

Um Mapeamento Sistemático sobre Técnicas de Classificação e Categorização de Bugs em Software

Por Igor Ramsés Temóteo dos Santos, Laís Carvalho Coutinho, Samuel Sousa Teles, Rubens Abraão da Silva Sousa, Manoel Lopes Filho, Ronaldo T. P. Milfont e Alex Ramos

Fonte: OpenAlex · IFCE · [Publicação original](#)

INOVAÇÃO	APLICABILIDADE	POTENCIAL ECONÔMICO	MATURIDADE
7/10	8/10	9/10	Média

Este mapeamento sistemático da literatura sobre classificação e categorização de bugs em software destaca a predominância de algoritmos supervisionados e a dependência de dados de repositórios open-source. A pesquisa aponta uma lacuna crítica na compreensão do ciclo de vida dos bugs em ambientes corporativos e a necessidade de explorar modelos híbridos de dados (estruturados e não-estruturados) para uma aplicação mais eficaz em contextos industriais.

NÚCLEO DA ANÁLISE

Ideia de startup ou produto

BugSense AI: Uma plataforma SaaS que utiliza inteligência artificial para classificação e categorização de bugs, com foco na integração de dados híbridos (textos, logs, telemetria, métricas de código) de ambientes corporativos. A plataforma ofereceria modelos adaptáveis a diferentes indústrias, insights sobre o ciclo de vida dos bugs e otimização da priorização e atribuição, superando as limitações dos dados open-source.

Aplicações práticas

Automação da triagem e atribuição de bugs em equipes de desenvolvimento de software, otimização da alocação de recursos para correção de defeitos, melhoria contínua dos processos de garantia de qualidade (QA) e desenvolvimento de sistemas preditivos para detecção e gestão de falhas.

Potencial de mercado

Alto. Todas as empresas de desenvolvimento de software, de startups a grandes corporações, enfrentam o desafio da gestão de bugs. Soluções que automatizam e aprimoram a classificação de bugs podem gerar economias significativas de tempo e custo, além de elevar a qualidade do software. O mercado de ferramentas de

Application Lifecycle Management (ALM) e QA é robusto e em crescimento.

FUNDAMENTO CIENTÍFICO

Problema abordado

A ineficiência na classificação e categorização de bugs em software, que impacta diretamente a qualidade do produto, o tempo de desenvolvimento e a alocação de recursos. A limitação atual reside na dependência de dados open-source, que não refletem adequadamente a complexidade e particularidades dos ambientes corporativos.

Metodologia

O estudo é um mapeamento sistemático da literatura, seguindo o modelo PRISMA. Foram analisados 33 artigos publicados entre 2019 e 2024, com busca nas bases ISI Web of Science, IEEE Xplore, Scopus e Engineering Village. A triagem foi realizada com a ferramenta Parsifal, e a síntese dos dados por análise qualitativa e extração padronizada.

Principais descobertas

1. Predominância de algoritmos supervisionados (Naive Bayes, Support Vector Machine) para classificação de bugs. 2. Forte dependência de dados textuais não estruturados de repositórios open-source (Mozilla, Eclipse). 3. Restrição na compreensão do ciclo de vida dos bugs em contextos industriais devido à exclusividade de dados open-source. 4. Necessidade de pesquisas futuras que explorem ambientes corporativos e a integração de modelos híbridos de dados.

LEITURA PARA GESTÃO PÚBLICA

Políticas de incentivo à pesquisa e desenvolvimento (P&D) em IA aplicada à qualidade de software, com foco em soluções para ambientes corporativos. Fomento à criação de datasets anonimizados de bugs de empresas locais para pesquisa acadêmica. Programas de capacitação profissional em automação de QA e gestão inteligente de bugs para fortalecer o ecossistema de tecnologia do Ceará.

