Universidade Federal de Pernambuco

Centro de Informática

Graduação em Sistemas de Informação

LLMs para Detecção e Reparo de Vulnerabilidades em Código: Um Survey com Especialistas e Análise Exploratória de Tendências

Trabalho de Graduação

Aluno: Gabriel Santana Fontanari

Orientadora: Carla Silva

Recife, Agosto de 2025

Universidade Federal de Pernambuco
Centro de Informática

Gabriel Santana Fontanari

LLMs para Detecção e Reparo de Vulnerabilidades em Código: Um Survey com Especialistas e Análise Exploratória de Tendências

Trabalho de Graduação apresentado ao curso de Sistemas de Informação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Carla Silva

Recife
2025

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Fontanari, Gabriel Santana.
LLMs para Detecção e Reparo de Vulnerabilidades em Código: Um Survey com Especialistas e Análise Exploratória de Tendências / Gabriel Santana Fontanari. - Recife, 2025.
77 p. : il., tab.

Orientador(a): Carla Taciana Lima Lourenco Silva Schuenemann
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Informática, Sistemas de Informação - Bacharelado, 2025.
Inclui referências, apêndices.

1. Modelos de Linguagem de Grande Escala. 2. LLMs. 3. Detecção de Vulnerabilidades. 4. Reparo de Vulnerabilidades.. I. Schuenemann, Carla Taciana Lima Lourenco Silva . (Orientação). II. Título.

000 CDD (22.ed.)

Universidade Federal de Pernambuco
Centro de Informática

Gabriel Santana Fontanari

LLMs para Detecção e Reparo de Vulnerabilidades em Código: Um Survey com Especialistas e Análise Exploratória de Tendências

Trabalho de Graduação apresentado ao curso de Sistemas de Informação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovado em: 13/08/2025

BANCA EXAMINADORA

Profa. Carla Taciana Lima Lourenco Silva Schuenemann (Orientadora)
Universidade Federal de Pernambuco

Prof. Kiev Santos da Gama (Examinador Interno)
Universidade Federal de Pernambuco

Resumo

O avanço significativo do potencial dos modelos de linguagem de grande escala (LLMs) tem resultado na adoção do seu uso em várias áreas da Engenharia de Software, o que inclui detecção e reparo de vulnerabilidades. Nesse contexto, este trabalho se propõe a corroborar ou confrontar com os achados, limitações e oportunidades futuras da Revisão Sistemática da Literatura (RSL) "Large Language Model for Vulnerability Detection and Repair: Literature Review and the Road Ahead" realizada por Zhou et al., 2024 [5]. Para isso foi feita uma triangulação, combinando a RSL, um survey conduzido com 10 especialistas de diversos países e uma análise exploratória de 23 artigos recentes e não considerados na RSL de Zhou et al. 2024.

Palavras-chave: Modelos de Linguagem de Grande Escala, LLMs, Detecção de Vulnerabilidades, Reparo de Vulnerabilidades.

Abstract

The significant advancement in the potential of large language models (LLMs) has resulted in their adoption across various areas of Software Engineering, including vulnerability detection and repair. In this context, this work aims to corroborate or contrast the findings, limitations, and future opportunities of the Systematic Literature Review (SLR) “Large Language Model for Vulnerability Detection and Repair: Literature Review and the Road Ahead” conducted by Zhou et al., 2024 [5]. To achieve this, a triangulation was performed, combining the SLR, a survey conducted with 10 experts from different countries, and an exploratory analysis of 23 recent articles not considered in Zhou et al.’s 2024 SLR.

Key-words: Large Language Models, LLMs, Vulnerability Detection, Vulnerability Repair.

Lista de Figuras

Lista de Tabelas

Sumário

1. INTRODUÇÃO

Vulnerabilidade de software se refere a um ponto fraco ou falha em um sistema digital, que pode ser explorada por usuários maliciosos. Este cenário vem se mostrando mais complexo e perigoso a cada ano, com os atores de ameaças cada vez mais bem-equipados e preparados, utilizando táticas, técnicas e ferramentas mais sofisticadas. A escala do problema é notável: a Microsoft, por exemplo, enfrenta mais de 600 milhões de ataques cibercriminosos diariamente [1]. Além disso, tem-se observado uma tendência preocupante em que os atacantes buscam explorar novas vulnerabilidades divulgadas de forma rápida, em vez de se concentrarem apenas em falhas mais antigas e conhecidas. Dados recentes indicam que vulnerabilidades reportadas em 2023 e 2022 foram responsáveis por 6% e 14% de todas as tentativas de exploração, respectivamente. Em comparação, vulnerabilidades relativamente novas, divulgadas entre 2021 e 2023, representaram mais de 30% das tentativas de exploração, um aumento significativo em relação aos 17% observados em 2021 para vulnerabilidades divulgadas entre 2019 e 2021 [2]. Essa mudança reflete que as vulnerabilidades mais recentes são consideradas mais severas e fáceis de serem exploradas

Nos últimos anos o uso de Inteligência Artificial (IA), principalmente de modelos de linguagem de grande escala, do inglês Large Language Models (LLMs), vem se intensificando e se faz presente em diversas tarefas. Inicialmente, sua aplicação se destacou no processamento de linguagem natural (PLN), mas rapidamente se expandiu para outros domínios, incluindo a Engenharia de Software (ES) [3]. Em ES, os LLMs têm sido empregados em uma vasta gama de atividades, como geração de código, refatoração, teste de software, documentação e sumarização de código, demonstrando um potencial transformador na produtividade e qualidade do desenvolvimento [4]. A capacidade de compreender e gerar código-fonte, bem como de raciocinar sobre lógicas de programação, posiciona os LLMs como ferramentas poderosas para enfrentar desafios complexos, como a identificação e correção automática de falhas de segurança.

1.1 MOTIVAÇÃO

Nesse contexto de crescente complexidade das vulnerabilidades e da ascensão da Inteligência Artificial, o emprego de LLMs tem se mostrado uma abordagem promissora para auxiliar na detecção e no reparo dessas falhas. Reconhecendo a importância e o rápido

avanço dessa área, Zhou et al. [5] realizaram uma Revisão Sistemática da Literatura (RSL) intitulada "Large Language Model for Vulnerability Detection and Repair: Literature Review and the Road Ahead". Este estudo consolidou o conhecimento existente até março de 2024, sumarizando os tipos de LLMs utilizados, categorizando as técnicas de adaptação empregadas na detecção e reparo de vulnerabilidades, e identificando as limitações dos estudos até então. Além disso, Zhou et al. propuseram um roteiro (roadmap) com oportunidades futuras de pesquisa, visando guiar o avanço do campo.

Apesar da relevância e atualidade da RSL de Zhou et al., as capacidades e usos de LLMs vem crescendo de forma extremamente dinâmica, o que impacta diretamente no seu uso para detectar e reparar vulnerabilidades em código. Novos modelos, técnicas e aplicações vêm surgindo em um ritmo acelerado e essa evolução contínua levanta a questão da persistência da validade e da relevância dos achados de uma RSL em um curto espaço de tempo.

1.2 OBJETIVOS

O objetivo geral deste trabalho é fazer um estudo exploratório da literatura e da prática na indústria a partir dos resultados da RSL de Zhou et al.[5] visando confirmar, refutar e/ou incrementar os resultados.

Para atingir esse propósito, foram definidos os seguintes objetivos específicos:

- Elaborar e aplicar um survey direcionado a pesquisadores e profissionais da área, para coletar dados sobre a validação dos achados, a persistência das limitações e a relevância das oportunidades futuras apontadas pela RSL original.
- Executar uma análise exploratória da literatura recente (artigos publicados após março de 2024 que citaram Zhou et al.[5]) para complementar os dados do survey com tendências emergentes.
- Analisar e sintetizar os dados coletados no survey e na análise exploratória, confrontando-os com os resultados da RSL de Zhou et al. [5] para gerar uma visão atualizada sobre o estado da arte no uso de LLMs para detecção e reparo de vulnerabilidades.

1.3 QUESTÕES DA PESQUISA

Para atingir os objetivos deste trabalho, as Questões da Pesquisa (QP) de Zhou et al. [5] foram adaptadas:

- ***QP1:*** *Qual a concordância de pesquisadores e profissionais com as arquiteturas de LLMs (encoder-only, decoder-only, encoder-decoder) e modelos específicos (CodeBERT, CodeT5, GPT-3.5/4) mais utilizados para detecção e reparo de vulnerabilidades, conforme identificado no artigo?*
- ***QP2:*** *Quais das técnicas de adaptação de LLMs (fine-tuning, prompt engineering, retrieval augmentation) para detecção e reparo de vulnerabilidades são consideradas mais promissoras ou eficazes na prática pelos especialistas?*
- ***QP3:*** *Qual a relevância e o impacto percebidos das limitações atuais identificadas no artigo (ex: granularidade de entrada, qualidade dos dados, desempenho, uso de LLMs leves, falta de consideração de implantação, acurácia e robustez) na pesquisa e prática da segurança de software?*
- ***QP4:*** *Quais das direções futuras e oportunidades de pesquisa propostas no roadmap do artigo são consideradas mais críticas ou prioritárias para o avanço da área?*

1.4 ESTRUTURA DO TRABALHO

Este trabalho está estruturado da seguinte forma:

- **Capítulo 1 - Introdução:** Apresenta o contexto geral sobre o cenário recente referente a vulnerabilidades de software e ascensão de LLMs juntamente com seu potencial para detectar e corrigir vulnerabilidades de código. Apresenta também a motivação para este estudo, os objetivos e as questões de pesquisa que guiarão o trabalho.
- **Capítulo 2 - Referencial Teórico:** Aborda os conceitos fundamentais necessários para a compreensão do estudo, como os próprios LLMs e sua aplicação em segurança de código.
- **Capítulo 3 - Metodologia de Pesquisa:** Detalha os métodos utilizados no trabalho, incluindo a abordagem inicial de atualização da RSL, a condução do survey com especialistas e a análise exploratória da literatura recente.
- **Capítulo 4 - Resultados do Survey:** Apresenta e analisa os dados coletados no survey de validação com pesquisadores e profissionais da área.
- **Capítulo 5 - Análise Exploratória da Literatura:** Complementa os dados do survey com uma análise de artigos recentes para identificar tendências emergentes.

- **Capítulo 6 - Discussão dos Resultados:** Realiza uma análise comparativa, cruzando os dados da RSL original, do survey e da análise da literatura para extrair conclusões consolidadas.
- **Capítulo 7 - Considerações Finais:** Encerra o trabalho, apresentando as conclusões gerais do estudo e sugerindo direções para trabalhos futuros.

2. REFERENCIAL TEÓRICO

2.1 INTRODUÇÃO

Este capítulo tem como objetivo apresentar os fundamentos teóricos essenciais para a compreensão do presente trabalho. Serão abordados os conceitos relacionados a Modelos de Linguagem de Grande Escala (LLMs) e sua aplicação no domínio de detecção reparo de vulnerabilidades em código

2.2 MODELOS DE LINGUAGEM DE GRANDE ESCALA

Modelos de linguagem de grande escala (LLMs) são redes neurais profundas, tipicamente baseadas na arquitetura Transformers, que são pré-treinadas com enormes volumes de dados textuais. Esse pré-treinamento extenso permite que os modelos de LLM tenham capacidade de compreender, gerar e manipular linguagem natural e, crucialmente para este trabalho, código-fonte [3].

A arquitetura Transformer, base dos LLMs modernos, é composta por dois módulos principais: o encoder (codificador) e o decoder (decodificador). Dependendo da tarefa, os LLMs podem utilizar um, outro ou ambos os módulos, resultando em três tipos principais de arquitetura:

- **Encoder-only:** São especializados em "ler" e compreender profundamente o contexto de um texto de entrada para criar uma representação numérica rica em significado. São ideais para tarefas de análise e classificação, como identificar o sentimento de uma frase e classificar em categorias pré estabelecidas.
- **Decoder-only:** São especializados em gerar texto. Eles recebem uma sequência de entrada (prompt) e preveem a continuação mais provável, o que os torna excelentes para tarefas de geração de texto e código.
- **Encoder-Decoder:** Utiliza ambos os módulos. O encoder primeiro processa e compreende toda a sequência de entrada. Em seguida, o decoder usa essa compreensão para gerar uma nova sequência de saída. Essa arquitetura é ideal para tarefas de tradução ou sumarização.

Dados de treinamento são um conjunto de informações rotuladas, ou seja, dados que foram previamente anotados ou marcados com a resposta ou categoria correta que o modelo

deve aprender a prever, que alimentam os algoritmos de IA e permitem que eles tenham a capacidade de identificar os padrões presentes nesses dados. A denominação de “modelos de linguagem de grande escala”, se dá pelo fato de que o treinamento desses modelos envolve bilhões de parâmetros, o que permite que eles possuam capacidades emergentes de raciocínio e generalização que os tornam ferramentas poderosas em diversos domínios, incluindo a ES.

Em ES, os LLMs têm sido empregados em uma vasta gama de atividades, como geração automática de código, refatoração, teste de software, documentação e sumarização de código, demonstrando um potencial transformador na produtividade e qualidade do desenvolvimento [4]. A capacidade de compreender e gerar código-fonte, bem como de raciocinar sobre lógicas de programação, posiciona os LLMs como ferramentas poderosas para enfrentar desafios complexos.

2.3 DETECÇÃO DE VULNERABILIDADES COM LLMS

A detecção de vulnerabilidades de software é uma tarefa crítica que visa identificar falhas ou pontos fracos no código que podem ser explorados por agentes maliciosos. Tradicionalmente, essa tarefa é realizada por meio de técnicas como análise estática (baseada em regras ou padrões), análise dinâmica (execução de código para encontrar falhas) e testes de fuzzing. No entanto, essas abordagens frequentemente enfrentam desafios como altas taxas de falsos positivos, dificuldade em lidar com a complexidade de códigos modernos e a incapacidade de detectar vulnerabilidades em diversos tipos de falhas [5].

Com a ascensão dos LLMs, a detecção de vulnerabilidades tem sido reformulada, tipicamente como um problema de classificação binária, ou seja, um problema cuja resolução pode ter duas classes, como por exemplo “possui vulnerabilidade” ou “não possui vulnerabilidade”. Nesse cenário, um LLM recebe um trecho de código (que pode variar em granularidade, como linha, função ou até mesmo repositório) e prediz se ele contém ou não uma vulnerabilidade [5]. A eficácia dos LLMs nessa tarefa reside em sua habilidade de aprender padrões complexos e sutis associados a vulnerabilidades a partir de grandes conjuntos de dados, superando, em alguns aspectos, as limitações das técnicas tradicionais. LLMs com arquiteturas do tipo encoder-only, como CodeBERT e GraphCodeBERT, têm sido predominantemente utilizados para essa finalidade [5].

2.4 REPARO DE VULNERABILIDADES COM LLMS

O reparo de vulnerabilidades, por sua vez, é a tarefa de corrigir as falhas de segurança identificadas no código-fonte. Este processo é complexo e demanda um profundo entendimento tanto da vulnerabilidade quanto da lógica do programa para garantir que a correção seja eficaz e não introduza novos defeitos. Ferramentas de reparo automatizado tradicionais enfrentam dificuldades em lidar com a diversidade de tipos de vulnerabilidades e a complexidade de bases de código reais [5].

No contexto dos LLMs, o reparo de vulnerabilidades é geralmente abordado como um problema de sequência-para-sequência (Seq2Seq), ou seja, recebe uma sequência de caracteres e com base nela, devolve outra sequência. No contexto de reparo de vulnerabilidade, o modelo recebe um fragmento de código vulnerável e gera como saída o código reparado correspondente [5]. Modelos do tipo decoder-only e encoder-decoder, incluindo LLMs comerciais como GPT-3.5, GPT-4 e CodeLlama, têm se mostrado mais proeminentes para essa tarefa, dada sua capacidade de gerar sequências de texto (código) de forma coerente e contextualizada [5]. A aplicação de LLMs no reparo automatizado de vulnerabilidades promete acelerar o ciclo de desenvolvimento seguro e reduzir o esforço manual.

2.5 TÉCNICAS DE ADAPTAÇÃO DE LLMS PARA DETECÇÃO E REPARO DE VULNERABILIDADES

Para otimizar o desempenho dos LLMs em tarefas específicas de detecção e reparo de vulnerabilidades, diversas técnicas de adaptação são empregadas. Embora possam ser usadas de forma isolada, comumente essas técnicas são combinadas para criar abordagens híbridas mais eficientes. O fine-tuning, por exemplo, modifica o conhecimento interno do modelo, enquanto o prompt engineering e o RAG aprimoram seu comportamento no momento da execução. A seguir, são descritas as técnicas mais comuns e como elas funcionam:

- **Fine-tuning:** Consiste em ajustar os parâmetros de treinamento do modelo. Para isso, o modelo é alimentado com dados de treinamento rotulados e específicos para a tarefa de detecção e reparo de vulnerabilidades
- **Prompt Engineering:** Criação e refinamento dos “prompts”, que são as instruções textuais que guiam as tarefas da LLM. Visa tornar as instruções mais claras, para uma melhor compreensão e execução da LLM. De acordo com Zhou et al, existem diferentes abordagens para essa técnica. São elas: zero-shot (sem exemplos); few-shot (com poucos exemplos); e chain-of-thought (passo a

passo de raciocínio) [5]

- **Retrieval Augmented Generation (RAG):** Integra ao contexto da LLM uma base extensa de informações (bases de conhecimento, documentos relevantes, exemplos de código, entre outros). A resposta formulada pela LLM passa a se basear nas informações presentes nesse contexto adicional. O que gera uma resposta mais completa e com menor índice de alucinação (refere-se às informações trazidas pela LLM que não tem base real, ou seja, são "inventadas")
- **Domain-specific Pre-training:** Envolve pré-treinar um LLM em um vasto conjunto de dados específico do domínio de segurança de software (e.g., código com vulnerabilidades, commits de correção), antes de realizar o fine-tuning para a tarefa final. Isso ajuda o modelo a aprender representações mais relevantes para o domínio [5].
- **Combinação com Análise de Programa e Outros Módulos de DL:** LLMs podem ser combinados com técnicas de análise de programa (e.g., AST, CFG, DFG) para incorporar informações estruturais do código, ou com outros módulos de Deep Learning (e.g., GNNs, Bi-LSTMs) para superar limitações inerentes à arquitetura Transformer [5]

2.6 SÍNTESE DO CAPÍTULO

Este capítulo apresentou os fundamentos teóricos necessários para compreender o uso de LLMs na detecção e reparo de vulnerabilidades em código-fonte. Explorando desde a arquitetura por trás do LLM até as técnicas de adaptação para um melhor funcionamento nesse contexto.

