

CONSTRUCTION OF AN ARTIFICIAL INTELLIGENCE- BASED MODEL FOR C LANGUAGE CODE DEFECT LOCALIZATION

DAVI MANOEL BERNARDES
FELIPE COSTA UNSONST
FELIPE QUITES LOPES



Dados do Artigo

Título: Construction of an Artificial Intelligence-Based Model for C Language Code Defect Localization

Autores: Guoyong Wang

Ano: 2024

Publicado: 5th International Conference on Electronic Communication and Artificial Intelligence

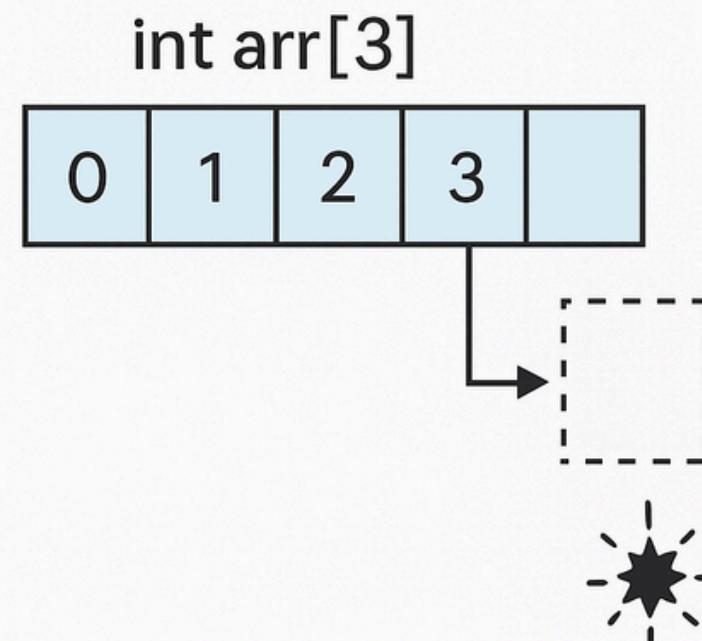


Introdução

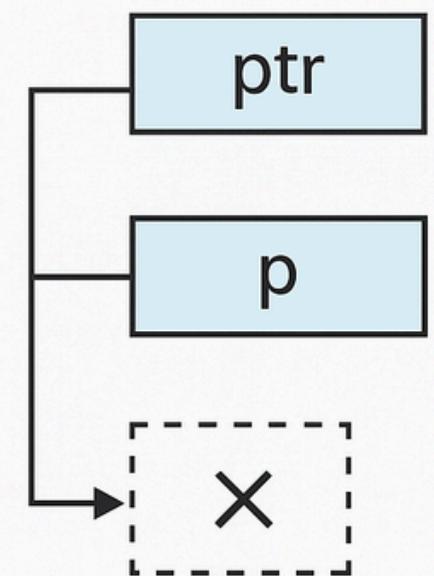
- O crescimento da complexidade dos softwares torna a inspeção manual ineficiente.
- A linguagem C é crítica em sistemas de baixo nível — qualquer defeito pode comprometer segurança e confiabilidade.
- Métodos tradicionais (testes manuais, regras estáticas) falham em detectar falhas sutis.
- Objetivo: automatizar e aprimorar a detecção de defeitos em C com técnicas de IA.



CWE-119 Buffer Overflow



CWE-416 Use of Released Memory



Problemas em C

Defeitos em C se dividem em: sintaxe, semântica, lógica e gerenciamento de recursos.