PARCERIA UNIVERSIDADE × EMPRESA

Parceria entre o IFCE e empresas de software do Ceará (e.g., do polo de Fortaleza) para desenvolver projetos-piloto de IA para classificação de bugs, utilizando dados reais (anonimizados) de seus produtos. Isso permitiria ao IFCE validar suas pesquisas em um contexto industrial e às empresas obterem soluções personalizadas e eficientes para seus desafios de QA.

OPORTUNIDADES DERIVADAS

4 direções estratégicas identificadas

STARTUP · IMPACTO ALTO · INTELIGÊNCIA ARTIFICIAL

Plataforma IA para Classificação Híbrida de Bugs Corporativos

Desenvolver uma plataforma SaaS que aplica IA para classificar e categorizar bugs, integrando dados estruturados e não-estruturados de ambientes corporativos, visando otimizar o ciclo de vida do bug e a alocação de recursos em empresas de software.

PARCERIA · IMPACTO MÉDIO · SOFTWARE

Colaboração Universidade-Empresa para Validação de Modelos de IA em QA

Estabelecer parcerias entre o IFCE e empresas de desenvolvimento de software para aplicar e validar modelos de IA de classificação de bugs em dados reais de projetos corporativos, preenchendo a lacuna de pesquisa identificada e gerando soluções práticas.

POLÍTICA PÚBLICA · IMPACTO MÉDIO · GOVTECH

Programa de Fomento à Inovação em Qualidade de Software com IA

Criar políticas públicas que incentivem o desenvolvimento de soluções de IA para automação de Quality Assurance e gestão de bugs em software, especialmente aquelas que abordam o contexto corporativo e a criação de datasets para pesquisa e desenvolvimento local.

PRODUTO CORPORATIVO · IMPACTO ALTO · SOFTWARE

Módulo de IA para Ferramentas de Gerenciamento de Bugs Existentes

Desenvolver um módulo de inteligência artificial que possa ser integrado a ferramentas de gerenciamento de bugs já utilizadas (e.g., Jira, Azure DevOps), automatizando a classificação, categorização e sugestão de atribuição de bugs com base em dados híbridos da empresa.

ABSTRACT ORIGINAL

Este estudo é um mapeamento sistemático da literatura sobre as abordagens de classificação e categorização de bugs em software, analisando 33 artigos finais publicados entre 2019 e 2024. O protocolo foi conduzido seguindo o modelo PRISMA, utilizando a ferramenta Parsifal para triagem, e a busca foi realizada nas bases ISI Web of Science, IEEE Xplore, Scopus e Engineering Village. A síntese dos dados foi realizada através de análise qualitativa e extração padronizada, focando em técnicas, tipos de dados, contextos de aplicação e critérios de categorização. A pesquisa revelou a predominância de algoritmos supervisionados (como Naive Bayes e Support Vector Machine) e a dependência de dados textuais não estruturados de repositórios open-source como Mozilla e Eclipse. A exclusividade de dados open-source restringe a compreensão do ciclo de vida dos bugs em contextos industriais. Consequentemente, identifica-se a necessidade de pesquisas futuras que explorem ambientes corporativos e a integração de modelos híbridos (dados estruturados e não-estruturados) para melhor reflexão do ciclo de vida dos bugs.

MATÉRIA PARA LEIGOS

Para leigos: Desvendando os bugs: o mapa da ciência para softwares mais confiáveis

O cenário atual

Todo mundo que usa um celular, computador ou qualquer sistema digital já deve ter se deparado com um "bug". Um bug é um erro ou falha no código de um programa de computador que faz com que ele se comporte de maneira inesperada, trave ou não funcione como deveria. Para quem desenvolve software, entender e corrigir esses bugs é uma parte crucial do trabalho. Classificar e categorizar esses erros ajuda a encontrar soluções mais rápido e a criar programas mais robustos.

O que os pesquisadores fizeram

Pesquisadores do IFCE realizaram um estudo detalhado para entender como a ciência tem abordado a classificação e categorização de bugs em softwares. Eles fizeram um "mapeamento sistemático da literatura", que é como uma grande pesquisa sobre outras pesquisas.

