Dependência de grandes conjuntos de dados de alta qualidade e bem rotulados	P4, P5, P6	3
M4	Generalização	P4, P5, P6	3
M5	Reprodutibilidade limitada	P4, P10	2
M6	Alta complexidade computacional	P6, P10	2
M7	Dificuldade de implementação	P5, P11	2
M8	Necessidade de revisão humana	P1, P2	2
M9	Risco de overfitting	P4, P6	2
M10	Baixa escalabilidade	P13	1
M11	Baixa interoperabilidade	P13	1
M12	Falta de disponibilidade de conjuntos de dados específicos	P4	1

Shafiq (2021), Khatibsyarhini et al. (2021) e Mehmood(2024) definiram outros contras, são eles, a dependência de base de dados de alta qualidade e bem rotulados, pois a utilização de dados menos específicos pode atrapalhar o aprendizado e futuros resultados e a generalização que pode ser um problema na utilização em novos cenários.

5.3. Com relação à Terceira Questão de Pesquisa: Quais as limitações de utilizar aprendizado de máquina em testes de software?

É notório que a aplicação do Aprendizado de Máquina nos processos de Teste de Software possuem diversas limitações e desafios. Visando extrair a mensagem principal dos artigos, foram criadas classes específicas com o objetivo de agrupar as limitações mencionadas nas publicações, considerando que os autores abordam questões similares de formas distintas, essa abordagem foi adotada para identificar e sintetizar as limitações recorrentes do uso de aprendizado de máquina em testes de software.

As categorias criadas refletem os desafios mais comuns apresentados nos artigos analisados, como dependência de dados, desafios técnicos e algorítmicos, escalabilidade e generalização, cobertura de testes, e limitações específicas de estudos, todas identificadas em diferentes publicações. Dessa maneira, é possível compreender as limitações de forma consolidada, apesar da variação na redação e na ênfase dos autores. Na Tabela 10 estão relacionadas as limitações e publicações encontradas na leitura dos artigos.

Tabela 10. Limitações identificadas

ID	Limitações	Publicações Relacionadas	Citação
L1	Dependência de Dados	P3, P4, P5, P6, P7, P8, P10, P11, P12, P13	10
L2	Desafios Técnicos e Algorítmicos	P5, P7, P8, P11, P13	5
L3	Escalabilidade e Generalização	P4, P5, P6, P12	4
L4	Cobertura de Testes	P1, P2, P10, P13	4
L5	Limitações Específicas de Estudos	P2, P7, P8, P10	4

Categorizou-se os desafios identificados nos artigos e foi destacado principalmente a dependência de dados para o treinamento dos modelos, estando presente em 10 das publicações analisada. Esse fato confirma os tópicos apontados por Sedighe Ajorloo et al. (2024) em sua revisão sistemática que demonstra a relevância dos desafios relacionados à qualidade, balanceamento, rotulagem e disponibilidade de dados para o sucesso das técnicas de aprendizado de máquina. Em particular, a ausência de dados representativos e de alta qualidade é apontada como um fator crítico para a eficácia e generalização dos modelos.

A dependência de grandes volumes de dados de alta qualidade é um desafio recorrente na aplicação do machine learning nos processos da etapa de teste de software, pois a falta ou a baixa qualidade dos dados de treinamento podem prejudicar a precisão dos modelos. Por outro lado, trabalhos como o de Pak Yuen et al. (2024), relatam que a escolha do algoritmo adequado, o ajuste de hiperparâmetros e a complexidade computacional são desafios técnicos que podem limitar a aplicabilidade do machine learning. Além disso, a escalabilidade e a generalização destacada principalmente por Mehmood et al. (2024), aborda que os modelos também são questões importantes, especialmente em sistemas de grande porte e com requisitos de adaptação contínua ou que são desenvolvidos utilizando metodologias ágeis.