3. METODOLOGIA DE PESQUISA

3.1 INTRODUÇÃO

Esta seção descreve a metodologia de pesquisa adotada para este Trabalho. Inicialmente, o estudo propôs a atualização da RSL de Zhou et al. [5]. Contudo, devido a limitações de tempo e recursos, o escopo foi redefinido para uma validação qualitativa dos achados da RSL original, complementada por uma análise exploratória da literatura recente. Detalha-se, a seguir, a abordagem inicial e os fatores que levaram à sua redefinição, bem como a metodologia da nova abordagem

3.2 ABORDAGEM INICIAL: TENTATIVA DE ATUALIZAÇÃO DA RSL E SUA INVIABILIDADE

O objetivo inicial deste trabalho de graduação era realizar uma atualização completa da RSL conduzida por Zhou et al. [5], focando no uso de LLMs para detecção e reparo de vulnerabilidades em código. A necessidade e a relevância dessa atualização foram amplamente justificadas e validadas, conforme detalhado na subseção a seguir.

3.2.1 VALIDAÇÃO DA NECESSIDADE DA ATUALIZAÇÃO

De acordo com Mendes et al. [8], para se validar a necessidade de uma atualização em uma RSL, é recomendado utilizar algum framework para estruturar o processo de análise. Para avaliar a necessidade de atualização da RSL conduzida por Zhou et al. [5], foi adotado o método recomendado por Mendes e originalmente proposto por Garner et al. [9], denominado 3PDF (Third-Party Decision Framework)

O 3PDF, que significa literalmente “Framework de Decisão por Terceiros”, é indicado especialmente para casos em que a atualização da RSL é realizada por uma equipe de pesquisadores diferente da original. O método consiste em um processo de decisão estruturado em 3 etapas, com o objetivo de garantir que uma atualização só seja conduzida quando for realmente necessária.

As etapas desse processo estão ilustradas na Figura 1, a seguir:

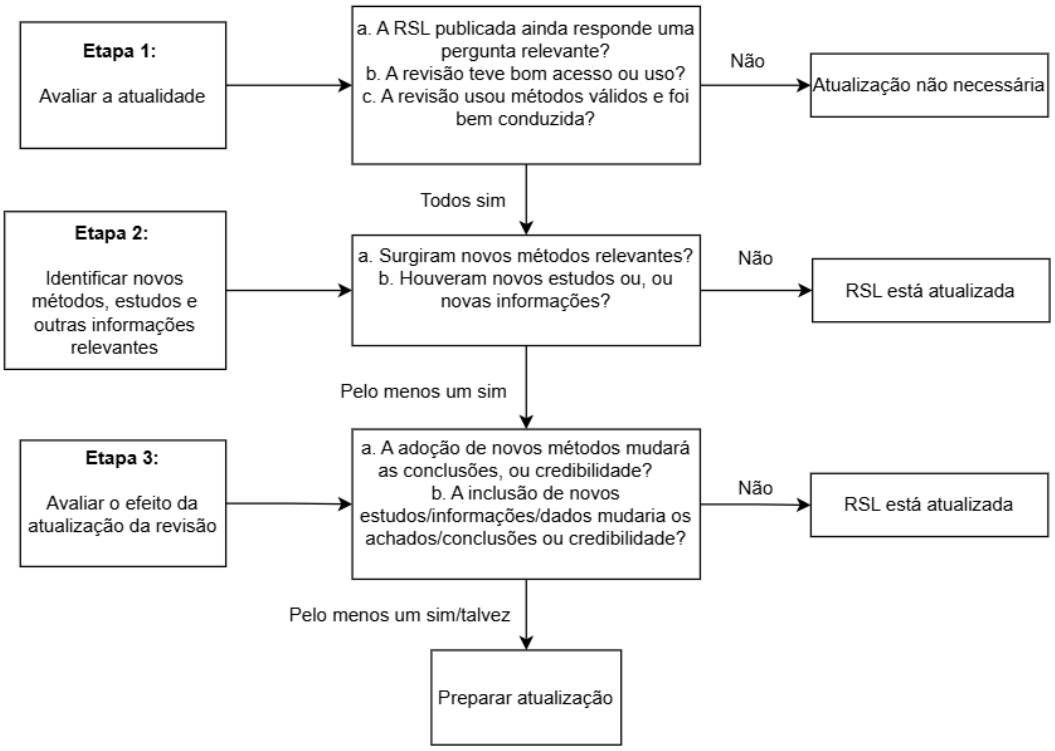


Figura 1. Framework de Decisão para Atualização de uma RSL
Fonte: Garner et al. [9]

3.2.2 RESULTADO DA ANÁLISE DE VALIDADE

O framework proposto foi aplicado a RSL [5], com o objetivo de verificar se a revisão publicada permanece atual ou se existe necessidade de atualização diante de novas evidências. Resultados são detalhados a seguir:

Etapa 1 – Avaliação da atualidade da RSL

a) A RSL publicada ainda responde uma pergunta relevante?

Sim. A RSL publicada por Zhou et al. em março de 2024 aborda um tema altamente atual: o uso de LLMs na detecção e reparo de vulnerabilidades em código-fonte. A pergunta de pesquisa permanece relevante, visto que o uso de LLMs em Engenharia de Software continua crescendo e os modelos estão cada vez mais robustos, trazendo constantemente novas aplicações.

b) A revisão teve bom acesso ou uso?

Sim. Apesar de recente, a RSL teve boa visibilidade, com cerca de 77 citações encontradas através do Google Scholar. Além disso, foi publicada em ACM TOSEM, que é reconhecido como um dos principais e mais influentes periódicos na área de Engenharia de Software

c) A revisão usou métodos válidos e foi bem conduzida?

Sim, a RSL foi conduzida com rigor metodológico, utilizando critérios claros para a inclusão, exclusão e avaliação de qualidade dos estudos analisados, o que resultou em um conjunto primário contendo 58 estudos. A revisão foi publicada em veículo qualificado e apresenta boa estrutura, o que reforça sua validade metodológica e científica.

Etapa 2 – Identificação de novos métodos ou estudos relevantes

a) Surgiram novos métodos relevantes?

Não.

b) Houveram novos estudos ou, ou novas informações?

Sim. Embora a RSL original contemple artigos publicados até Março de 2024, a área de Large Language Models e suas aplicações em Engenharia de Software, incluindo segurança, é extremamente dinâmica e com rápida evolução. Em uma rápida busca preliminar consultando a base do IEEE e utilizando termos de pesquisa semelhantes aos utilizados na pesquisa original (detalhes das queries em APÊNDICE A), foi possível encontrar 68 artigos disponibilizados na base de dados entre 2017 e Março de 2024 e 327 encontrados entre Abril de 2024 e Maio de 2025, o que demonstra um crescimento amplo e significativo de pesquisas relacionadas ao tema.

Etapa 2 – Identificação de novos métodos ou estudos relevantes

a) A adoção de novos métodos mudará as conclusões, ou credibilidade?

Não. O método adotado é o mesmo usado pelo autor da RSL original

b) A inclusão de novos estudos/informações/dados mudaria os achados/conclusões ou credibilidade?

Talvez. Não há literatura suficiente para dizer com certeza.

3.2.3 TENTATIVA DE ATUALIZAÇÃO DA RSL: METODOLOGIA, EXECUÇÃO E REDEFINIÇÃO DE ESCOPO

A metodologia inicialmente planejada para a atualização da RSL seguia as diretrizes de Wohlin et al. [6] e consistia nos seguintes passos:

- Período da Busca: De abril de 2024 até o presente momento.
- Conjunto Semente: A RSL de Zhou et al. [5] e seus 58 estudos primários (lista extraída da seção de referências).
- Ferramenta Principal: Google Scholar.
- Técnica: Forward Snowballing

O forward snowballing é uma técnica de busca de estudos em RSLs, que parte de um conjunto inicial de artigos (o "conjunto semente") e expande a busca para artigos mais recentes que citam esses estudos. É como seguir a trilha de um trabalho para descobrir quais pesquisas posteriores se basearam nele. Para atualizações de RSLs, essa técnica é considerada eficiente, pois presume-se que artigos relevantes mais novos que continuam a linha de pesquisa citam a RSL original ou seus estudos primários.

Os passos adicionais da metodologia inicial incluíam:

- Critérios de Inclusão/Exclusão: Utilizar os critérios da RSL original de Zhou et al. [5], ajustando se necessário para o período da atualização.
- Processo de Seleção: Realizar triagem inicial de títulos e resumos

A execução dessa metodologia, contudo, revelou a sua inviabilidade dentro do prazo e

recursos disponíveis para este TCC. Para realização do forward snowballing e triagem inicial, foram realizados os seguintes passos:

- Criação de um documento (`Conjunto Inicial.csv`) com os 58 estudos primários originais da RSL de Zhou et al. (2024), contendo um ID para o artigo, o nome e o link do Google Scholar. Documento pode ser encontrado no APÊNDICE E
- Criação de um script para acessar o link do Google Scholar e realizar o forward snowballing para cada estudo do `Conjunto Inicial.csv`, identificando artigos citantes publicados a partir de abril de 2024 e salvando seus nomes e links. Este script encontrou 2904 artigos
- Desenvolvimento de um script de deduplicação para remover artigos encontrados mais de uma vez, resultando em 1324 artigos únicos.
- Criação de um script para acessar os links dos 1324 artigos únicos e salvar seus resumos. Foi possível obter os resumos de 1162 artigos.
- Desenvolvimento de um script para enviar cada linha da planilha de artigos (título e resumo) para a API do Gemini, contextualizando-a com os critérios de inclusão e exclusão do artigo original de Zhou et al. [5]. O objetivo foi obter dois valores: `Probabilidade_de_inclusao` (indicando a probabilidade do artigo ser relevante para a atualização) e `Justificativa` (explicando a probabilidade).

Com a aplicação deste script, foram obtidos os seguintes resultados preliminares de probabilidade de inclusão:

- 183 artigos com alta probabilidade de inclusão.
- 43 artigos com média probabilidade de inclusão.
- 1099 artigos com baixa probabilidade de inclusão.

Diante do volume de artigos considerados relevantes (183 de alta e 43 de média probabilidade, totalizando 226 artigos), e considerando a necessidade de uma análise manual rigorosa (leitura completa, avaliação de qualidade, extração de dados) para cada um deles, somado à limitação de tempo (dois meses) e ao fato de o trabalho ser executado individualmente (ausência de revisão por pares), a conclusão de uma RSL atualizada com o rigor metodológico necessário foi considerada inviável dentro do prazo estabelecido. Esta análise preliminar demonstrou o esforço necessário para uma RSL completa e a inviabilidade

de sua conclusão.

3.3 SEGUNDA ABORDAGEM: VALIDAÇÃO QUALITATIVA DOS ACHADOS DA RSL ORIGINAL

Diante da redefinição do escopo, esse trabalho passará a focar em uma validação qualitativa da RSL original de Zhou et al. [5] realizada por meio de um survey. Essa nova abordagem permitirá explorar a relevância e a persistência dos achados e limitações originais no contexto do avanço recente da área, de forma mais alinhada com as restrições de tempo e recursos.

3.3.1 OBJETIVO DO SURVEY

O objetivo principal da condução desse survey é realizar uma análise crítica e qualitativa dos principais achados, conclusões, limitações e oportunidades identificados na RSL original de Zhou et al. [5]. Visa-se verificar a validade e relevância desses pontos no cenário atual de LLMs para a detecção e reparo de vulnerabilidades em código.

3.3.2 METODOLOGIA DO SURVEY

A condução do survey de validação dos achados da RSL original foi delineada e executada seguindo explicitamente as diretrizes e o checklist para surveys em Engenharia de Software proposto por Molléri, Petersen e Mendes [7]. Esta abordagem garante rigor e sistematicidade em todas as fases do processo, desde o planejamento até a análise, maximizando a confiabilidade dos resultados obtidos, mesmo com as limitações inerentes a um trabalho individual. Detalhes sobre o design, planejamento, coleta e análise dos dados do survey serão apresentados no Capítulo 4, onde os resultados obtidos serão explorados em profundidade.

3.3.3 DESIGN DO SURVEY

O planejamento do survey é uma etapa fundamental para garantir que os dados coletados sejam relevantes e confiáveis para responder às questões de pesquisa apresentadas no tópico 1.3. Esta fase engloba a definição detalhada do instrumento de coleta, a caracterização da população-alvo e a estratégia de amostragem, bem como as considerações

éticas, alinhando-se às melhores práticas em pesquisa empírica em Engenharia de Software [10].

3.3.3.1 EXTRAÇÃO DE ACHADOS DA RSL ORIGINAL

O ponto de partida para a formulação do questionário do survey foi uma releitura cuidadosa e sistemática da RSL "Large Language Model for Vulnerability Detection and Repair"[5]. As Questões de Pesquisa da RSL original foram reaproveitadas e adaptadas para o contexto do survey. Os principais achados, conclusões e limitações foram identificados. Esses elementos constituíram a base conceitual para a criação das afirmações e perguntas que compõem o survey, garantindo que a validação fosse diretamente focada nos pontos centrais da RSL de referência.

3.3.3.2 FORMULAÇÃO DO QUESTIONÁRIO

O questionário foi cuidadosamente formulado para garantir que as perguntas fossem claras, objetivas e capazes de capturar a percepção dos especialistas sobre os achados e limitações da RSL [5] no contexto atual. O processo de criação seguiu as seguintes diretrizes:

- **Idioma de apresentação:** O survey foi formulado em inglês, com o intuito de aumentar o público alvo e a respectiva adesão.
- **Tipos de Perguntas:** Foram utilizados predominantemente perguntas fechadas com escalas Likert de 5 pontos para avaliar a percepção dos participantes (e.g., concordância, persistência de limitações, relevância e exploração de oportunidades). Adicionalmente, foram incluídas perguntas abertas para permitir que os especialistas forneçam comentários e insights qualitativos mais aprofundados. Perguntas de múltipla escolha também foram usadas para coletar dados demográficos, facilitando a caracterização do perfil dos respondentes. Ao todo, foram elaboradas 27 perguntas que estão detalhados no APÊNDICE B.
- **Relação com a RSL Original:** Cada pergunta do questionário foi diretamente relacionada a um achado, conclusão, limitação ou oportunidade extraída da RSL de Zhou et al. [5], garantindo que o survey valide especificamente os pontos do estudo base.
- **Clareza e Compreensibilidade:** As perguntas foram elaboradas de forma

concisa e com linguagem clara, evitando ambiguidades e jargões excessivos que pudessem dificultar a compreensão por parte dos especialistas, conforme recomendado por Molléri, Petersen e Mendes [7].

3.3.3.3 POPULAÇÃO E AMOSTRAGEM

Para este estudo, a população alvo consiste em *pesquisadores e profissionais que atuam nas áreas de Engenharia de Software ou Segurança de Software e que utilizam Modelos de Linguagem de Grande Escala (LLMs) para auxiliar na detecção e/ou correção de vulnerabilidades em código*. Essa população foi escolhida por ser a mais adequada para fornecer uma validação informada dos achados da RSL de Zhou et al. [5], possuindo conhecimento teórico e/ou prático relevante.

3.3.3.4 INSTRUMENTO DO SURVEY

O survey completo está detalhado no APÊNDICE B. Sua hospedagem foi feita no Google Forms, por ser um plataforma online, de fácil acesso, gratuita e permitir exportação dos dados.

3.3.3.5 TESTE PILOTO

Antes da distribuição em larga escala, um teste piloto do questionário foi realizado com um estudante de mestrado da USP que atua na área. O objetivo foi identificar quaisquer ambiguidades ou falta de clareza nas perguntas, verificar a coerência da lógica das seções e a fluidez do questionário, e receber feedback geral sobre a usabilidade e a relevância das perguntas.

Com base no feedback recebido do estudante de mestrado que realizou o teste piloto, foram implementadas melhorias no questionário do survey para aumentar sua clareza e abrangência. Na questão sobre "Combination with Other Deep Learning Modules" (Q12.c), foi adicionado um campo "Other (please specify)" para permitir que os respondentes especifiquem módulos não listados. As questões referentes à qualidade dos dados (Q19.a e Q19.b) foram reformuladas: "Dependence on heuristic labels and potential noise in detection datasets?" foi ajustada para "Dependence on labels generated by automated rules or tools (heuristic labels), often leading to noise or inaccuracies in detection datasets?" para explicar o

termo "heuristic labels" e o conceito de ruído. De forma similar, "Lack of associated test cases in vulnerability repair datasets?" foi reescrita para "Lack of test cases to verify the correctness of vulnerability fixes and prevent new bugs in repair datasets?" para explicitar o significado de "test cases" nesse contexto. Essas alterações visaram garantir a compreensão das perguntas por parte dos especialistas, otimizando a qualidade dos dados coletados.

3.3.4 COLETA DE DADOS

A fase de execução do survey envolveu a distribuição do instrumento aos participantes e o gerenciamento da coleta de respostas.

3.3.4.1 ESTRATÉGIA DE DISTRIBUIÇÃO

A seleção de participantes do survey foi feita de duas formas. De forma indireta, através de posts em redes sociais (instagram e linkedin). E de forma direta, através do disparo de emails para pesquisadores que se enquadram na população alvo. Os emails dos pesquisadores foram extraídos da RSL original [5] e dos 84 artigos que a citaram até o presente momento. Ao todo, 766 e-mails foram disparados

3.3.4.2 PERÍODO DE COLETA

O formulário ficou disponível para resposta durante 1 mês. O Survey foi aberto no dia 30 de Junho e finalizado no dia 30 de Julho.

3.4 ANÁLISE EXPLORATÓRIA DA LITERATURA RECENTE

Dado as limitações referentes a abrangência do survey e sua adesão limitada, por conta da sua especificidade, é necessário complementar o resultado do survey com dados obtidos a partir de estudos recentes sobre o tema.

3.4.1 OBJETIVO

O objetivo deste estudo é complementar a validação dos achados da RSL original [5] e os resultados do survey através de uma análise exploratória de trabalhos recentes na área de LLMs para detecção e reparo de vulnerabilidades. Visa-se identificar a relevância e

exploração das limitações e oportunidades evidenciadas na RSL e as tendências emergentes que possam corroborar ou contrastar com as percepções dos especialistas, especialmente considerando a natureza dinâmica do campo e a potencial limitação no número de respostas do survey.

3.4.2 PROCEDIMENTO

O processo adotado para a análise exploratória consiste em:

- **Fonte de Dados:** A busca será focada nos estudos primários que citaram a RSL de Zhou et al. [5] no Google Scholar, publicados a partir de abril de 2024.
- **Critério de Seleção:** Será realizada uma leitura exploratória e não sistemática dos títulos e resumos desses artigos. Nessa etapa não haverá leitura completa dos artigos, avaliação de qualidade formal ou aplicação rigorosa de critérios de inclusão/exclusão além da relevância aparente pelo título e resumo.
- **Foco da Análise:** Utilizar o Google NotebookLM para sintetizar os achados, conclusões e metodologia dos artigos para responder de forma indireta as questões de pesquisa. Detalhes do processo de extração dos dados presentes no APÊNDICE D.
- **Natureza da Análise:** Esta análise terá um caráter qualitativo e descritivo, visando fornecer um panorama das tendências mais recentes, e não se configura como uma revisão sistemática da literatura.

3.4.3 SELEÇÃO DOS ARTIGOS

A busca inicial retornou um total de 77 artigos que citavam a RSL. Para a triagem, foi feita a leitura de título e resumo dos artigos encontrados a fim de remover os que apresentavam baixa relevância para o tema. Ao todo foram selecionados 23 estudos primários. Os artigos selecionados estão presentes no APÊNDICE C.