- CWE-119: Buffer Overflow
- CWE-416: Use of Released Memory

díficeis de detectar e frequentemente levam a falhas de segurança



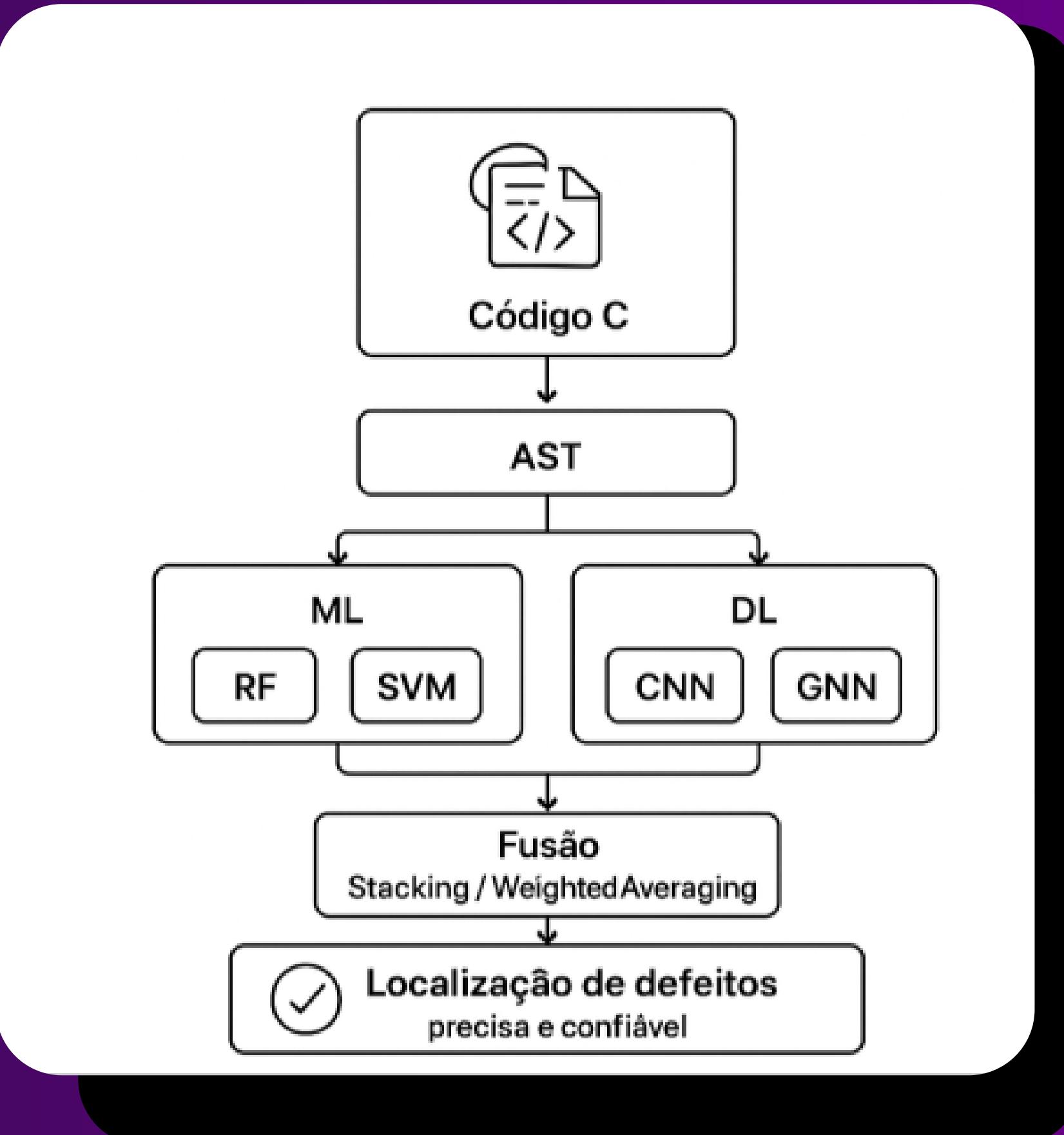
Trabalhos Relacionados

- Métodos anteriores baseados apenas em ML (ex.: Random Forest, SVM) dependem de features manuais.
- Modelos de DL (CNN, GNN) melhoram a precisão, mas têm custo computacional alto e pouca interpretabilidade.
- O artigo propõe um modelo híbrido, combinando ML e DL, superando as limitações individuais.

Metodologia

Etapas:

- 1 - Pré-processamento: conversão do código em ASTs (Árvores de Sintaxe Abstrata).
- 2 - Extração de features (para ML) + aprendizado estrutural (para DL).
- 3 - Modelos individuais:
 - ML: Random Forest, SVM.
 - DL: CNN, GNN (Graph Neural Network).
- 4 - Fusão de modelos (Stacking / Weighted Averaging): combinação das previsões para gerar um resultado final mais robusto.



Pontos Importantes

- Testado em datasets públicos: PROMISE e CGD.
- GNNs tiveram melhor desempenho individual ($F1 \approx 0.92$).
- Modelo híbrido (fusão) superou todos, com F1 Score até 0.91 (melhor equilíbrio entre precisão e recall).
- A fusão aumentou a performance em até 5 p.p. em relação aos modelos únicos





Discussão e implicações

- O modelo reduz falsos positivos e melhora a confiança na detecção.
- Pode ser integrado em IDEs para feedback em tempo real.
- Desafio: interpretabilidade – explicar por que o modelo marca uma linha como defeituosa.
- Caminho futuro: mecanismos de atenção e visualização de risco em ferramentas como Codemap.



Conclusão

- A abordagem híbrida une o melhor do ML e DL.
- Resulta em detecção mais precisa, confiável e automatizada.
- Promove segurança e qualidade no desenvolvimento em C.
- Próximos passos: integração prática e interpretabilidade.

Destaques

DAVI

Achei muito interessante a ideia de treinar uma IA para aprender e analisar uma linguagem específica ao invés de utilizar uma LLM já existente

FELIPE C

A parte que mais me chamou atenção foi a ideia do artigo de juntar duas abordagens de aprendizado de máquina para criar um mais eficiente

FELIPE Q

Um destaque interessante é a IA poder ser utilizada nas IDEs, dessa forma ela ajuda o desenvolvedor durante o processo de criação, ataca os problemas na base

Referências

Wang, G. (2024). Construction of an Artificial Intelligence-Based Model for C Language Code Defect Localization.

Proceedings of the 5th International Conference on Electronic Communication and Artificial Intelligence (ICECAI), IEEE.

OBRIGADO!

DAVI MANOEL BERNARDES

FELIPE COSTA UNSONST

FELIPE QUITES LOPES