Para isso, analisaram 33 artigos científicos publicados entre 2019 e 2024. Eles seguiram um método rigoroso (o modelo PRISMA) e usaram ferramentas específicas para encontrar e selecionar os artigos mais relevantes. A busca foi feita em grandes bases de dados acadêmicas, como ISI Web of Science, IEEE Xplore, Scopus e Engineering Village. O objetivo era descobrir quais técnicas, tipos de dados, contextos e critérios são mais usados para organizar e entender os bugs.

Como funciona na prática

O estudo mostra que, na prática, muitas das técnicas para identificar e classificar bugs em software usam algoritmos de "aprendizado de máquina". Pense neles como sistemas inteligentes que aprendem a reconhecer padrões. Os mais comuns são o Naive Bayes e o Support Vector Machine, que são tipos de algoritmos supervisionados. Isso significa que eles são "treinados" com exemplos de bugs que já foram classificados por humanos.

Esses algoritmos geralmente trabalham com dados textuais, como as descrições de erros que os programadores ou usuários escrevem. Esses textos vêm de grandes repositórios de software de código aberto (open-source), como os projetos Mozilla e Eclipse.

Resultados e evidência

A análise dos 33 artigos revelou alguns pontos importantes:

- Algoritmos populares: Há uma forte preferência por algoritmos de aprendizado supervisionado, como Naive Bayes e Support Vector Machine, para classificar bugs.
- Dados textuais: A maioria dos estudos depende de informações em texto (não estruturadas), como descrições de bugs.
- Origem dos dados: Quase todos os dados vêm de projetos de software de código aberto (open-source), como Mozilla e Eclipse.

Um achado crucial é que essa dependência exclusiva de dados de projetos open-source restringe a nossa compreensão sobre como os bugs realmente surgem, se desenvolvem e são corrigidos em empresas e indústrias.

Implicações práticas

Para desenvolvedores e empresas, este mapeamento indica as técnicas mais usadas e eficazes para lidar com bugs. Ao saber que algoritmos como Naive Bayes e Support Vector Machine são predominantes, eles podem focar no uso e aprimoramento dessas ferramentas. No entanto, o estudo também aponta que o conhecimento atual está muito focado em projetos de código aberto. Isso significa que as estratégias e ferramentas podem não ser totalmente aplicáveis ou otimizadas para o ambiente corporativo, onde os tipos de bugs, a forma como são relatados e o ciclo de vida podem ser diferentes.

Limitações e próximos passos

A principal limitação identificada pelos pesquisadores é que o foco quase exclusivo em dados de projetos de código aberto impede que se tenha uma visão completa do ciclo de vida dos bugs em empresas. O paper não detalha exemplos específicos de como isso se manifesta. Por isso, os pesquisadores sugerem que futuras pesquisas explorem os ambientes corporativos.

Além disso, eles veem a necessidade de integrar "modelos híbridos", que combinem dados estruturados (como códigos de erro específicos ou métricas de desempenho) com dados não-estruturados (como as descrições em texto). Essa abordagem mais completa pode oferecer uma reflexão mais fiel sobre a vida dos bugs em diferentes contextos.

QUEM SÃO OS PESQUISADORES

Quem são os pesquisadores

Os pesquisadores por trás deste mapeamento sistemático são Igor Ramsés Temóteo dos Santos, Laís Carvalho Coutinho, Samuel Sousa Teles, Rubens Abraão da Silva Sousa, Manoel Lopes Filho, Ronaldo T. P. Milfont e Alex Ramos. Todos são vinculados ao IFCE (Instituto Federal de Educação, Ciência e Tecnologia do Ceará). O paper não detalha a afiliação específica de cada autor dentro do IFCE, nem seus papéis individuais no estudo ou sua trajetória profissional.

PALAVRAS-CHAVE

Software · Scopus · Context (archaeology) · Software development · Bayesian probability