Sobre os artigos descartados, os motivos para a exclusão foram variados: alguns artigos, embora mencionasse LLMs, tinham como foco principal tópicos adjacentes, como o planejamento de ataques, geração de vulnerabilidades ou segurança do próprio sistema LLM, em vez da detecção ou reparo de vulnerabilidades em código. Outros trabalhos não utilizavam LLMs como a tecnologia central, focando em abordagens tradicionais de Deep Learning ou análise de grafos. Artigos com escopo muito amplo, como revisões sobre "LLMs

em Cibersegurança", foram mantidos apenas quando o resumo confirmava uma seção substancial sobre segurança de código. Finalmente, alguns artigos foram descartados por questões práticas, como não serem estudos primários, a impossibilidade de acesso ao texto completo ou por serem publicações de trabalhos já analisados.

3.4.4 ANÁLISE DOS ESTUDOS PRIMÁRIOS

A análise dos 23 estudos primários foi conduzida com auxílio do Google Notebook LM, que resumiu os principais tópicos de cada artigo em relação às questões de pesquisa. O resumo feito utiliza referências diretas ao material original, permitindo checagem das informações resumidas. Essa abordagem permitiu uma extração de dados mais rápida a partir da leitura parcial dos trabalhos, focando nas seções de metodologia e resultados para a checagem dos fatos. É importante ressaltar que os artigos não foram lidos de forma completa, mas sim de maneira direcionada para a extração das informações pertinentes a este estudo. A seguir, são apresentadas as tendências identificadas a partir da análise dos dados consolidados. O processo adotado e o prompt utilizado são detalhados no APÊNDICE D.

3.5 TRIANGULAÇÃO DOS DADOS

Para sintetizar os resultados e fortalecer a validade das conclusões, este trabalho emprega a triangulação, uma abordagem que combina os dados das três fontes de evidência utilizadas: a Revisão Sistemática da Literatura (RSL) original de Zhou et al. [5], o survey conduzido com especialistas e a análise exploratória da literatura recente.

De acordo com Runeson e Höst [11] :

“A triangulação é importante para aumentar a precisão da pesquisa empírica. A triangulação significa adotar diferentes ângulos em relação ao objeto estudado, proporcionando, assim, um panorama mais amplo. A necessidade da triangulação é óbvia quando se baseia principalmente em dados qualitativos, que são mais amplos e ricos, mas menos precisos do que dados quantitativos.”

Neste estudo, a triangulação permitirá confrontar os achados consolidados pela RSL com a percepção prática dos especialistas e com as tendências emergentes identificadas na literatura recente. O objetivo é gerar uma visão mais completa e atualizada sobre o uso de LLMs para detecção e reparo de vulnerabilidades, cujos resultados detalhados serão discutidos no Capítulo 6.

3.6 SÍNTESE DO CAPÍTULO

Este capítulo descreveu a metodologia de pesquisa adotada, detalhando a evolução do escopo do trabalho. Foram apresentados os procedimentos da tentativa inicial de atualização da RSL, que justificou a pivotagem para a nova abordagem. Em seguida, foi detalhada a metodologia geral do survey de validação com especialistas e da análise exploratória da literatura recente, que constituem as principais fontes de dados do estudo. Por fim, foi introduzido o método de triangulação, que será empregado para analisar e sintetizar os dados dessas diferentes fontes, fortalecendo a base analítica para as conclusões deste estudo.

4. RESULTADOS DO SURVEY

Este capítulo apresenta os resultados detalhados da condução do survey de validação e as análises realizadas, bem como a caracterização dos participantes e os achados principais.

4.1 ANÁLISE DOS DADOS DO SURVEY

Após a conclusão da coleta de dados, as respostas do survey foram submetidas a um processo rigoroso de análise para extrair percepções e responder às questões de pesquisa.

4.1.1 PREPARAÇÃO DOS DADOS

Os dados brutos coletados do Google Forms foram exportados e organizados no Google Planilhas. Essa etapa incluiu limpeza dos dados, tratamento de respostas incompletas e padronização nos formatos. As células em branco nas perguntas de escala (onde o respondente não marcou nada por não ter contato com a LLM) foram interpretadas como "sem experiência" ou "não aplicável" para aquele item específico e foram excluídas dos cálculos de média, mas sua contagem foi registrada para análise descritiva.

4.1.2 ANÁLISE QUANTITATIVA

Foram calculadas estatísticas descritivas para as respostas fechadas (escalas Likert e perguntas de múltipla escolha). Isso incluiu frequências, porcentagens e médias para cada item avaliado. Os resultados serão apresentados por meio de gráficos para facilitar a visualização e interpretação.

4.1.3 ANÁLISE QUALITATIVA

A análise das respostas abertas complementa os dados quantitativos, revelando os principais argumentos por trás das percepções dos especialistas. Análise dos dados presente no tópico 4.5.

4.2 CARACTERIZAÇÃO DOS PARTICIPANTES

Esta seção apresentará os dados demográficos e de experiência dos respondentes do

survey. Ao todo, o survey recebeu apenas 10 respostas válidas. Das 10 respostas, 8 foram obtidas a partir do disparo de emails, o que nos dá uma taxa de respostas de 1,04%.

Tabela 1. Tabela de respostas demográficas

Código	Área primária de pesquisa ou trabalho	País	Tempo de experiência em segurança de software	Tempo de experiência com LLMs	Tempo de experiência LLMs aplicadas em segurança de software	Leram a RSL
P1	Cyber-segurança/ Segurança de software, Pesquisa acadêmica	Brasil	1 a 3 anos	6 meses a 1 ano	6 meses a 1 ano	Parcialmente
P2	Cyber-segurança/ Segurança de software, Pesquisa acadêmica	EUA	Mais de 5 anos	1 a 2 anos	1 a 2 anos	Sim
P3	Engenharia de Software (Geral), Desenvolvimento de software	Colômbia	Menos de 1 ano	6 meses a 1 ano	Menos de 6 meses	Não
P4	Cyber-segurança/ Segurança de software, Pesquisa acadêmica	Brasil	3 a 5 anos	1 a 2 anos	6 meses a 1 ano	Não
P5	Engenharia de Software (Geral)	China	1 a 3 anos	1 a 2 anos	6 meses a 1 ano	Sim
P6	Inteligência Artificial / Aprendizado de máquina, Cyber-segurança/ Segurança de software, Pesquisa acadêmica	Perú	Mais de 5 anos	1 a 2 anos	6 meses a 1 ano	Parcialmente
P7	Engenharia de Software (Geral), Inteligência Artificial / Aprendizado de máquina, Cyber-segurança/ Segurança de software, Pesquisa acadêmica, Desenvolvimento de software, Pesquisa acadêmica	EUA	3 a 5 anos	1 a 2 anos	1 a 2 anos	Sim
P8	Cyber-segurança/ Segurança de software, Pesquisa acadêmica	EUA	1 a 3 anos	1 a 2 anos	6 meses a 1 ano	Parcialmente
P9	Inteligência Artificial / Aprendizado de máquina	EUA	1 a 3 anos	1 a 2 anos	6 meses a 1 ano	Não, mas gostaria
P10	Engenharia de Software (Geral), Cyber-segurança/ Segurança de software, Pesquisa acadêmica	Suécia	1 a 3 anos	2 a 3 anos	1 a 2 anos	Parcialmente

Fonte: Dados do Survey

4.2.1 ÁREA PRIMÁRIA DE ESTUDO OU PESQUISA (Q1)

Essa foi uma pergunta de seleção múltipla, ou seja, o respondente poderia selecionar

uma ou mais alternativas. “Cyber-segurança/Segurança de software” foi a mais frequente, com 70%, seguida por “Pesquisa acadêmica” com 50%, “Engenharia de Software (Geral)” com 40%, “Inteligência Artificial / Aprendizado de máquina” com 30%, e “Desenvolvimento de software” com 20%. Não houve respostas na categoria "Outro". Esses dados indicam que a maioria dos respondentes possui forte alinhamento com as áreas centrais da pesquisa, com destaque para segurança de software e pesquisa acadêmica, o que é fundamental para a validade das percepções coletadas.

4.2.2 PAÍS DE ATUAÇÃO PRINCIPAL (Q2)

Os Estados Unidos da América foi o país com maior número de respondentes, representando 40%, seguido pelo Brasil com 20%. Suécia, Colômbia, China e Peru tiveram 10% cada. Essa distribuição geográfica, embora concentrada em alguns países, demonstra um alcance internacional relevante do survey.

4.2.3 TEMPO DE TRABALHO COM SEGURANÇA DE SOFTWARE (DETECÇÃO/REPARO DE VULNERABILIDADE) (Q3)

Em relação ao tempo de trabalho com segurança de software (detecção/reparo de vulnerabilidade), 50% dos participantes possuem de 1 a 3 anos de experiência, 20% têm de 3 a 5 anos, e outros 20% têm mais de 5 anos. Apenas 10% possuem menos de 1 ano de experiência. Isso indica que a amostra é composta por profissionais com experiência consolidada e relevante no domínio de segurança de software.

4.2.4 TEMPO DE TRABALHO COM LLMS (Q4)

Quanto ao tempo de trabalho com LLMs, a experiência é mais recente para a maioria: 70% têm entre 1 e 2 anos de experiência, 10% têm de 2 a 3 anos, e 20% têm de 6 meses a 1 ano. Este dado reflete o crescimento recente da área de LLMs, com muitos profissionais iniciando seu contato nos últimos 2 anos.

4.2.5 TEMPO DE TRABALHO COM LLMS APLICADAS EM SEGURANÇA DE SOFTWARE (DETECÇÃO/REPARO DE VULNERABILIDADE) (Q5)

A aplicação específica de LLMs em segurança é ainda mais recente para a maioria dos respondentes: 60% têm de 6 meses a 1 ano de experiência, 30% têm de 1 a 2 anos, e 10% têm menos de 6 meses. Isso corrobora a natureza emergente do campo de pesquisa e aplicação de LLMs em segurança de software.

4.2.6 LEITURA DA RSL DE ZHOU ET AL. (Q6)

20% dos respondentes já leram a RSL de Zhou et al. [5], e 40% a leram parcialmente. 40% não leram. Um total de 60% dos respondentes já tinham algum contato com a RSL, o que é um indicador positivo para a validação crítica dos achados, pois a amostra possui familiaridade com o estudo base.

4.3 VALIDAÇÃO DOS ACHADOS DA RSL ORIGINAL (RESULTADOS DO SURVEY)

Esta seção apresenta os resultados detalhados da aplicação do survey de validação em relação aos achados, limitações e oportunidades propostas na RSL de Zhou et al. [5]. Os dados são organizados e analisados, refletindo as percepções dos especialistas sobre os seguintes pontos, conforme as seções do questionário.

4.3.1 LLM PARA DETECÇÃO DE VULNERABILIDADES (Q7)

50% dos respondentes não possuíam opinião acerca da pergunta. 30% consideraram decoder-only (e.g., GPT-3.5/4, CodeGPT) como a mais promissora, enquanto 20% apontam encoder-decoder (e.g., CodeT5, T5). Nenhum respondente elegeu encoder-only (e.g., CodeBERT, BERT) como a mais promissora. Estes dados indicam uma preferência por arquiteturas decoder-only para detecção na percepção da comunidade, o que pode contrastar com o achado da RSL de Zhou et al. [5] que apontava predominância de encoder-only, sugerindo uma evolução na percepção da comunidade.

4.3.2 LLM PARA REPARO DE VULNERABILIDADES (Q8)

No que concerne ao reparo de vulnerabilidade, a quantidade de respondentes que não possuíam opinião acerca do assunto diminuiu para 30%. A preferência por arquiteturas decoder-only (e.g., GPT-3.5/4, CodeGPT) também se mantém forte nesse tópico, com 50%

dos respondentes considerando-as as mais promissoras. Encoder-only (e.g., CodeBERT, BERT) se manteve sem nenhum respondente escolhendo-a. encoder-decoder (e.g., CodeT5, T5) ficou com 10% das respostas . Um respondente mencionou "Claude models" na opção "Outro". Esta percepção alinha-se com o achado da RSL de Zhou et al. (2024) sobre a proeminência de decoder-only para reparo.

4.3.3 ADEQUAÇÃO DE LLMS PARA DETECÇÃO E REPARO DE VULNERABILIDADES (Q9)

A adequação de LLMs específicos para detecção e reparo de vulnerabilidades foi avaliada pelos respondentes com base em sua experiência. Para “CodeBERT (encoder-only)”, houve 6 respostas válidas, com 100% avaliando-o como 3 (Moderadamente Adequado). Para “CodeT5 (encoder-decoder)”, com 5 respostas válidas, 60% o consideraram 3 (Moderadamente Adequado), 20% 4 (Muito Adequado) e 20% 5 (Extremamente Adequado). Para GPT-3.5, com 7 respostas válidas, 42,9% o avaliaram como 3, 28,6% como 4, e 14,3% como 1 e 2. Para GPT-4, com 9 respostas válidas, 55,6% o consideraram 3, e 33,3% como 4. Para outras LLMs, com 6 respostas válidas, 66,7% o avaliaram como 4, e 16,7% como 1 e 3. As respostas em branco (sem experiência) foram mais frequentes para LLMs menos conhecidos, indicando menor familiaridade na comunidade. A especificação de "outras LLMs" inclui menções a "Gemini", "Llama", "Phi-2" e "Multi-task learning", sugerindo que esses modelos também são relevantes na prática.

Na Figura 2, podemos perceber que a LLM com maior taxa de adequação foi a CodeT5 (encoder-decoder), mas foi também a com maior taxa de não respostas. Ou seja, tem um baixo uso, porém uma alta percepção de adequação

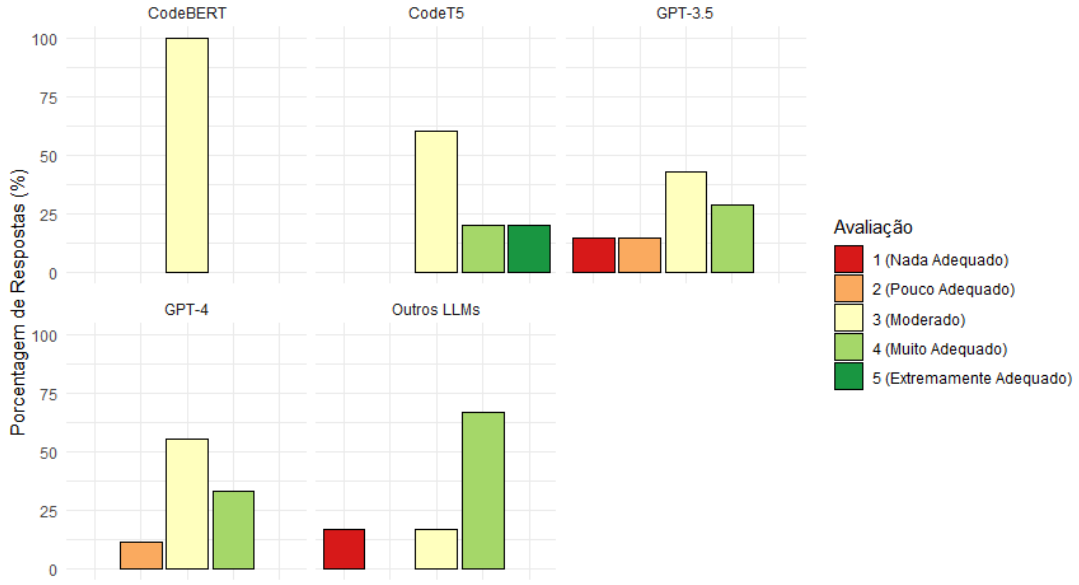


Figura 2. Adequação por LLM
Fonte: Dados do Survey

4.3.4 RANKING TÉCNICAS DE ADAPTAÇÃO PARA DETECÇÃO DE VULNERABILIDADES (Q10 E Q11)

O ranqueamento dos respondentes (onde 1 é o mais promissor e 3 o menos promissor) revelou que:

- **Fine-tuning:** Recebeu 4 votos para a posição 1, 2 votos para 2 e 3 votos para 3.
- **Prompt Engineering:** Recebeu 1 voto para a posição 1, 4 votos para 2 e 4 votos para 3.
- **Retrieval Augmentation:** Recebeu 0 votos para a posição 1, 4 votos para 2 e 5 votos para 3.

Isso sugere que o Fine-tuning é percebido como a técnica mais promissora para detecção, seguido por Prompt Engineering e Retrieval Augmentation. As explicações (Q11) indicaram que o Fine-tuning é valorizado pela "adaptação precisa do modelo ao domínio de segurança", enquanto o Prompt Engineering é visto como "rápido e computacionalmente barato", e o RAG é "essencial para capturar detalhes de código". P1 mencionou a importância da combinação de LLMs com outras estruturas de dados (e.g., CFGs).

4.3.5 INOVAÇÕES REFERENTES A FINE-TUNING PARA DETECÇÃO DE VULNERABILIDADES (Q12)

A análise da distribuição das respostas (escala de 1 a 5, onde 5 é Extremamente Promissor) revelou que as inovações "Data-centric Innovations", "Combination with DL Modules" e “Combination with program analysis” foram as mais promissoras, todas com pelo menos uma avaliação de nota 5. "Outras inovações" não apresentou grande destaque. "Causal Learning" e “Domain-specific Pre-Training” foram as mais mal avaliadas.

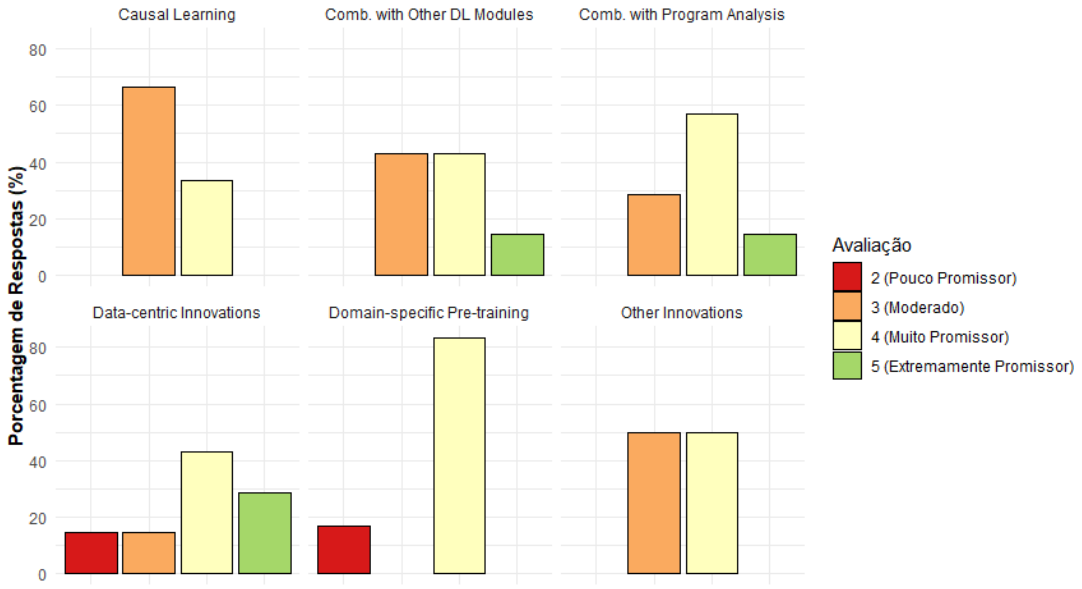


Figura 3. Promessa percebida por inovação (detecção)
Fonte: Dados do Survey

4.3.6 ESTRATÉGIAS REFERENTES A PROMPT ENGINEERING PARA DETECÇÃO DE VULNERABILIDADES (Q13)

A análise da distribuição das respostas de promessa percebida (escala de 1 a 5, onde 5 é Extremamente Promissor) revelou que a "Inclusion of Auxiliary Information" foi a estratégia mais promissora, com maior quantidade de respondentes elegendo como extremamente promissor, seguida por "Chain-of-thought (CoT) Prompting". "Few-shot Prompting" e "Zero-shot Prompting" foram as com menor destaque, sendo a segunda, a única que possui uma avaliação nada promissora.