6. Considerações Finais

A Revisão Rápida realizada demonstrou que a aplicação de Aprendizado de Máquina em Testes de Software oferece avanços significativos, mas também apresenta desafios notáveis. As principais contribuições incluem a identificação das técnicas mais utilizadas, como redes neurais, árvores de decisão e KNN, além de novas abordagens como aprendizado por reforço e Deep Learning. Essas técnicas têm sido aplicadas em tarefas diversas, como geração de casos de teste e priorização de defeitos, trazendo benefícios como automação, redução de custos e maior precisão. Contudo, as limitações observadas incluem a dependência de dados de alta qualidade, dificuldades de generalização e desafios de escalabilidade, evidenciando a necessidade de avanços metodológicos e maior disponibilidade de datasets balanceados e rotulados.

A Revisão Rápida revelou que o aprendizado de máquina possui grande potencial para transformar os processos de teste de software, proporcionando avanços significativos em automação, precisão e redução de custos. No entanto, para que essas soluções sejam amplamente adotadas, é necessário superar desafios como dependência de dados de alta qualidade, dificuldades de escalabilidade e a necessidade de maior generalização das técnicas. Além disso, a pesquisa destacou a importância de explorar novos caminhos, como o aprendizado não supervisionado e abordagens híbridas, para lidar com cenários complexos e diversos. Assim, o estudo reforça a necessidade de esforços contínuos em pesquisa e desenvolvimento para garantir que essas técnicas possam atender às crescentes demandas da indústria de software.

Referências

- Abo-Eleneen, A. Palliyali, A. and Catal, C. (2023). The role of reinforcement learning in software testing. *Information and Software Technology*, v. 164.
- Ajorloo, S. Jamarani, A. Kashfi, M. Kashani, Haghi M. (2024). A systematic review of machine learning methods in software testing. *Applied Soft Computing*, p. 111805.

- Cartaxo, B. Pinto, G. e Soares, S. (2020). Rapid Reviews in Software Engineering. Chapter in Book: Contemporary Empirical Methods in Software Engineering. Springer, p.357–384.
- Chan, P. Y. P. Keung, J. (2024). Validating unsupervised machine learning techniques for software defect prediction with generic metamorphic testing. IEEE Access, v. 12, p. 165155-165172.
- Hourani, H. Hammad, A. Lafi, M. (2019). The impact of artificial intelligence on software testing. IEEE Jordan International Conference on Electrical Engineering and Information Technology (JEEIT), p. 565-570.
- Hoffmann, J. Frister, D. (2024). Generating software tests for mobile applications using fine-tuned large language models. IEEE/ACM International Conference on Automation of Software Test (AST). p. 76-77.
- Mendoza, I. Silva F, Fernando. M, Gustavo. P, Aline. N, Vânia O. Comparative Analysis of Large Language Model Tools for Automated Test Data Generation from BDD. Simpósio Brasileiro De Engenharia De Software (SBES). Porto Alegre: Sociedade Brasileira de Computação, p. 280-290
- Mehmood, A. Ilyas, M. Ahmad, M. Shi, Z. (2024). Test suite optimization using machine learning techniques: a comprehensive study. IEEE Access, v. 12, p. 168645-168671.
- Mehmood, I. et al. (2023). A novel approach to improve software defect prediction accuracy using machine learning. IEEE Access, v. 11, p. 63579-63597.
- Kitchenham, B. (2004). Procedures for performing systematic reviews. Keele, UK, Keele University, v. 33, s/n, p. 1-26.
- Khangura, S. Konnyu, K. Cushman, R. Grimshaw, J. e Moher, D. (2012). Evidence Summaries: The Evolution of a Rapid Review Approach. Systematic Reviews 1, n. 10.
- Santos, J. G. Maciel, R. S. P. (2024). AutomTest 3.0: An automated test-case generation tool from User Story processing powered with LLMs. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE (SBES), 38. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, p. 769-774.
- Sofian, H.; Yunus, N. A. M.; Ahmad, R. Systematic mapping: artificial intelligence techniques in software engineering. IEEE Access, v. 10, p. 51021-51040.
- Shafiq, S.; Mashkoo, A.; Mayr-dorn, C.; Egyed, A. (2021). A literature review of using machine learning in software development life cycle stages. IEEE Access, v. 9, p. 140896-140920
- Khatibsyarbini, M.; et al. (2021). Trend application of machine learning in test case prioritization: a review on techniques. IEEE Access, v. 9, p. 166262-166282.