

COMO O APRENDIZADO DE MÁQUINA POR REFORÇO PODE AUXILIAR NO AMBIENTE DE TESTE DE SOFTWARE

Matheus Sanchez Duda¹ JULIO PEDRO DE SIQUEIRA JUNIOR² CLAUDIA
BIANCHI PROGETTI³

Centro Universitário Senac: Matheus.sduda@senacsp.edu.br

Centro Universitário Senac: julio.psjunior@senacsp.edu.br

Centro Universitário Senac: claudia.bprogetti@senacsp.edu.br

Resumo: Este trabalho investiga o potencial do Aprendizado por Reforço (*Reinforcement Learning* — RL) como abordagem para otimização e automação inteligente de processos de teste de software, considerando o aumento da complexidade de sistemas computacionais e a crescente demanda por qualidade, confiabilidade e rapidez na entrega de aplicações. A introdução contextualiza a evolução das práticas de Garantia de Qualidade (QA) e o papel emergente da Inteligência Artificial na geração automática de casos de teste, priorização de execuções e identificação de falhas não triviais. Diferentemente da evolução previsível do hardware descrita pela Lei de Moore, o avanço da Inteligência Artificial não segue uma regra linear ou determinística de progresso, tornando necessário investigar empiricamente quais abordagens apresentam ganhos reais e sustentáveis em ambientes aplicados. O objetivo geral do estudo é analisar como agentes de RL podem aprender políticas eficientes para seleção e geração de entradas de teste, maximizando cobertura de código e detecção de defeitos, ao mesmo tempo em que reduzem redundâncias e esforço humano. Como objetivos específicos, busca-se: mapear algoritmos de RL aplicáveis ao contexto de testes automatizados; projetar funções de recompensa alinhadas a métricas de qualidade de software; e avaliar comparativamente o desempenho desses agentes frente a técnicas tradicionais de automação. A metodologia adotada foi quantitativa e exploratória, combinando revisão sistemática da literatura com experimentação controlada. Foram analisadas bases acadêmicas e relatórios técnicos de organizações como MIT e IBM, identificando tendências em IA aplicada à engenharia de *software*. Em seguida, desenvolveu-se um protótipo experimental baseado em *Q-Learning* e variações de *Deep Q-Networks* (DQN), integrado a um ambiente simulado capaz de representar fluxos de execução e múltiplos caminhos de código. O desenho experimental seguiu o princípio de isolamento de variáveis, controlando parâmetros como taxa de aprendizado, fator de desconto e estratégia de exploração, a fim de avaliar o impacto individual de cada componente no desempenho do agente. As métricas utilizadas incluíram cobertura de linhas e ramos, número de falhas detectadas e redução de testes redundantes. Os resultados indicaram que agentes de RL conseguem aprender estratégias que aumentam a cobertura em cenários específicos e identificam padrões de falha não explorados por abordagens determinísticas tradicionais. Observou-se, entretanto, sensibilidade significativa à modelagem da função de recompensa e necessidade de

múltiplas iterações para estabilização do aprendizado. Conclui-se que o uso de RL em testes de *software* representa uma linha promissora para automação adaptativa e inteligente, mas ainda carece de padronização metodológica e validação em ambientes industriais de larga escala. O estudo reforça que, embora a IA apresente avanços acelerados, sua evolução depende de fatores experimentais, arquiteturais e contextuais, não obedecendo a uma progressão previsível como a observada na miniaturização de transistores, o que demanda investigação contínua e rigor científico para consolidação prática.

Palavras-chave: Aprendizado por Reforço; Teste de Software; Automação; Inteligência Artificial; QA.