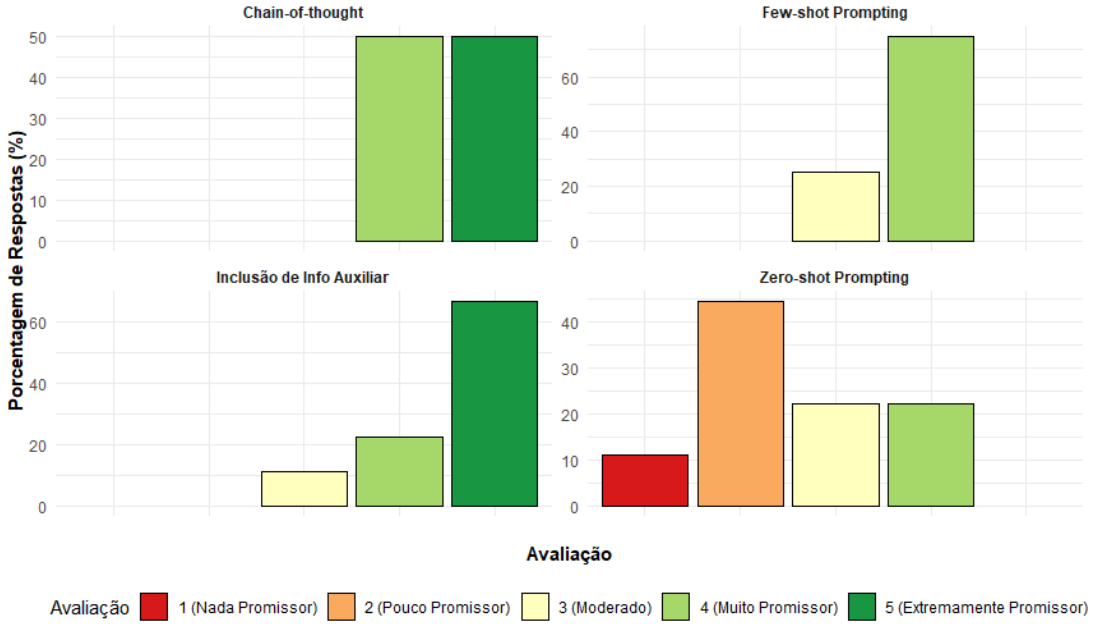


Figura 4. Promessa percebida por estratégia (detecção)
Fonte: Dados do Survey

4.3.7 TÉCNICAS DE ADAPTAÇÃO PARA REPARO DE VULNERABILIDADES (Q14)

A tendência evidenciada na Q11 se mantém também para o reparo de vulnerabilidades. Fine-tuning se mantém sendo a técnica mais promissora, com 70% dos votos. Em seguida temos prompt engineering, com 20% dos votos e por fim 10% dos respondentes não tinham opinião acerca do assunto

4.3.8 INOVAÇÕES REFERENTES A FINE-TUNING PARA REPARO DE VULNERABILIDADES (Q15)

A análise da distribuição das respostas de promessa percebida revelou que “Reinforcement Learning” foi a inovação mais promissora, com a grande maioria dos respondentes elegendo-a como extremamente promissora. A inovação "Data-centric Innovations" vem em seguida, também com boas avaliações. "Domain-specific Pre-training” fica em terceiro lugar, apresentando avaliações mistas. Por fim, tivemos “Model-centric Innovations” sendo a única com avaliação ruim (nada promissor).

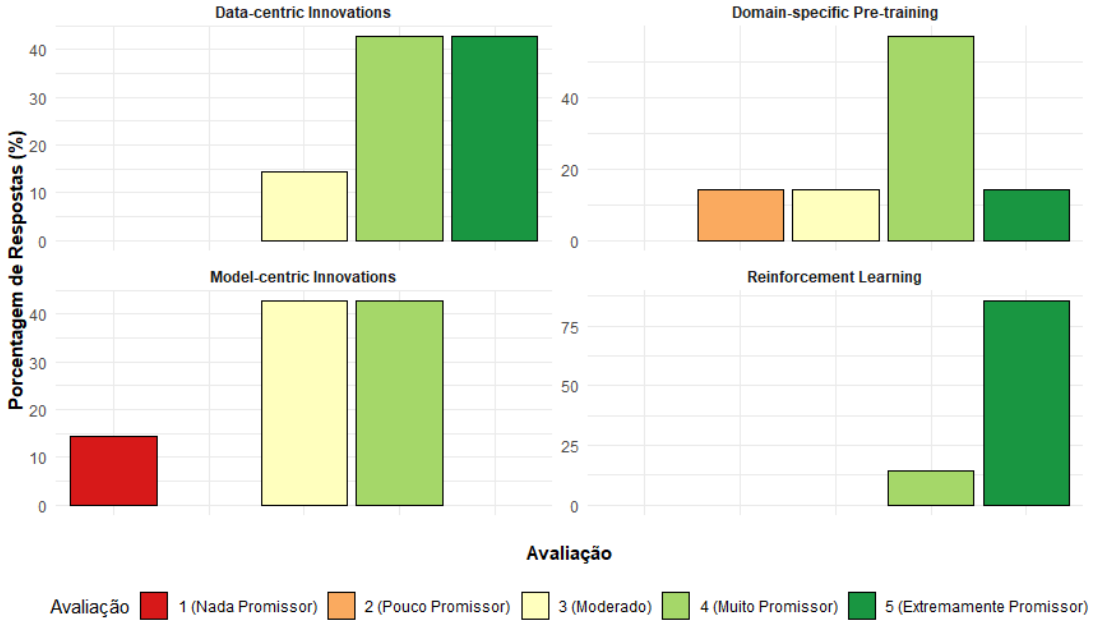


Figura 5. Promessa percebida por inovação (reparo)
Fonte: Dados do Survey

4.3.9 ESTRATÉGIAS REFERENTES A PROMPT ENGINEERING PARA REPARO DE VULNERABILIDADES (Q16)

A análise da distribuição das respostas de promessa percebida revelou que a "Inclusion of Auxiliary Information" foi a estratégia mais promissora, com a maior parte dos respondentes elegendo como extremamente promissora. "Few-shot Prompting" uma percepção mista, ficando em segundo lugar. Por fim, "Zero-shot Prompting" obteve a pior avaliação, sendo a única com voto nada promissor. A tendência observada na Q13 se mantém aqui, ou seja, quão maior o nível de informação e contexto dado ao prompt, melhor a LLM tende a performar.

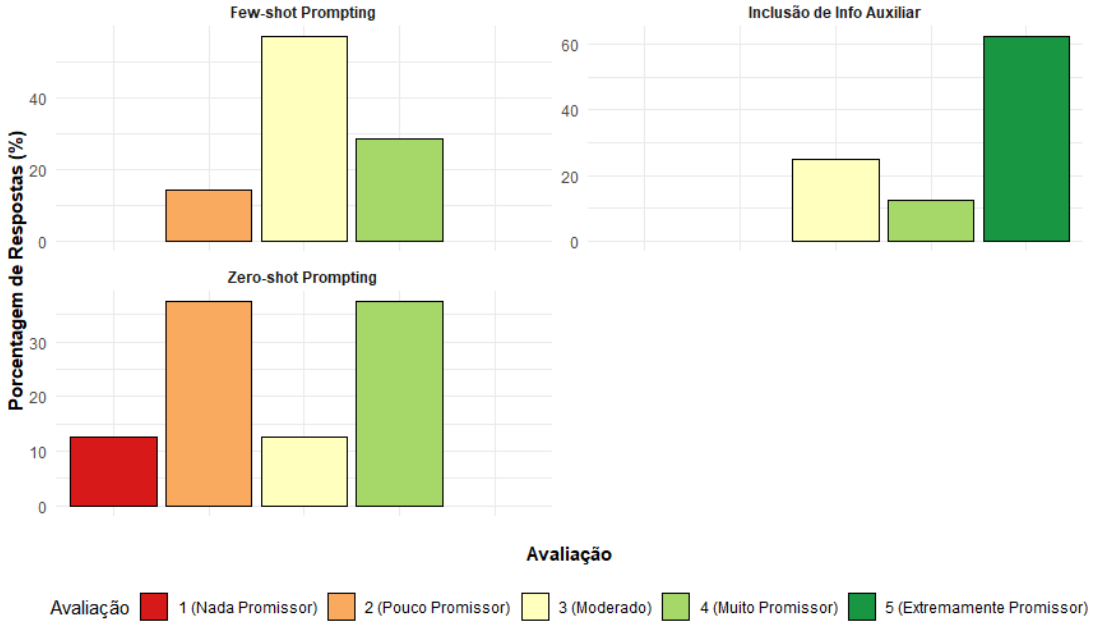


Figura 6. Promessa percebida por estratégia
(reparo)
Fonte: Dados do Survey

4.3.10 RELEVÂNCIA DE INVESTIGAR DETECÇÃO E REPARO DE VULNERABILIDADE EM UM NÍVEL DE GRANULARIDADE MAIOR QUE FUNÇÃO/LINHA (Q17 E Q18)

Foi avaliada com média de 4.9 de 5, com 90% dos respondentes considerando "Extremamente Importante" e 10% "Muito Importante". Os respondentes P4, P6, P7 e P10 reforçaram que muitas vulnerabilidades complexas exigem contexto além da linha/função, e que a falta de granularidade maior pode levar à perda de detecções em alvos de alto valor. Segundo P6 “isso ocorre porque realizar a detecção e depois o reparo no nível de função ou de linha pode ser útil, porém há muitas vulnerabilidades que não se manifestam localmente”.

4.3.11 CRITICIDADE PERCEBIDA POR LIMITAÇÃO DE QUALIDADE DE DADOS (Q19)

A limitação mais crítica foi "Lack of test cases to verify the correctness of vulnerability fixes and prevent new bugs in repair datasets", com a maior parte dos respondentes considerando-a extremamente crítica. "Dependence on labels generated by automated rules or tools (heuristic labels), often leading to noise or inaccuracies in detection datasets" apresentou também uma alta criticidade, com maior parte dos respondentes elegendo como

muito crítica. Por fim, "Concern about "data contamination" (evaluation datasets present in the pre-training corpus of LLMs)" obteve uma avaliação menos crítica, sendo a única votada como pouco crítica.

Figura 7. Criticidade percebida por limitação de qualidade de dados
Fonte: Dados do Survey

4.3.12 ASPECTOS IMPORTANTES EM SOLUÇÕES BASEADAS EM LLM (Q20)

O aspecto de “High accuracy and robustness against perturbations/attacks” foi o mais valorizado, com a maior parte dos respondentes elegendo como extremamente importante. Em segundo lugar, “Ability to interact and collaborate with developers (e.g., feedback, explanations)” apresentando também uma alta percepção de importância. Já “Continuous and seamless integration into developer workflows and tools (e.g., IDEs)” uma percepção mais diversa e foi a considerada menos importante, porém, ainda assim, apresentou alto grau de importância. Todos os aspectos avaliados foram considerados importantes. 80% dos respondentes avaliaram todos os aspectos e as notas variaram em sua maioria entre 4 e 5.

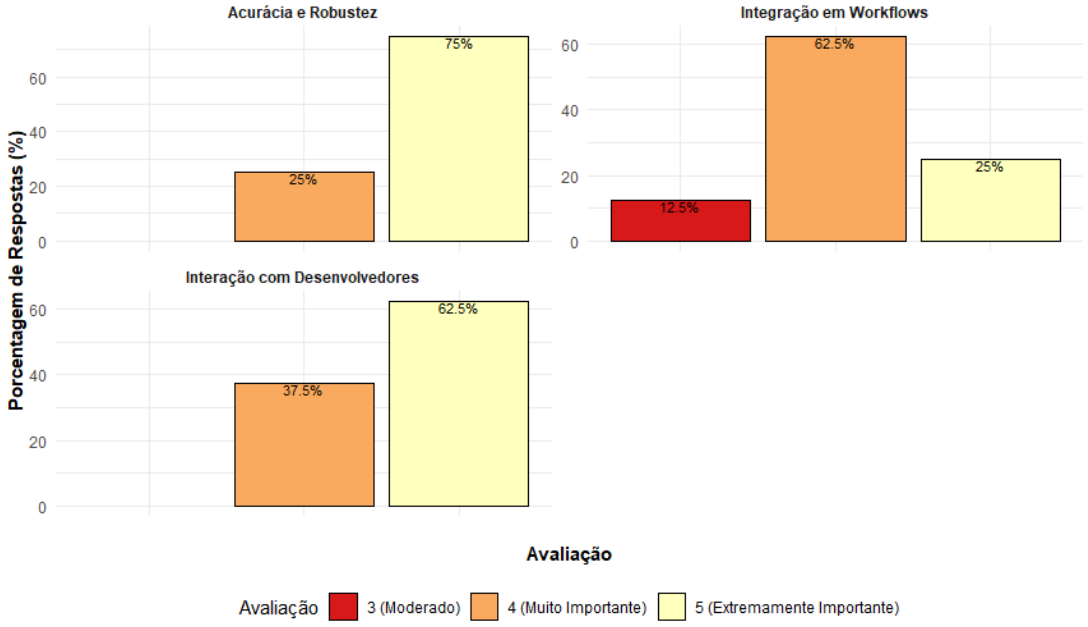


Figura 8. Importância percebida por aspecto
Fonte: Dados do Survey

4.3.13 NÍVEL DE MATURIDADE NA APLICAÇÃO DE LLMS PARA DETECÇÃO DE VULNERABILIDADE EM SOFTWARE (Q21)

Em relação ao nível de maturidade de LLMs para detecção, 60% consideram "Moderate (some products/tools, but with significant challenges)", 20% "Incipient", 10% não tiveram opinião sobre o assunto e apenas e 10% categorizam como "Advanced".

4.3.14 NÍVEL DE MATURIDADE NA APLICAÇÃO DE LLMS PARA REPARO DE VULNERABILIDADE EM SOFTWARE (Q22)

Para reparo, 50% indicam "Moderate", 40% "Incipient", 10% não tiveram opinião sobre o assunto. Isso sugere que a área de reparo apresenta resultados inferiores quando comparado a área de detecção e ainda está em fase de desenvolvimento, com desafios práticos significativo

4.3.15 CRITICIDADE DAS LIMITAÇÕES (Q23 E Q24)

A limitação mais crítica foi a "Suboptimal performance due to vulnerability data complexity" (L3 - "Complexidade dos Dados") com 40%, seguida por "Lack of high-quality

vulnerability datasets" (L2 - “Falta de Datasets de Qualidade”) com 30% e "Lack of high accuracy and robustness" (L6 - “Falta de Acurácia e Robustez”) com 20%. As explicações de P1 indicam que a complexidade das vulnerabilidades, especialmente aquelas que envolvem múltiplas camadas, como interações on-chain e off-chain em contratos inteligentes, é um desafio central. P4, P6, P7 e P10 destacam a dificuldade em avaliar corretamente os resultados gerados pelos LLMs, a instabilidade e alucinações nos outputs, além da baixa qualidade dos dados utilizados atualmente, muitas vezes obsoletos e com baixo volume de código. Essas respostas sugerem uma demanda por benchmarks mais robustos e realistas, tanto para treinamento quanto para validação.

4.3.16 RANKING DAS OPORTUNIDADES DE PESQUISA E DIREÇÕES FUTURAS (Q25 E Q26)

O ranqueamento dos respondentes (onde 1 é o mais promissor e 3 o menos promissor) revelou que:

- **Curation of high-quality benchmark datasets for vulnerability detection (O1 - “Curadoria de Datasets de Alta Qualidade”):** Recebeu 2 votos para a posição 1, 2 voto para 2, e 4 votos para 3.
- **Repository-level (higher granularity) vulnerability detection/repair (O2 - “Análise em Nível de Repositório”):** Recebeu 4 votos para a posição 1, 2 voto para 2, e 2 votos para 3.
- **Development of vulnerability-specialized LLMs (O3 - “Desenvolvimento de LLMs Especializados”):** Recebeu 2 votos para a posição 1, 4 voto para 2, e 2 voto para 3.
- **Use of more advanced LLM techniques (O4 - “Uso de Técnicas Avançadas de LLM”):** Recebeu 3 votos para a posição 1, 2 voto para 2, e 1 voto para 3.
- **Development of deployment-ready features (O5 - “Desenvolvimento de Features Prontas para Deploy”):** Recebeu 3 voto para a posição 1, nenhum voto para 2, e 3 votos para 3.

Para facilitar a leitura dos dados e traçar uma tendência, foram atribuídos pontos de acordo com o voto de cada item. Os votos “1” somam 3 pontos, os votos “2” 2 pontos e os votos “3” 1 ponto. Dessa forma, foi criada a FIGURA 9, que nos permite observar quais oportunidades foram consideradas mais importantes. A distância entre as pontuações é

pequena e nos permite concluir que a “Análise em Nível de Repositório” (O2) se mostra mais relevante, com 18 pontos e “Desenvolvimento de Features Prontas para Deploy” (O5) menos relevantes, com 12 pontos.

As justificativas fornecidas (Q26) reforçaram a importância da qualidade dos dados e da contextualização no tratamento de vulnerabilidades mais complexas. O respondente P1 destacou que alguns tópicos, como “Desenvolvimento de LLMs Especializados” (O3), já avançaram significativamente, com marcos estabelecidos, e por isso foram considerados menos prioritários. Outro ponto enfatizado por P7 foi que, sem dados de qualidade, os modelos atuais não conseguirão produzir resultados confiáveis, ficando suscetíveis a falhas em relação aos princípios fundamentais da segurança. Além disso, foi reiterada a necessidade de integração com ferramentas que permitam melhor avaliação e fornecimento de ground truth, especialmente em contextos que exigem maior profundidade na análise das vulnerabilidades.

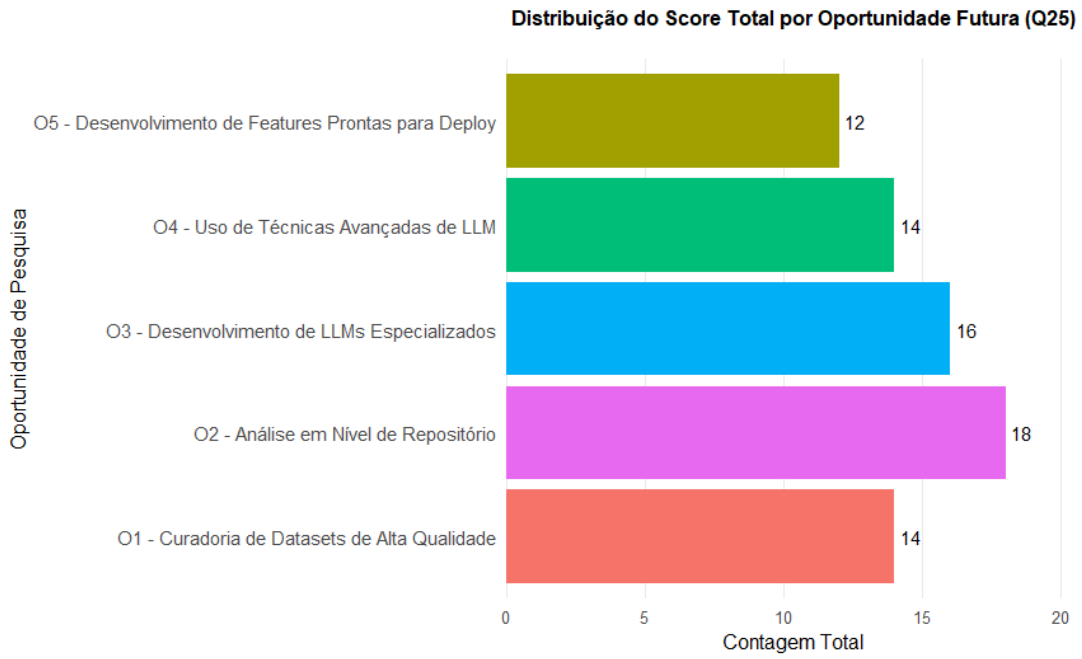


Figura 9. Distribuição das pontuações de oportunidades e limitações
Fonte: Dados do Survey

4.3.17 COMENTÁRIOS ADICIONAIS (Q27)

A última pergunta do survey consistia em um campo aberto onde os respondentes poderiam colocar comentários adicionais.

Dentre as respostas recebidas, o participante P4 recomendou acompanhar a iniciativa

da DARPA relacionada ao uso de IA para segurança cibernética, destacando que, embora ainda não haja publicações formais, já existem comentários relevantes em entrevistas e palestras de conferências da área. Outro respondente (P6) apontou a importância de incorporar modelos multimodais para avançar na detecção e reparo de vulnerabilidades, sugerindo que a combinação de diferentes tipos de entrada (como código, texto e possivelmente imagens) pode ser um caminho promissor ainda pouco explorado.

4.4 RESPOSTA ÀS QUESTÕES DE PESQUISA

A construção do survey foi estruturada a partir das QPs mencionadas no tópico 1.3. A QP1 foi respondida com as questões da sessão 2 do survey (Q7 à Q9). A QP2 foi respondida nas sessões 3 e 4 (Q10 à Q16). A QP3 foi respondida nas sessões 6 e 7 (Q17 à Q27). Por fim, a QP4 foi respondida na sessão 6(Q21 à Q27).

4.4.1 QP1: ARQUITETURAS E MODELOS DE LLMS PARA DETECÇÃO E REPARO

As conclusões do Survey apontam para uma preferência dos especialistas por arquiteturas de LLMs com fortes capacidades generativas. Para a tarefa de detecção de vulnerabilidades, os especialistas consideram as arquiteturas decoder-only (e.g., GPT-3.5/4) como as mais promissoras (30% dos votos), seguidas pelas encoder-decoder (20%). A arquitetura encoder-only não recebeu nenhum voto. Para o reparo de vulnerabilidades, essa tendência se intensifica, com 50% dos especialistas indicando as arquiteturas decoder-only como as mais promissoras, com menções a outros modelos generativos como "Claude". Em relação a modelos específicos, GPT-4 foi um dos mais bem avaliados em adequação, enquanto CodeBERT foi considerado apenas "Moderadamente Adequado". Essa percepção atual dos especialistas, coletada no survey, contradiz parcialmente os achados da RSL de Zhou et al. [5] para detecção, mas confirma e reforça suas conclusões para reparo. A RSL concluiu que, nos estudos publicados até então, os modelos encoder-only (como CodeBERT) dominaram a área de detecção de vulnerabilidades , sendo o CodeBERT o modelo mais utilizado. O fato de que os especialistas do survey não consideram essa arquitetura como promissora sugere uma rápida evolução na percepção da comunidade, que agora prioriza modelos mais recentes e com maior capacidade generativa, mesmo para tarefas de classificação. Por outro lado, para o reparo, os resultados do survey alinham-se perfeitamente com a RSL, que já havia identificado que os modelos comerciais e decoder-only eram

proeminentes nesta tarefa. A forte preferência dos especialistas por essa arquitetura no survey valida e fortalece essa conclusão da literatura.

4.4.2 QP2: TÉCNICAS DE ADAPTAÇÃO DE LLMS

As conclusões do survey indicam que o fine-tuning é percebido como a técnica de adaptação mais promissora tanto para detecção quanto para reparo de vulnerabilidades, recebendo a grande maioria dos votos para a primeira posição. Prompt engineering foi consistentemente classificada como a segunda mais relevante. Para as inovações específicas de fine-tuning, o Aprendizado por Reforço (Reinforcement Learning) foi considerado o mais promissor para reparo, com uma média de avaliação de 4.86 de 5. No âmbito de prompt engineering, a estratégia de "Inclusion of Auxiliary Information" foi a mais bem avaliada para ambas as tarefas, reforçando a percepção de que o desempenho do LLM melhora significativamente com o aumento do contexto e da informação fornecida. A percepção de eficácia dos especialistas valida a proeminência das técnicas identificadas por Zhou et al. [5]. A RSL aponta que o fine-tuning é a abordagem de adaptação mais comum na literatura, sendo utilizada em 73% dos estudos de detecção e 63% dos de reparo. O ranqueamento do survey confirma essa popularidade e tendência. Além disso, o survey valida a importância de sub-técnicas específicas catalogadas na RSL. A alta avaliação do "Reinforcement Learning" e da "Inclusion of Auxiliary Information" confirma que as estratégias detalhadas na taxonomia da RSL são consideradas importantes pela comunidade para o avanço da área.

4.4.3 QP3: RELEVÂNCIA E IMPACTO DAS LIMITAÇÕES ATUAIS

Os resultados do survey demonstram que os especialistas consideram as limitações atuais da área como extremamente relevantes e impactantes. A necessidade de analisar vulnerabilidades em uma granularidade maior que função/linha foi o ponto de maior consenso, avaliado como "Extremamente Importante" por 90% dos respondentes (média 4.9 de 5). Em relação à qualidade dos dados, a falta de casos de teste para validar os reparos foi apontada como a limitação mais crítica (média 4.71 de 5). Finalmente, ao avaliar os aspectos mais importantes de uma solução, a alta acurácia e robustez (média 4.75) e a interação com o desenvolvedor (média 4.62) foram as mais valorizadas. Esses achados confirmam e reforçam a criticidade das limitações apontadas na seção de "Limitações" da RSL de Zhou et al. [5]. A "Limitação 1: Pequena Granularidade de Entrada" , que segundo a RSL pode levar à omissão de vulnerabilidades complexas, foi ecoada e validada pela quase totalidade dos especialistas.

Da mesma forma, a "Limitação 2: Falta de Dataset de Vulnerabilidades de Alta Qualidade" , especialmente a falta de testes para avaliar os reparos, foi validada como a preocupação mais crítica sobre dados. As prioridades dos especialistas também se alinharam diretamente com a "Limitação 6: Falta de Alta Acurácia e Robustez" e a "Limitação 5: Falta de Consideração de Implantação", que aborda a interação com o desenvolvedor e a integração ao workflow. O survey, portanto, valida que os desafios identificados na literatura são barreiras ainda percebidas na prática.

4.4.4 QP4: DIREÇÕES FUTURAS E OPORTUNIDADES DE PESQUISA

A percepção dos especialistas é de que a área ainda está em um nível de maturidade "Moderado" para detecção e "Incipiente" para reparo, indicando um vasto campo para novas pesquisas. Ao ranquear as oportunidades futuras, a "Repository-level (higher granularity) vulnerability detection/repair" foi considerada a mais relevante. No entanto, a pontuação entre as diferentes oportunidades foi muito próxima, sugerindo que todas as direções de pesquisa listadas são vistas como importantes. As justificativas qualitativas reforçaram que a qualidade dos dados e a criação de benchmarks robustos são pré-requisitos fundamentais para o avanço em qualquer uma das outras frentes. Esses resultados oferecem uma validação prática para o roadmap de pesquisa proposto por Zhou et al.. A oportunidade mais bem ranqueada no survey ("repository-level (higher granularity) vulnerability detection/repair") corresponde diretamente à "Oportunidade 2" da RSL, identificada como um caminho promissor, porém sub-explorado. O fato de os especialistas considerarem todas as opções relevantes, com pontuações próximas, também se alinha à visão da RSL de que existem múltiplos "módulos subexplorados" que precisam ser investigados. A ênfase dos especialistas na necessidade de dados de qualidade superior também valida fortemente a "Oportunidade 1: Curadoria de Conjuntos de Teste de Alta Qualidade" como um pilar para pesquisas futuras. Em suma, as prioridades da comunidade, capturadas pelo survey, alinham-se diretamente com as direções futuras traçadas pela RSL.

4.5 AMEAÇAS À VALIDADE E LIMITAÇÕES DO SURVEY

Seguindo as boas práticas de avaliação empírica propostas por Molléri et al. [7], é fundamental refletir sobre as ameaças à validade que podem ter impactado este estudo. Esta seção discute as limitações observadas e sua relação com os principais tipos de validade sugeridos no checklist.

4.5.1 VALIDADE DE CONSTRUÇÃO

Refere-se à correspondência entre o que o questionário mede e os conceitos que se propõe a investigar. A principal ameaça seria a má interpretação das perguntas pelos especialistas. Para mitigar este risco, as questões foram diretamente extraídas e adaptadas dos achados da RSL de Zhou et al. [5], garantindo o alinhamento com a literatura. Adicionalmente, foi realizado um teste piloto com um pesquisador da área, cujo feedback levou à reformulação de questões para clarificar termos técnicos e remover possíveis ambiguidades. Ainda assim, reconhece-se que interpretações subjetivas das questões por parte dos respondentes podem ter ocorrido.

4.5.2 VALIDADE EXTERNA

A validade externa, que trata da capacidade de generalizar os resultados, representa a limitação mais significativa deste survey. A amostra obtida foi pequena, com apenas 10 respostas válidas. Além disso, a estratégia de amostragem, focada em autores da RSL de referência e de artigos que a citaram, pode ter introduzido um viés de seleção, favorecendo a participação de indivíduos com visões potencialmente já alinhadas às da RSL. Esses fatores impedem a generalização estatística dos achados para toda a comunidade.

4.5.3 VALIDADE DA CONCLUSÃO

Refere-se à robustez das conclusões tiradas a partir da análise dos dados. O baixo número de participantes é uma ameaça a esta validade, pois não permite a aplicação de testes estatísticos robustos. Portanto, os resultados quantitativos devem ser interpretados como indicativos preliminares e ilustrativos da percepção do grupo, e não como conclusões generalizáveis.

4.6 SÍNTESE DO CAPÍTULO

Este capítulo apresentou os resultados e análise do survey conduzido com 10 especialistas. Foi feita a categorização demográfica dos participantes, com intuito de demonstrar o alinhamento da amostra com os interesses da pesquisa. As respostas foram analisadas a fim de responder às questões de pesquisa, o que revelou o contraste entre a percepção atual dos especialistas e os achados da RSL original, com destaque para a

preferência por arquiteturas decoder-only para a detecção de vulnerabilidades. Adicionalmente, a análise apontou o fine-tuning como a técnica de adaptação mais promissora e reforçou a criticidade de limitações como a complexidade e a baixa qualidade dos dados. Por fim, as ameaças à validade do survey foram apresentadas.

5. ANÁLISE EXPLORATÓRIA DA LITERATURA

Este capítulo tem como objetivo complementar e enriquecer a análise dos resultados do survey apresentado no Capítulo 4. Dada a natureza exploratória do survey e o número limitado de respondentes, uma análise não sistemática da literatura recente foi conduzida para fornecer um panorama mais amplo das tendências emergentes na área de LLMs para detecção e reparo de vulnerabilidades.

Conforme a metodologia descrita na seção 3.4, esta análise não se configura como uma revisão sistemática, mas sim como uma leitura exploratória e qualitativa de trabalhos publicados a partir de abril de 2024 que citaram a RSL de Zhou et al. [5]. O foco é identificar como a pesquisa recente aborda as mesmas questões de pesquisa (QPs) utilizadas neste trabalho, permitindo uma triangulação dos dados obtidos na RSL original, no survey com especialistas e na literatura emergente.

5.1 RESPOSTA ÀS QUESTÕES DE PESQUISA

5.1.1 QP1: ARQUITETURAS E MODELOS DE LLMS PARA DETECÇÃO E REPARO

A análise dos estudos recentes, confirma a tendência observada no Survey, que concerne a dominância de modelos decoder-only tanto para detecção quanto para reparo. Dos 23 estudos primários analisados, 20 abordaram o tema de detecção:

- 55% dos estudos utilizaram exclusivamente modelos decoder-only, como as séries GPT e Llama. (E13, E19, E21, E28, E29, E32, E46, E48, E49, E58 e E74).
- 20% dos estudos mencionaram o uso exclusivo de modelos encoder-only e quase sempre como uma linha de base para comparação, validando a percepção dos especialistas de que o foco da comunidade migrou para modelos generativos. (E04, E22, E23 e E44).
- 25% dos artigos mencionaram o uso dos dois tipos de LLM (encoder-only e decoder-only)(E37, E43, E47, E75 e E76).

A maioria dos estudos primários focava exclusivamente na tarefa de detecção, com apenas 9 abordando o tema de reparo:

- 89% dos artigos utilizaram exclusivamente modelos decoder-only, como GPT-4, Deepseek, Llama 3, Gemini e Claude. (E15, E22, E23, E28, E30, E46, E59 e E74).
- 11% utilizaram ambas as arquiteturas. (E76).

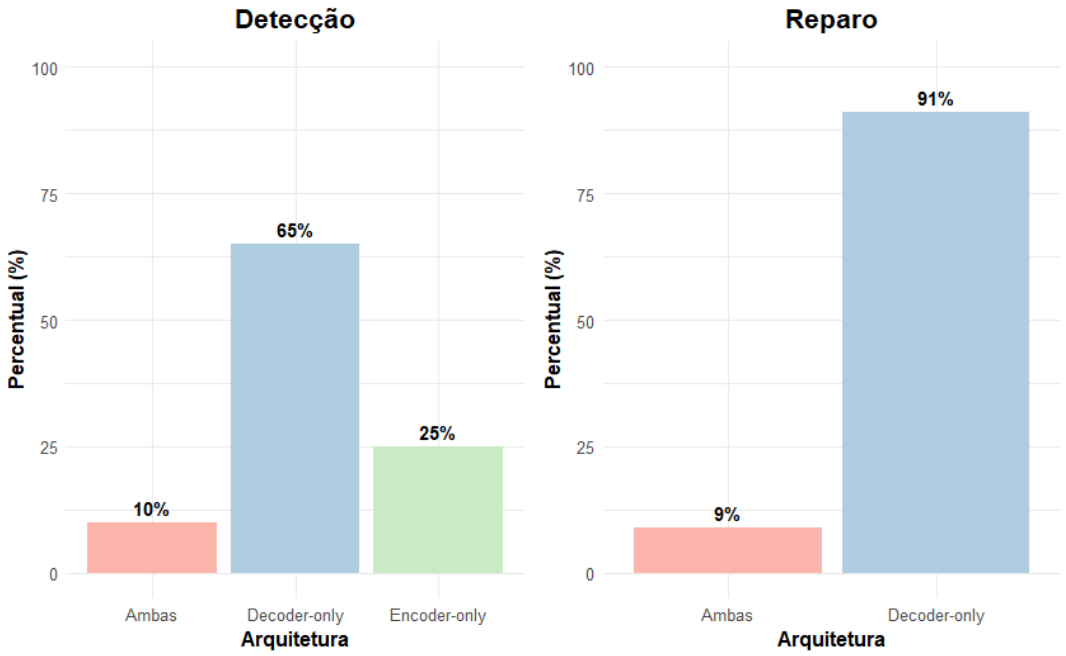


Figura 10. Distribuição de arquitetura LLM por tarefa
Fonte: Dados da análise exploratória

5.1.2 QP2: TÉCNICAS DE ADAPTAÇÃO DE LLMS

A análise das técnicas de adaptação revela uma clara preferência por métodos que não exigem o re-treinamento completo dos modelos. É importante notar que muitos artigos utilizaram múltiplas técnicas simultaneamente.

Para a tarefa de detecção (20 artigos):

- Prompt-engineering foi a técnica mais utilizada, estando presente em 65% dos artigos, com abordagens como "Chain-of-Thought (CoT)" e "In-Context Learning (ICL)" sendo bem comuns. (E13, E19, E21, E28, E32, E37, E43, E46, E48, E49, E58, E74 e E76).
- 35% dos estudos mencionaram o uso de fine-tuning com métodos como "Parameter-Efficient Fine-Tuning (PEFT)," e "LoRA". (E04, E23, E37, E43, E44, E48 e E74).
- Por fim, RAG foi mencionado em 25% dos estudos. (E13, E21, E32, E43 e

E48).

Para a tarefa de reparo (9 artigos), a tendência se mantém:

- Prompt engineering foi utilizada em 78% dos estudos que abordaram reparo. (E15, E28, E30, E46, E59, E74 e E76).
- Fine-tuning foi utilizado em 45% dos estudos. (E15, E23, E59 e E74).

Prompt engineering se mostrou a técnica mais popular por sua simplicidade, mas de acordo com a maioria dos autores que utilizaram fine-tuning (E37, E48, E59, E74 e E76), ela é a técnica que traz melhores resultados e precisão, porém é mais custosa, visto que necessita de dados rotulados e confiáveis para o treinamento. Outras técnicas foram também mencionadas com menor frequência, como o uso de Aprendizado por Reforço (RL) (E43 e E47), especialmente em conjunto com fine-tuning, e a combinação de LLMs com ferramentas de análise estática, atuando como um oráculo para fornecer feedback iterativo (E15, E23, E28 e E59).

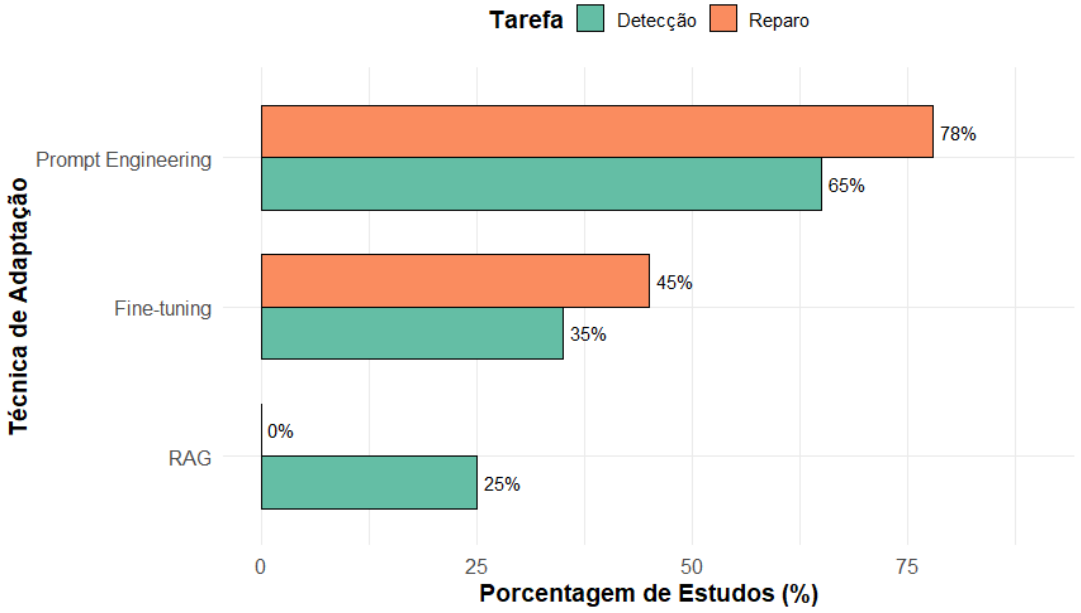


Figura 11. Frequência de técnica de adaptação por estudo
Fonte: Dados da análise exploratória

5.1.3 QP3: RELEVÂNCIA E IMPACTO DAS LIMITAÇÕES ATUAIS

A análise exploratória da literatura recente revela que as limitações identificadas por Zhou et al. [5] continuam sendo um foco central de pesquisa. A frequência com que cada limitação foi abordada nos 23 estudos primários é a seguinte:

- L6 (Falta de Acurácia e Robustez) foi a mais abordada, estando presente em 78% dos estudos primários. Isso demonstra que a melhoria do desempenho dos LLMs ainda é a principal preocupação da comunidade (E15, E19, E21, E22, E23, E28, E30, E37, E43, E44, E46, E47, E48, E49, E58, E59, E74 e E75).
- L2 (Falta de Datasets de Qualidade) recebeu atenção em 61% dos estudos, onde os autores reconhecem a qualidade dos dados como um gargalo. (E04, E15, E21, E22, E23, E30, E32, E37, E43, E44, E47, E74, E75 e E76).
- L1 (Pequena Granularidade de Entrada) foi abordada em 39% dos estudos (E13, E21, E22, E23, E32, E37, E46, E59 e E74).
- L3 (Complexidade dos Dados), referente a vulnerabilidades interprocedurais e complexas, foi abordada em 35% dos estudos (E13, E21, E22, E23, E32, E43, E46, E47, E74, E75 e E76).
- L5 (Falta de Integração com o Desenvolvedor) foi discutida em 30% dos estudos, e sua baixa exploração contrasta com a alta importância atribuída pelos especialistas no survey (E13, E15, E23, E30, E58, E59 e E74).
- L4 (Dependência de LLMs Leves) foi a menos explorada, com 26% de menções, o que é consistente com a tendência de uso de modelos maiores (E23, E32, E43, E48, E59 e E76).

Esses dados quantitativos corroboram fortemente os resultados do survey. A alta criticidade atribuída pelos especialistas à falta de acurácia (L6), à qualidade dos dados (L2) e à necessidade de maior granularidade (L1) reflete-se diretamente no volume de pesquisas dedicadas a esses temas.

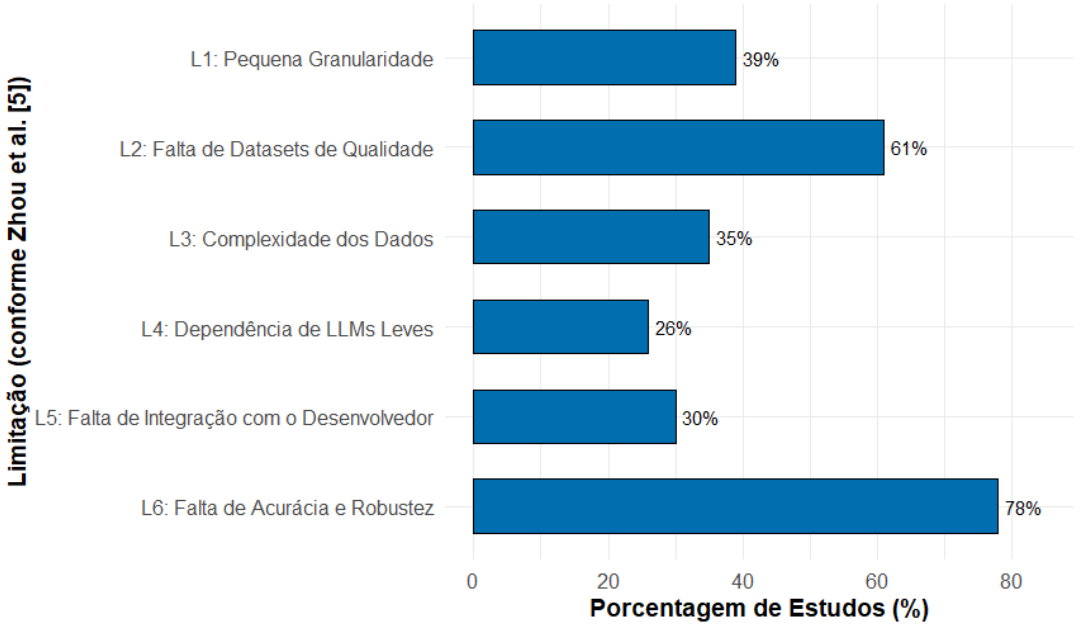


Figura 12. Frequência das limitações abordadas por estudo
Fonte: Dados da análise exploratória

5.1.4 QP4: DIREÇÕES FUTURAS E OPORTUNIDADES DE PESQUISA

A exploração das oportunidades de pesquisa propostas por Zhou et al. [5] nos 23 estudos revela os seguintes focos da comunidade:

- A oportunidade mais explorada foi a O1 (Curadoria de Datasets de Alta Qualidade), abordada em 69% dos estudos (E04, E13, E21, E22, E23, E29, E30, E32, E37, E43, E44, E46, E47, E49, E74 e E75). Isso demonstra um esforço massivo da comunidade para construir benchmarks mais robustos e dados de treinamento mais confiáveis, reconhecendo que a qualidade dos dados é um pré-requisito para o avanço da área. Dentre os principais frameworks e datasets presentes nos estudo, temos:
    - LLM4VFD (E13) utilizou o novo dataset BigVulFixes, com 1.689 commits de correção de vulnerabilidade posteriores a 2023 para evitar vazamento de dados.
    - JITVUL (E21) foi introduzido como um benchmark para detecção just-in-time (JIT), composto por 1.758 commits pareados (vulnerável vs. corrigido).
    - Zero-Day do framework APPATCH (E30), coletado especificamente com vulnerabilidades não presentes nos dados de treinamento dos

LLMs.

    - LLMVulExp (E37) utilizou e processou datasets existentes como SeVC e DiverseVul, aplicando técnicas de deduplicação e downsampling para garantir o balanceamento.

- Em segundo lugar, a O4 (Uso de Técnicas Avançadas de LLM) foi abordada em 57% dos estudos (E13, E15, E19, E21, E28, E28, E29, E30, E48, E49, E59, E74 e E76). A pesquisa recente demonstra um forte interesse em ir além de abordagens simples, focando em prompt engineering avançada, como as estratégias dinâmicas e adaptativas do APPATCH (E30) e o uso de Chain-of-Thought (CoT) no LLM4VFD (E13). Além disso, há um esforço para aprimorar as capacidades de raciocínio dos modelos. Destaca-se também a implementação de processos iterativos, onde o LLM refina suas soluções com base em feedback (E15, E23, E28 e E59), tem se mostrado eficaz para melhorar significativamente as taxas de sucesso no reparo de vulnerabilidades.
- O3 (Desenvolvimento de LLMs Especializados) apareceu em 30% dos estudos (E04, E37, E43, E44, E47, E74 e E75), geralmente através de fine-tuning.
- O5 (Desenvolvimento de Features Prontas para Deploy) foi explorada em 21% dos estudos (E13, E43, E58, E59 e E74), com discussões sobre integração ao workflow do desenvolvedor.
- O2 (Análise em Nível de Repositório) foi a oportunidade menos explorada, com apenas 17% de menções (E21, E22, E30 e E43).

A baixa exploração da O2 (granularidade maior) e da O5 (deploy) alinha-se com as limitações menos abordadas (L1 e L5), reforçando a ideia de que a pesquisa ainda está mais focada em desafios fundamentais de desempenho (L6) e dados (L2) do que em questões de escopo e aplicação prática.

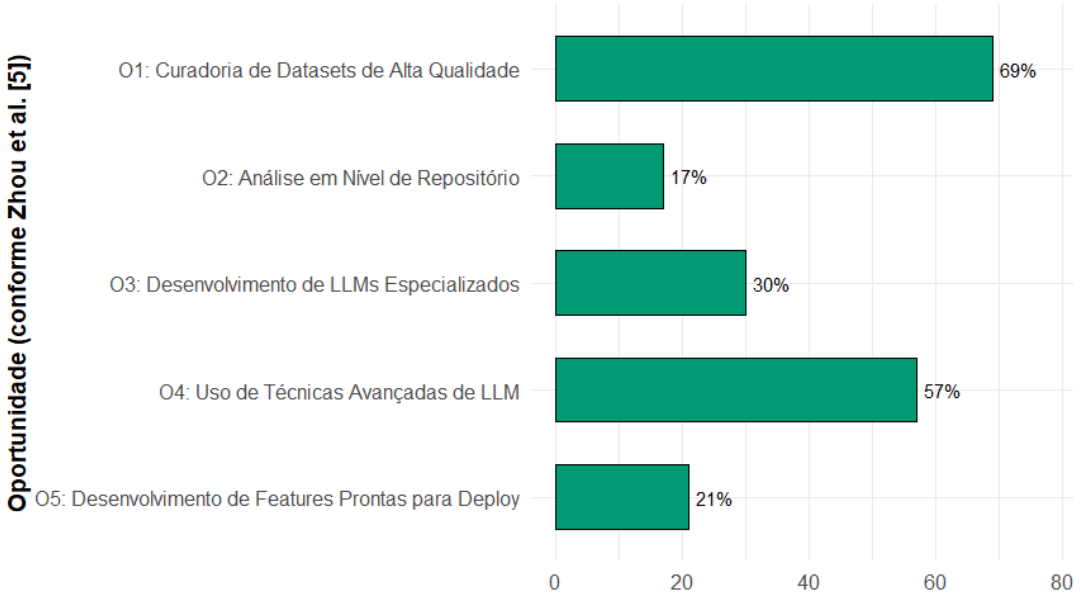


Figura 13. Frequência das oportunidades abordadas por estudo
Fonte: Dados da análise exploratória

5.2 AMEAÇAS À VALIDADE E LIMITAÇÕES DA ANÁLISE EXPLORATÓRIA

Para avaliar as ameaças à validade desta análise exploratória, foi utilizada a categorização apresentada por Wohlin et al. [13], que inclui quatro tipos de ameaças à validade: construção, interna, externa e de conclusão.

5.2.1 VALIDADE DE CONSTRUÇÃO

Está relacionada à correspondência entre o que o estudo mede e os conceitos que ele se propõe a investigar.

A extração de dados foi realizada com o auxílio do Google NotebookLM, uma ferramenta baseada em LLM. A escolha por essa abordagem se justifica pela necessidade de processar um grande volume de artigos de forma eficiente dentro das restrições de tempo. Reconhece-se que o uso de LLMs para extração de dados em estudos primarios é uma área emergente e que, conforme apontado por Felizardo et al. [12], apresenta desafios significativos de reprodutibilidade e fidelidade. Questões como a sensibilidade ao prompt, a aleatoriedade inerente dos modelos e a falta de transparência no processo de raciocínio do LLM constituem ameaças à validade dos resultados. Para mitigar essa ameaça e garantir a confiabilidade dos dados, foi implementado um rigoroso processo de validação humana detalhado no APÊNDICE D. Ainda assim, embora tenha havido checagem manual direcionada, a interpretação da LLM pode não capturar todas as nuances dos artigos, podendo

levar a classificações, resumos imprecisos e principalmente omissão de detalhes relevantes. A natureza não-sistemática e a ausência de uma revisão por pares no processo de extração de dados aumentam o risco de viés do pesquisador

Além disso, a análise considerou apenas artigos que citaram a RSL de Zhou et al. [5], o que pode introduzir viés de seleção. Trabalhos relevantes que não citaram este estudo específico não foram incluídos. Além disso, o processo de triagem inicial, baseado na leitura de títulos e resumos para constatar se havia uso de LLMs para detecção ou reparo de vulnerabilidade, pode ter levado à exclusão de artigos pertinentes cujo conteúdo não estava claramente refletido nessas seções.

Não foi adotado como critério de exclusão a não publicação em veículos revisados por pares, o que permitiu a inclusão de pre-prints do arXiv. Como o Google Scholar pode priorizar a versão de acesso público, foi realizada uma verificação manual posterior de cada um dos 23 estudos. Constatou-se que apenas uma parte (9 dos 23) possuía uma versão oficial publicada, e os 14 restantes foram mantidos pela escassez de estudos numa área recente e ainda em crescimento, sendo suas conclusões analisadas com a devida cautela.

5.2.2 VALIDADE DE INTERNA

Refere-se ao grau em que é possível estabelecer, de forma confiável, uma relação de causa e efeito entre os fenômenos observados, sem influência de fatores externos ou vieses. Em uma análise da literatura, uma ameaça comum é o viés do pesquisador na seleção, codificação ou interpretação dos dados, que pode levar a conclusões causais incorretas. Uma ameaça significativa à validade interna neste estudo foi a introdução de viés de confirmação por parte do pesquisador. A análise exploratória foi conduzida com o objetivo principal de corroborar ou contrastar os achados do survey, e não como uma investigação totalmente aberta sobre tendências emergentes. Esse viés manifestou-se na priorização seletiva dos achados da literatura. Por exemplo, embora técnicas promissoras como o uso de Aprendizado por Reforço (RL) e a combinação de LLMs com ferramentas de análise estática tenham sido identificadas nos artigos, elas foram apenas mencionadas brevemente e não exploradas em profundidade, pois não se alinhavam diretamente com os tópicos abordados na estrutura de validação do survey.

5.2.3 VALIDADE EXTERNA

Trata-se do grau em que os resultados podem ser generalizados, dependendo da representatividade dos estudos primários. A principal ameaça à validade externa neste estudo é o viés de seleção e de publicação. A análise baseou-se exclusivamente em artigos que citaram a RSL de Zhou et al., o que pode ter excluído trabalhos relevantes que não fizeram essa citação específica.

5.2.4 VALIDADE A CONCLUSÃO

Está relacionada à robustez das conclusões tiradas a partir da análise dos dados. O tamanho reduzido da amostra (23 artigos) e a natureza não sistemática da análise fazem com que as conclusões sejam de caráter descritivo e qualitativo. A análise quantitativa apresentada visa ilustrar tendências, mas não permite inferências estatísticas robustas.. A confiança nas conclusões está atrelada à precisão do processo de extração de dados e à representatividade da amostra selecionada. Além disso, destaca-se a extração automatizada de informações com o Google NotebookLM, como principal ameaça à validade.

5.3 SÍNTESE DO CAPÍTULO

Este capítulo detalhou a análise exploratória de 23 artigos recentes, cujo objetivo foi complementar os achados do survey e atualizar a visão sobre o estado da arte. A análise confirmou a tendência observada no survey, evidenciando o domínio de modelos de arquitetura decoder-only tanto para detecção quanto para reparo na literatura recente. Em relação às técnicas, o prompt engineering mostrou-se como a abordagem mais frequentemente utilizada nos estudos, provavelmente por sua agilidade , embora o fine-tuning seja frequentemente citado como o método que produz melhores resultados. A análise também revelou que a falta de acurácia (L6) e a curadoria de datasets de alta qualidade (O1) são, respectivamente, a limitação e a oportunidade mais exploradas pela comunidade de pesquisa. Por fim, foram discutidas as ameaças à validade desta análise.

6. DISCUSSÃO DOS RESULTADOS

O objetivo deste capítulo é realizar uma triangulação dos dados provenientes da RSL, do survey e da análise exploratória com o intuito de construir uma visão atualizada sobre o estado da arte, confrontando os achados da literatura com a percepção prática de especialistas.

Para sintetizar os resultados desta triangulação, a Tabela 2 apresenta uma matriz de rastreamento dos principais achados deste trabalho, indicando os pontos de convergência e divergência entre as fontes. Em seguida, cada achado é discutido em detalhe.

Tabela 2. Matriz de rastreamento

Achado Principal	**RSL**	**Survey**	**Análise exploratória**
Arquiteturas decoder-only são as mais utilizadas para detecção e reparo	Δ	✔	✔
Fine-tuning é percebido como método de adaptação mais eficaz	✔	✔	✔
Fine-tuning é percebido como método de adaptação mais utilizado	✔	✔	Δ
L3 (Complexidade dos Dados) é uma limitação criticamente destacada ou estudada	✔	✔	Δ
L6 (Falta de acurácia e robustez) é uma limitação criticamente destacada ou estudada	✔	Δ	✔
O1 (Curadoria de datasets de alta qualidade) é uma oportunidade criticamente destacada ou estudada	✔	✔	✔
O2 (Análise em nível de repositório) é uma oportunidade criticamente destacada ou estudada	✔	✔	Δ
O3 (Desenvolvimento de LLMs especializados) é uma oportunidade criticamente destacada ou estudada	✔	✔	Δ

Legenda: ✔ = Convergência/Alinhamento; Δ = Divergência/Descompasso

Fonte: Próprio Autor

6.1 DISCUSSÃO DETALHADA DOS ACHADOS

6.1.1 ASCENSÃO DAS ARQUITETURAS DECODER-ONLY

A triangulação revela uma clara divergência em relação à RSL original no que tange às arquiteturas de modelo para detecção. Enquanto a RSL apontava um cenário onde modelos

encoder-only eram proeminentes, tanto o survey com especialistas quanto a análise da literatura recente convergem ao indicar que as arquiteturas decoder-only se tornaram a escolha predominante não apenas para reparo, mas também para detecção. Este achado sugere uma rápida evolução do campo, onde as capacidades de raciocínio e geração dos modelos mais modernos são agora vistas como essenciais, mesmo para tarefas de classificação.

6.1.2 TÉCNICAS DE ADAPTAÇÃO

Há um consenso entre as três fontes de que o fine-tuning é percebido como o método de adaptação mais eficaz para especializar um modelo e alcançar alta precisão. No entanto, uma divergência interessante surge em relação ao método mais utilizado. Enquanto a RSL e o survey apontam o fine-tuning como a prática mais comum, a análise da literatura recente mostra que o prompt engineering é, na verdade, mais frequentemente usado nos estudos primários, provavelmente por sua agilidade e não dependência de dados rotulados para treinamento.

6.1.3 LIMITAÇÕES

A análise das limitações expõe um descompasso significativo entre a percepção dos especialistas e o foco da comunidade de pesquisa. O survey destacou a L3 (Complexidade dos Dados) como a limitação mais crítica, refletindo uma preocupação com os desafios do mundo real, como a presença de vulnerabilidades interprocedurais e a dificuldade em lidar com falhas que se estendem por múltiplas unidades de código. Em contraste, a análise da literatura mostra que a L6 (Falta de acurácia e robustez) é, de longe, a área mais crítica e estudada, visto que sem acurácia, não há confiança. Isso sugere uma lacuna: enquanto os pesquisadores estão concentrados em resolver o problema fundamental de fazer os modelos funcionarem corretamente (acurácia), os especialistas já estão preocupados com o próximo nível de desafio, que é aplicar esses modelos a vulnerabilidades inerentemente complicadas e em cenários realistas.

6.1.4 OPORTUNIDADES E DIREÇÕES FUTURAS

A análise das oportunidades propostas no roadmap da RSL de Zhou et al. [5] revela um cenário complexo de alinhamentos e descompassos.

Existe um forte alinhamento entre o survey e a análise da literatura sobre a importância da O1 (Curadoria de datasets de alta qualidade). A RSL original já destacava a ausência de um benchmark de alta qualidade como um obstáculo significativo. O survey e a análise exploratória confirmam que esta continua sendo uma prioridade máxima, com a comunidade de pesquisa dedicando um esforço massivo para criar datasets mais robustos, reconhecendo que a melhoria dos dados é um pré-requisito para o avanço em todas as outras frentes.

O maior descompasso reside na O2 (Análise em nível de repositório). A RSL apontou a análise de granularidade maior como uma oportunidade clara, e o survey com especialistas validou isso de forma contundente, elegendo a O2 como a oportunidade mais relevante. No entanto, a análise da literatura recente mostra que esta é uma das áreas menos exploradas. Isso evidencia uma lacuna significativa: enquanto a indústria e os especialistas anseiam por soluções que compreendam o contexto completo de um projeto, a pesquisa acadêmica ainda está concentrada em resolver desafios em um escopo mais limitado. É importante frisar que a ascensão de modelos maiores, com janelas de contexto mais amplas, aparenta ser uma resposta natural a essa oportunidade, permitindo a análise de bases de código mais extensas. Contudo, não é possível afirmar com certeza que essa capacidade está sendo plenamente explorada para este fim, visto que ainda há poucos trabalhos direcionados especificamente para a análise em nível de repositório.

6.2 SÍNTESE DO CAPÍTULO

Este capítulo realizou a triangulação dos dados, cruzando os achados da RSL original de Zhou et al. [5], do survey com especialistas e da análise exploratória da literatura recente. A discussão evidenciou evolução do campo, com destaque para a ascensão das arquiteturas decoder-only para detecção, um ponto de divergência com a RSL original. Houve consenso de que o fine-tuning é o método de adaptação mais eficaz, mas divergência quanto ao mais utilizado, com a literatura recente favorecendo prompt engineering.

Nas limitações, especialistas priorizam a complexidade dos dados (L3), enquanto a pesquisa recente foca na falta de acurácia e robustez (L6). Quanto às oportunidades, a curadoria de datasets de alta qualidade (O1) foi unanimemente destacada, enquanto a análise em nível de repositório (O2) mostrou forte demanda prática, mas baixa exploração acadêmica. A triangulação consolidou tendências, expôs desalinhamentos e indicou direções para pesquisas futuras.

7. CONCLUSÕES

7.1 CONTRIBUIÇÕES

Esse trabalho buscou validar e atualizar os achados da RSL de Zhou et al. [5] sobre o uso de LLMs para detecção e reparo de vulnerabilidades em código, através de uma metodologia de triangulação que combinou a RSL original, um survey com especialistas e uma análise exploratória da literatura recente. As conclusões indicam uma área em rápida e constante evolução.

A principal conclusão é a consolidação de modelos de arquitetura decoder-only, que se tornaram a escolha predominante tanto para detecção quanto para reparo, superando o domínio anterior dos modelos encoder-only na detecção. Essa mudança reflete uma valorização crescente das capacidades de raciocínio e geração de código desses modelos. Em relação às técnicas, o Fine-tuning continua sendo a abordagem padrão para alta precisão, mas o Prompt Engineering e, principalmente, as abordagens híbridas que integram LLMs com ferramentas externas, ganham força e apontam para o futuro da área.

Apesar dos avanços, o estudo conclui que os desafios fundamentais persistem. A falta de acurácia e robustez, a necessidade crítica por datasets de alta qualidade e a limitação da análise em pequena granularidade são gargalos que continuam a dominar a agenda de pesquisa. Foi identificada uma lacuna entre as prioridades da pesquisa acadêmica e as necessidades práticas da indústria, especialmente no que tange à integração com o desenvolvedor, um tema de alta importância para os especialistas, mas ainda pouco explorado na literatura.

Por fim, destaca-se que a própria elaboração do survey constitui uma contribuição deste trabalho, pois oferece um instrumento de pesquisa detalhado que pode ser aplicado novamente no futuro em estudos de replicação para monitorar a contínua evolução das percepções de especialistas neste campo dinâmico.

7.2 TRABALHOS FUTUROS

Este trabalho validou a necessidade de uma atualização da RSL de Zhou et al. [5] e demonstrou, por meio de uma tentativa inicial, a inviabilidade de conduzi-la com rigor metodológico. A análise exploratória realizada forneceu indícios relevantes de mudanças no

cenário da pesquisa, mas não substitui uma atualização sistemática formal.

Portanto, permanece como trabalho futuro essencial a realização de uma atualização sistemática completa da RSL original, conduzida por uma equipe de pesquisadores e respaldada por um cronograma mais extenso. Tal atualização permitiria consolidar o vasto volume de novas publicações, revisar criticamente os métodos utilizados, reavaliar as tendências da área e atualizar as lacunas e oportunidades com o grau de profundidade e confiabilidade que uma RSL formal requer.

Outro trabalho futuro consiste na realização de uma análise manual aprofundada dos 23 estudos primários. Essa análise permitiria contrastar os resultados com aqueles obtidos a partir do NotebookLM, servindo como uma validação da precisão e eficácia da extração de dados assistida por LLM. Além disso, uma análise manual aberta possibilitaria a busca por achados e tendências emergentes que não foram o foco da análise atual. Isso ajudaria a mitigar o viés de confirmação, discutido nas ameaças à validade.

Por fim, sugere-se a replicação do survey conduzido nesta pesquisa. O tempo de disponibilização do formulário e os canais de divulgação utilizados limitaram a quantidade de respostas obtidas. Um estudo futuro com um período de coleta de dados mais longo e com meios de divulgação ampliados poderia aumentar as chances de obter uma amostra maior e mais diversificada de especialistas. Embora a aplicação de LLMs em segurança de software seja uma área ainda incipiente, seu grande potencial de crescimento torna a replicação deste survey uma forma valiosa de acompanhar a evolução das percepções da comunidade acadêmica e industrial.

Referências

[1] MICROSOFT. Microsoft Digital Defense Report 2024: The foundations and new frontiers of cybersecurity. 2024. Disponível em: https://www.microsoft.com/pt-br/security/security-insider/intelligence-reports/microsoft-digital-defense-report-2024. Acesso em: 12 jul. 2025.

[2] CHECK POINT SOFTWARE TECHNOLOGIES. Cyber Security Report 2024. 2024. Disponível em: https://www.checkpoint.com/resources/report-3854/report--cyber-security-report-2024. Acesso em: 12 jul. 2025.

[3] HOU, X.; ZHAO, Y.; LIU, Y.; YANG, Z.; WANG, K.; LI, L.; WANG, H. Large language models for software engineering: A systematic literature review. ACM Transactions on Software Engineering and Methodology, v. 33, n. 8, p. 1-79, 2024.

[4] ZHANG, Q.; FANG, C.; XIE, Y.; ZHANG, Y.; YANG, Y.; SUN, W.; CHEN, Z. A survey on large language models for software engineering. arXiv preprint arXiv:2312.15223, 2023. Disponível em: https://arxiv.org/abs/2312.15223. Acesso em: 12 jul. 2025.

[5] ZHOU, Xin; CAO, Sicong; SUN, Xiaobing; LO, David.. Large language model for vulnerability detection and repair: Literature review and the road ahead. ACM Transactions on Software Engineering and Methodology, v. 34, n. 5, p. 1-31, 2025.

[6] WOHLIN, Claes; MENDES, Emilia; FELIZARDO, Katia R.; KALINOWSKI, Marcos. Guidelines for the search strategy to update systematic literature reviews in software engineering. Information and Software Technology, v. 127, p. 106366, 2020.

[7] MOLLÉRI, Jefferson Seide; PETERSEN, Kai; MENDES, Emilia. An empirically evaluated checklist for surveys in software engineering. Information and Software Technology, v. 119, p. 106240, 2020.

[8] MENDES, Emilia; WOHLIN, Claes; FELIZARDO, Katia; KALINOWSKI, Marcos. When to update systematic literature reviews in software engineering. Journal of Systems and Software, v. 167, p. 110607, set. 2020.

[9] GARNER, Paul; HOPEWELL, Sally; CHANDLER, Jackie; MACLEHOSE, Harriet; AKL, Elie A.; BEYENE, Joseph; SCHÜNEMANN, Holger J. When and how to update systematic reviews: consensus and checklist. bmj, v. 354, p. i3507, 2016.

[10] KASUNIC, Mark. Designing an effective survey. [S.l.: s.n.], set. 2005.

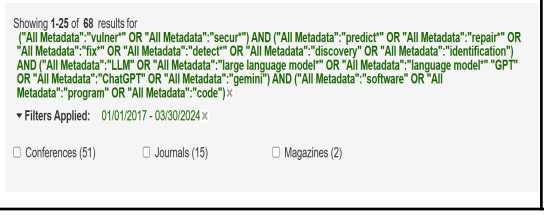
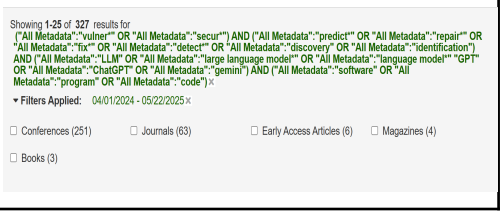
[11] RUNESON, Per; HÖST, Martin. Guidelines for conducting and reporting case study

research in software engineering. Empirical Software Engineering, v. 14, n. 2, p. 131–164, 2009.

[12] FELIZARDO, K. R.; DEIZEPE, A.; COUTINHO, D.; GOMES, G.; MEIRELES, M.; GEROSA, M.; STEINMACHER, I. On the difficulties of conducting and replicating systematic literature reviews studies using LLMs in software engineering. In: IEEE/ACM INTERNATIONAL WORKSHOP ON METHODOLOGICAL ISSUES WITH EMPIRICAL STUDIES IN SOFTWARE ENGINEERING (WSESE), 2025, [Anais...]. [S.l.]: IEEE, 2025. p. 20–23

[13] WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. Experimentation in Software Engineering: An Introduction. Norwell: Kluwer Academic Publishers, 2000.

APÊNDICE A - DETALHES DAS QUERIES DE BUSCA PRELIMINAR NO IEEE

Query	IEEE QUERY 2017-03/2024	IEEE QUERY 04/2024-05/2025
String de Busca	*("All Metadata":"vulner*" OR "All Metadata":"secur*") AND ("All Metadata":"predict*" OR "All Metadata":"repair*" OR "All Metadata":"fix*" OR "All Metadata":"detect*" OR "All Metadata":"discovery" OR "All Metadata":"identification") AND ("All Metadata":"LLM" OR "All Metadata":"large language model*" OR "All Metadata":"language model*" "GPT" OR "All Metadata":"ChatGPT" OR "All Metadata":"gemini") AND ("All Metadata":"software" OR "All Metadata":"program" OR "All Metadata":"code")*	
Filtro de Período	Publicado entre 01/01/2017 e 31/03/2024	Publicado entre 01/04/2024 e 31/05/2025
Número de Resultados	68 artigos	327 artigos
Imagem IEEE		

APÊNDICE B - SURVEY DE VALIDAÇÃO

Planilha com dados coletados a partir do survey: https://docs.google.com/spreadsheets/d/1ekOiLq1Rr0e2Vort2D4Vcu1Ct0BHVVFpObzQ2N9Y8nE/edit?usp=sharing

Validation Survey: LLMs in Vulnerability Detection and Repair in Code

Dear Colleague,

I am conducting academic research as part of the final graduation project at the Federal University of Pernambuco (UFPE), Brazil. The aim of this research is to critically validate with practitioners the main findings, limitations, and research opportunities identified in the Systematic Literature Review (SLR) conducted by Zhou et al. (2024) entitled “Large Language Model for Vulnerability Detection and Repair: Literature Review and the Road Ahead”.

The target audience for this survey consists of researchers and professionals working in the field of Software Engineering and Software Security who utilize Large Language Models (LLMs) to aid in detecting and/or repairing vulnerabilities in code.

Contact:

If you have any questions or would like more information about the research, please contact me at: gsf4@cin.ufpe.br

Informed Consent Form for Research Participation

You are invited to participate in an academic research study. Its objective is to critically validate the main findings, limitations, and research opportunities identified in the Systematic Literature Review (SLR) "Large Language Model for Vulnerability Detection and Repair" (Zhou et al., 2024).

Target Audience:
Researchers and professionals working in the field of Software Engineering or Software Security who utilize Large Language Models (LLMs) to aid in detecting and/or repairing vulnerabilities in code.

Data Collection and Use:
All information provided in this questionnaire will be collected anonymously and kept confidential. Your responses will be stored in a secure environment and used exclusively for scientific purposes related to this research. To promote open science and allow for future research, aggregated and anonymized data may be shared in scientific repositories, contributing to the transparency and reliability of this study.

Voluntary Participation and Rights:
Your participation in this survey is entirely voluntary. You may choose to withdraw at any point during the completion of the form, without needing to provide any justification and without any prejudice to you. To do so, simply close your browser window. Please note that after submitting your responses, it will not be possible to delete your participation data, as the responses are anonymous and cannot be traced back to an individual.

Privacy and Data Protection:
No personally identifiable information will be collected from you. The information gathered in this form will be used exclusively for scientific research purposes.

Access to Results:
If you wish, you may request a copy of the report with the results of this research by contacting the principal researcher via the email provided at the end of the survey (if applicable).

Declaration of Consent:
By proceeding and submitting your responses, you confirm that:

- You have read and understood the information presented above.
- You freely and voluntarily agree to participate in this research.
- You authorize the collection and use of the information provided for the purposes described.

I agree to the terms of participation in this research:

- Yes
- No

Section 1: Professional Profile

1. What is your primary area of work and/or research? (Select one or more)
    a. Software Engineering (General)
    b. Artificial Intelligence / Machine Learning
    c. Cybersecurity / Software Security
    d. Software Development
    e. Academic Research
    f. Other (Please specify):
2. In which country do you primarily work/research? (Open)

3. How long have you been working with software security (vulnerability detection/repair)?
    a. Less the 1 year
    b. 1 to 3 years
    c. 3 to 5 years
    d. More than 5 years

4. How long have you been working with LLMs?
    a. Less than 6 months
    b. 6 months to 1 year
    c. 1 to 2 years
    d. 2 to 3 years

e. More than 3 years

5. Have you read the Systematic Literature Review "Large Language Model for Vulnerability Detection and Repair"?
    a. Yes
    b. No
    c. Partially
    d. No, but I would like to read it.

6. How long have you been working with LLMs applied to software security (vulnerability detection/repair)?
    a. Less than 6 months
    b. 6 months to 1 year
    c. 1 to 2 years
    d. 2 to 3 years
    e. More than 3 years

Section 2: LLMs for Vulnerability Detection and Repair

7. In your opinion, which LLM architecture do you consider the most promising for software vulnerability detection?
    a. Encoder-only (e.g., CodeBERT, BERT)
    b. Encoder-decoder (e.g., CodeT5, T5)
    c. Decoder-only (e.g., GPT-3.5/4, CodeGPT)
    d. I don't know / No opinion
    e. Other (Please specify):

8. In your opinion, which LLM architecture do you consider the most promising for software vulnerability repair?
    a. Encoder-only (e.g., CodeBERT, BERT)
    b. Encoder-decoder (e.g., CodeT5, T5)
    c. Decoder-only (e.g., GPT-3.5/4, CodeGPT)
    d. I don't know / No opinion
    e. Other (Please specify):

9. To what extent the following LLMs are suitable for vulnerability detection and repair?(If you have no experience with a particular LLM listed, please leave that specific question unmarked) (Likert scale: 1=Strongly Disagree, 5=Strongly Agree)
    a. CodeBERT (Encoder-only): 1 | 2 | 3 | 4 | 5
    b. CodeT5 (Encoder-decoder): 1 | 2 | 3 | 4 | 5
    c. GPT-3.5 (Commercial, undisclosed architecture): 1 | 2 | 3 | 4 | 5
    d. GPT-4 (Commercial, undisclosed architecture): 1 | 2 | 3 | 4 | 5
    e. Other LLMs (any architecture): 1 | 2 | 3 | 4 | 5
        i. If you selected "Other LLMs", please specify which one(s):

Section 3: LLM Adaptation Techniques for Vulnerability Detection

10. In your opinion, please rank the following LLM adaptation techniques from 1 to 3, where 1 is the most promising and 3 is the least promising for vulnerability detection (Ensure

each rank is used only once across all techniques. If you select fewer than three techniques, rank only those you chose)

a. Fine-tuning (adjusting model parameters with labeled data)
b. Prompt Engineering (crafting prompts to guide the LLM without changing parameters)
c. Retrieval Augmentation (integrating with knowledge retrieval systems)

11. Please explain why you think so. (Open)

12. Regarding fine-tuning for vulnerability detection, how promising do you consider the following innovations? (Please rate only the innovations you have direct experience with or knowledge about. If you have no experience or knowledge of a particular innovation, please leave that specific question unmarked) (Likert scale: 1=Not promising at all, 5=Very promising)
    a. Data-centric Innovations (e.g., handling label imbalance, noise, scarcity):
        i. 1 | 2 | 3 | 4 | 5
    b. Combination with Program Analysis (e.g., using AST, PDG, Program Slicing to extract structural features):
        i. 1 | 2 | 3 | 4 | 5
    c. Combination with Other Deep Learning Modules (e.g., GNN, Bi-LSTM to handle structural features or input length constraints):
        i. 1 | 2 | 3 | 4 | 5
    d. Domain-specific Pre-training (e.g., Masked Language Modeling, Contrastive Learning on vulnerability data):
        i. 1 | 2 | 3 | 4 | 5
    e. Causal Learning (addressing lack of robustness, promoting causality-based prediction):
        i. 1 | 2 | 3 | 4 | 5
    f. Other (Please specify):
        i. 1 | 2 | 3 | 4 | 5

13. Regarding prompt engineering for vulnerability detection, how promising do you consider the following strategies? (Please rate only the strategies you have direct experience with or knowledge about. If you have no experience or knowledge of a particular strategy, please leave that specific question unmarked)(Likert scale: 1=Not promising at all, 5=Very promising)
    a. Zero-shot Prompting (prompts without examples):
        i. 1 | 2 | 3 | 4 | 5
    b. Few-shot Prompting (prompts with a few examples):
        i. 1 | 2 | 3 | 4 | 5
    c. Chain-of-thought (CoT) Prompting (guiding the model to think step-by-step):
        i. 1 | 2 | 3 | 4 | 5
    d. Inclusion of Auxiliary Information (e.g., task description, LLM role, vulnerability/program analysis information):
        i. 1 | 2 | 3 | 4 | 5

Section 4: LLM Adaptation Techniques for Vulnerability Repair

14. In your opinion, which of the following LLM adaptation techniques do you consider the most promising for vulnerability repair?

a. Fine-tuning (adjusting model parameters with labeled data)
b. Prompt Engineering (crafting prompts to guide the LLM without changing parameters)
c. I don't know / No opinion

15. Regarding fine-tuning for vulnerability repair, how promising do you consider the following innovations? (Please rate only the innovations you have direct experience with or knowledge about. If you have no experience or knowledge of a particular innovation, please leave that specific question unmarked) (Likert scale: 1=Not promising at all, 5=Very promising)
    a. Data-centric Innovations (e.g., incorporating AST, vulnerability descriptions, fix commits, handling input length limits):
        i. 1 | 2 | 3 | 4 | 5
    b. Model-centric Innovations (e.g., revising the Transformer architecture, vulnerability queries):
        i. 1 | 2 | 3 | 4 | 5
    c. Domain-specific Pre-training (e.g., pre-training on bug-fixing tasks):
        i. 1 | 2 | 3 | 4 | 5
    d. Reinforcement Learning (using syntactic/semantic rewards for training optimization):
        i. 1 | 2 | 3 | 4 | 5

16. Regarding prompt engineering for vulnerability repair, how promising do you consider the following strategies? (Please rate only the strategies you have direct experience with or knowledge about. If you have no experience or knowledge of a particular strategy, please leave that specific question unmarked)(Likert scale: 1=Not promising at all, 5=Very promising)
    a. Zero-shot Prompting (prompts without examples):
        i. 1 | 2 | 3 | 4 | 5
    b. Few-shot Prompting (prompts with a few examples):
        i. 1 | 2 | 3 | 4 | 5
    c. Inclusion of Auxiliary Information (e.g., vulnerability descriptions, location, semantics, program analysis):
        i. 1 | 2 | 3 | 4 | 5

Section 5: Dataset Characteristics and Deployment Strategies

17. How important is it to investigate vulnerability detection and repair at higher levels of granularity than function/line? (Likert scale: 1=Not important at all, 5=Very important)
    a. 1 | 2 | 3 | 4 | 5
18. Please explain why you think so. (Open)

19. Regarding data quality, how critical do you consider the following issues for advancing research in LLMs for security? (Please rate only the issues you have knowledge about. If you have no knowledge of a particular issue, please leave that specific question unmarked)(Likert scale: 1=Not critical al all, 5=Extremely critical)
    a. Dependence on labels generated by automated rules or tools (heuristic labels), often leading to noise or inaccuracies in detection datasets?

    i. 1 | 2 | 3 | 4 | 5
b. Lack of test cases to verify the correctness of vulnerability fixes and prevent new bugs in repair datasets?
    i. 1 | 2 | 3 | 4 | 5
c. Concern about "data contamination" (evaluation datasets present in the pre-training corpus of LLMs)?
    i. 1 | 2 | 3 | 4 | 5

20. Regarding the deployment of LLM-based solutions, how important do you consider the following aspects? (Likert scale: 1=Not important at all, 5=Very important)
    a. Ability to interact and collaborate with developers (e.g., feedback, explanations)?
        i. 1 | 2 | 3 | 4 | 5
    b. Continuous and seamless integration into developer workflows and tools (e.g., IDEs)?
        i. 1 | 2 | 3 | 4 | 5
    c. High accuracy and robustness against perturbations/attacks?
        i. 1 | 2 | 3 | 4 | 5

Section 6: Overall Perceptions, Limitations, and Future Directions

21. In your opinion, what is the maturity level of applying LLMs for Software Vulnerability Detection?
    a. Very incipient (only basic research)
    b. Incipient (some prototypes, little practical application)
    c. I don't know / No opinion
    d. Moderate (some products/tools, but with significant challenges)
    e. Advanced (widespread use, reliable results)

22. In your opinion, what is the maturity level of applying LLMs for Software Vulnerability Repair?
    a. Very incipient (only basic research)
    b. Incipient (some prototypes, little practical application)
    c. I don't know / No opinion
    d. Moderate (some products/tools, but with significant challenges)
    e. Advanced (widespread use, reliable results)

23. In your opinion, which of the following current limitations in LLM research for vulnerability detection and repair do you consider the most critical?
    a. Small input granularity (focus on line/function)
    b. Lack of high-quality vulnerability datasets
    c. Suboptimal performance due to vulnerability data complexity (e.g., inter-procedural, less frequent CWE types)
    d. Dependence on lightweight LLMs (<1B parameters)
    e. Lack of deployment consideration (developer interaction, workflow integration)
    f. Lack of high accuracy and robustness
    g. Other (Please specify):

24. Please explain why you think so. (Open)
25. Regarding the research opportunities and future directions proposed in the SLR's roadmap, please rank the following topics from 1 to 3, where 1 is the most promising and 3 is the least promising for vulnerability detection and repair (Ensure each rank is used only once across all techniques. If you select fewer than three techniques, rank only those you chose)
    a. Curation of high-quality benchmark datasets for vulnerability detection
    b. Repository-level (higher granularity) vulnerability detection/repair
    c. Development of vulnerability-specialized LLMs
    d. Use of more advanced LLM techniques (e.g., LLM Agents, External Tool Use, iterative/recursive/adaptive RAG)
    e. Development of deployment-ready features (user interaction, workflow integration)
    f. Other (Please specify)

26. Please explain why you think so. (Open)
27. Would you like to add any comments, insights, or suggestions about the use of LLMs for vulnerability detection and repair that were not covered in this survey? (Open-ended question)

APÊNDICE C - TABELA DE ESTUDOS SELECIONADOS PARA ANÁLISE EXPLORATÓRIA

Planilha completa, com relação entre os estudos e as QPs: https://docs.google.com/spreadsheets/d/1nqoRmts5zbhxXl_rBBq6Ua8Mg0iRp2gMnAbytCL35TQ/edit?usp=sharing

Código	Referência
E04	SHESTOV, A.; LEVICHEV, R.; MUSSABAYEV, R.; MASLOV, E.; ZADOROZHNY, P.; CHESHKOV, A.; KRASSOVITSKIY, A. Finetuning large language models for vulnerability detection. IEEE Access, 2025.
E13	YANG, X.; ZHU, W.; PACHECO, M.; ZHOU, J.; WANG, S.; HU, X.; LIU, K. Code Change Intention, Development Artifact, and History Vulnerability: putting them together for Vulnerability Fix Detection by LLM. Proceedings of the ACM on Software Engineering, v. 2, n. FSE, p. 489-510, 2025
E15	FAKIH, M.; DHARMAJI, R.; BOUZIDI, H.; ARAYA, G. Q.; OGUNDARE, O.; FARUQUE, M. A. A. LLM4CVE: enabling iterative automated vulnerability repair with large language models. 2025. Disponível em: https://arxiv.org/abs/2501.03446. Acesso em: 9 ago. 2025.
E19	MAO, Z.; LI, J.; JIN, D.; LI, M.; TEI, K. Multi-role consensus through LLMs discussions for vulnerability detection. In: IEEE INTERNATIONAL CONFERENCE ON SOFTWARE QUALITY, RELIABILITY, AND SECURITY COMPANION (QRS-C), 24., 2024, Cambridge. Anais [...]. Piscataway: IEEE, 2024. p. 1318-1319.
E21	YILDIZ, A.; TEO, S. G.; LOU, Y.; FENG, Y.; WANG, C.; DIVAKARAN, D. M. Benchmarking LLMs and LLM-based agents in practical vulnerability detection for code repositories. 2025. Disponível em: https://arxiv.org/abs/2503.03586. Acesso em: 9 ago. 2025.
E22	HU, X.; NIU, F.; CHEN, J.; ZHOU, X.; ZHANG, J.; HE, J.; LO, D. Assessing and advancing benchmarks for evaluating large language models in software engineering tasks. 2025. Disponível em: https://arxiv.org/abs/2505. Acesso em: 9 ago. 2025.
E23	LI, Y.; SHEZAN, F. H.; WEI, B.; WANG, G.; TIAN, Y. SoK: towards effective automated vulnerability repair. 2025. Disponível em: https://doi.org/10.48550/arXiv.2501.18820. Acesso em: 9 ago. 2025.
E28	GONG, J.; DUAN, N.; TAO, Z.; GONG, Z.; YUAN, Y.; HUANG, M. How well do large language models serve as end-to-end secure code producers? 2024. Disponível em: https://arxiv.org/abs/2408. Acesso em: 9 ago. 2025.
E29	DAI, S. C.; XU, J.; TAO, G. A comprehensive study of LLM secure code generation. 2025. Disponível em: https://arxiv.org/abs/2503.15554. Acesso em: 9 ago. 2025.
E30	NONG, Y.; YANG, H.; CHENG, L.; HU, H.; CAI, H. APPATCH: automated adaptive prompting large language models for real-world software vulnerability patching. 2024. Disponível em: https://arxiv.org/abs/2408.13597. Acesso em: 9 ago. 2025.
E32	YU, J.; LIANG, P.; FU, Y.; TAHIR, A.; SHAHIN, M.; WANG, C.; CAI, Y. An insight into security

	code review with LLMs: capabilities, obstacles and influential factors. 2024. Disponível em: https://arxiv.org/abs/2401. Acesso em: 9 ago. 2025.
E37	MAO, Q.; LI, Z.; HU, X.; LIU, K.; XIA, X.; SUN, J. Towards explainable vulnerability detection with large language models. 2024. Disponível em: https://arxiv.org/abs/2406.09701. Acesso em: 9 ago. 2025.
E43	WEYSSOW, M.; YANG, C.; CHEN, J.; LI, Y.; HUANG, H.; WIDYASARI, R.; LO, D. R2Vul: learning to reason about software vulnerabilities with reinforcement learning and structured reasoning distillation. 2025. Disponível em: https://arxiv.org/abs/2504.04699. Acesso em: 9 ago. 2025.
E44	LAN, X.; MENZIES, T.; XU, B. Smart cuts: enhance active learning for vulnerability detection by pruning bad seeds. 2025. Disponível em: https://arxiv.org/abs/2506.20444. Acesso em: 9 ago. 2025.
E46	HUYNH, L.; ZHANG, Y.; JAYASUNDERA, D.; JEON, W.; KIM, H.; BI, T.; HONG, J. B. Detecting code vulnerabilities using LLMs. In: ANNUAL IEEE/IFIP INTERNATIONAL CONFERENCE ON DEPENDABLE SYSTEMS AND NETWORKS (DSN), 55., 2025, Brisbane. Anais [...]. Piscataway: IEEE, 2025. p. 401-414.
E47	WEN, X. C.; YANG, Y.; GAO, C.; XIAO, Y.; YE, D. Boosting vulnerability detection of LLMs via curriculum preference optimization with synthetic reasoning data. 2025. Disponível em: https://arxiv.org/abs/2506.07390. Acesso em: 9 ago. 2025.
E48	SIMONI, M.; FONTANA, A.; ROSSOLINI, G.; SARACINO, A. Improving LLM reasoning for vulnerability detection via group relative policy optimization. 2025. Disponível em: https://arxiv.org/abs/2507.03051. Acesso em: 9 ago. 2025.
E49	QIN, W.; SUO, L.; LI, L.; YANG, F. Advancing software vulnerability detection with reasoning LLMs: DeepSeek-R1's performance and insights. Applied Sciences, v. 15, n. 12, p. 6651, 2025.
E58	FARR, D.; TALTY, K.; FARR, A.; STOCKDALE, J.; CRUICKSHANK, I.; WEST, J. Expert-in-the-loop systems with cross-domain and in-domain few-shot learning for software vulnerability detection. 2025. Disponível em: https://arxiv.org/abs/2506.10104. Acesso em: 9 ago. 2025.
E59	DHARMAJI, R. Large language models for programming industrial control systems and mitigating real-world software vulnerabilities. 2024. Dissertação (Mestrado) – University of California, Irvine, Irvine, 2024.
E74	ZIA, I. Towards securing systems: leveraging generative AI for enhanced vulnerability detection and correction. 2024. Dissertação (Mestrado) – California State University, Long Beach, Long Beach, 2024.
E75	JIE, L.; ZHANG, L.; YAN, S.; CHEN, Y.; YAN, F.; FENG, Y. Modified visual language model for robust use-after-free vulnerability detection. 2024. Disponível em: https://arxiv.org/abs/2404.14442. Acesso em: 9 ago. 2025.
E76	SANTNER, Elias. Evaluating ChatGPT-4 for Automatic Vulnerability Repair in C/C++ Code. 2025. Dissertação (Mestrado em Informática) – Alpen-Adria-Universität Klagenfurt , Klagenfurt, 2025.

APÊNDICE D - PROCESSO DE EXTRAÇÃO DE DADOS ASSISTIDA POR LLM

Este apêndice detalha o processo metodológico utilizado para a extração de dados dos 23 estudos primários da análise exploratória, conforme descrito na seção 5.2.

1. Ferramenta utilizada
    - Plataforma: Google NotebookLM
    - Modelo de Linguagem Subjacente: Gemini Pro
2. Preparação das fontes
    - Para cada um dos 23 estudos, o arquivo PDF correspondente foi carregado como uma fonte de dados no ambiente do NotebookLM.
    - O NotebookLM indexa o conteúdo do PDF fornecido, permitindo que o modelo de linguagem baseie suas respostas exclusivamente nas informações contidas nos documentos.
3. Prompt de extração

Para cada artigo carregado, o seguinte prompt foi usado. O prompt foi projetado para mapear as informações do artigo diretamente para as Questões de Pesquisa (QPs), Limitações (L1-L6) e Oportunidades (O1-O5) definidas neste trabalho.

Prompt:

Você é um assistente de pesquisa. Sua tarefa é analisar o artigo acadêmico fornecido e extrair informações, mapeando seu conteúdo para os tópicos específicos das Questões de Pesquisa (QPs) listadas
ARTIGO A SER ANALISADO: <Código do artigo>.pdf
Baseado APENAS no ARTIGO A SER ANALISADO responda:
QP1 (Modelos):
Detecção: Identifique modelos/arquiteturas de LLM (ex: GPT-4, Llama 3) e o tipo dessa arquitetura (encoder-only, decoder-only, encoder-decoder) para a tarefa de Detecção.
Reparo: Identifique modelos/arquiteturas e o tipo de arquitetura de LLM para a tarefa de Reparo.
QP2 (Técnicas):
Detecção: Identifique técnicas de adaptação (ex: Fine-tuning, RAG) para a tarefa de Detecção.
Reparo: Identifique técnicas de adaptação para a tarefa de Reparo.
QP3 (Limitações): Verifique se o artigo aborda alguma das seguintes limitações da literatura:
L1_Pequena_Granularidade: Foco em função/linha.

L2_Falta_Dataset_Qualidade: Problemas com datasets (ruído, falta de testes).
L3_Complexidade_Dados: Lidar com vulnerabilidades complexas (ex: interprocedurais).
L4_Dependencia_LLMs_Leves: Foco em LLMs com menos de 1B de parâmetros.
L5_Falta_Deploy: Falta de integração com desenvolvedor/workflow.
L6_Falta_Acuracia_Robustez: Foco em melhorar acurácia ou robustez (ex: testes adversariais).
QP4 (Oportunidades): Verifique se o artigo explora alguma das seguintes oportunidades:
O1_Curadoria_Dataset: Propor ou usar um novo benchmark/dataset de alta qualidade.
O2_Nivel_Repositorio: Aplicar a análise em um nível de granularidade maior (ex: repositório).
O3_LLMs_Especializados: Desenvolver ou discutir LLMs específicos para vulnerabilidades.
O4_Uso_Avancado_LLM: Usar técnicas avançadas como LLM Agents ou Tool Use.
O5_Features_Deploy: Propor funcionalidades prontas para deploy (ex: plugins de IDE).
Para cada limitação ou oportunidade avalie de acordo com os status:
Explorada: implementa ou avança
Mencionada: ainda é uma limitação ou oportunidade
Não Mencionada: Não discutida
Novas limitações e oportunidades: Liste novas limitações e oportunidades mencionadas no artigo

4. Processo de Validação Humana e Refinamento

As informações geradas pelo Google Notebook LM sempre referenciam o trecho do texto original responsável por aquela informação.

Para as QPs 1 e 2, uma verificação era feita nos trechos referenciados para comprovar se de fato o artigo utilizou as arquiteturas e técnicas de adaptação listadas pela LLM. Na grande maioria dos casos, o resumo trazido pela LLM se mostrou correto.

Para as QPs 3 e 4, quando a LLM categorizava uma limitação ou oportunidade como “Explorada”, uma leitura das referências era feita para comprovar se de fato houve a exploração. Aqui, o índice de alucinação foi maior. Algumas das oportunidades e limitações categorizadas com “Exploradas” eram apenas mencionadas.

Por fim, uma planilha foi organizada, resumindo as arquitetura mencionadas para detecção e reparo, as técnicas utilizadas e a lista de limitações ou oportunidades que foram listadas no estudo primário.

APÊNDICE E - ARTEFATOS DA CHECAGEM DE ATUALIZAÇÃO DA RSL

Este apêndice apresenta a planilha utilizada para realizar o forward snowballing mencionado no tópico 3.2.3

1. Planilha dos estudos primários:
    - Planilha contendo a lista dos 58 estudos-semente da RSL de Zhou et al
    - Link: https://docs.google.com/spreadsheets/d/1Jvz5mHY8HqD2-wGE94X5uoWFZSM_A-vLcA8ZmL4clvc/edit?gid=799309806#gid=799309806

2. Planilha forward snowballing
    - Planilha contendo a lista dos estudos encontrados a partir do foward snowballing e seus respectivos títulos e resumos
    - Link: https://docs.google.com/spreadsheets/d/1Jvz5mHY8HqD2-wGE94X5uoWFZSM_A-vLcA8ZmL4clvc/edit?gid=1519900238#gid=1519900238