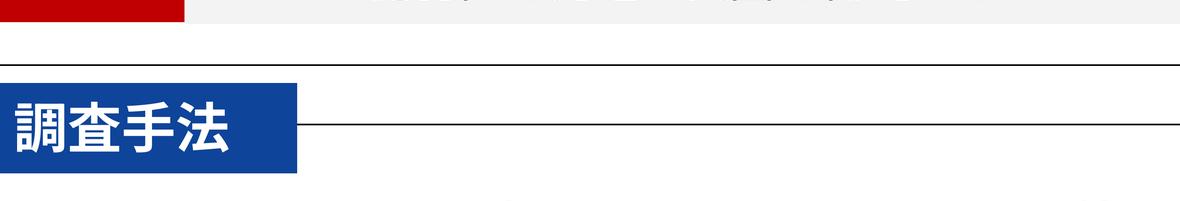
脆弱性混入要因の特定に向けた GitHub Pull Requestに含まれる特徴量の分析

◯ 石田優希　上原哲太郎（立命館大学）

研究背景

コードレビューとは

コード変更に対して他の開発者がその妥当性や品質を検査する仕組み



- ✓ OSSプロジェクトに加え、商用組織でも広く採用
- ✓ バグや脆弱性を防止する効果があることが示されている

➡ 特定の技術スタックに依存しない**脆弱性発生防止手法**

- ✓ 一方で、レビューで脆弱性が見逃されてしまうこともある

➡ レビューにおける脆弱性見逃しの要因を明らかにし
脆弱性の発生を防ぐ技術開発の基礎とする

目標　コードレビューに含まれる特徴量に着目し
脆弱性を見逃す要因を特定する

調査手法

ケース・コントロール研究を用いたPull Requestの比較

- ✓ GitHub上で開発されているOSSを対象としてケース群、コントロール群となるPull Request（以降、PR）を収集

ケース群

レビューで脆弱性の発生を**見逃した**PR

コントロール群

レビューで脆弱性の発生を**発見した**PR

両者のPRに含まれる様々な特徴量を比較する

PRが持つ特徴量

コードの変更行数 / 変更ファイル数 / コメントの数 / レビュアーの人数 / マージまでの期間 / 人的属性

レビュアーやレビュイーの持つ特徴量

プロジェクトへの貢献度 / 過去のレビュー数 / レビューの経験値 / PRへのコメント数 / 開発歴

データセットの構築

- ✓ スター数とPR数を基準としてプロジェクトを選定
- ✓ NVD(脆弱性情報DB)やPR内の議論を参考にデータを収集

owner/repo (組織名/リポジトリ名)	vercel/ next.js	elastic/ elasticsearch	mattermost/ mattermost
総PR数（Close含む）	32880	97168	23304
ケース群の数	14	13	32
コントロール群の数	177	287	315

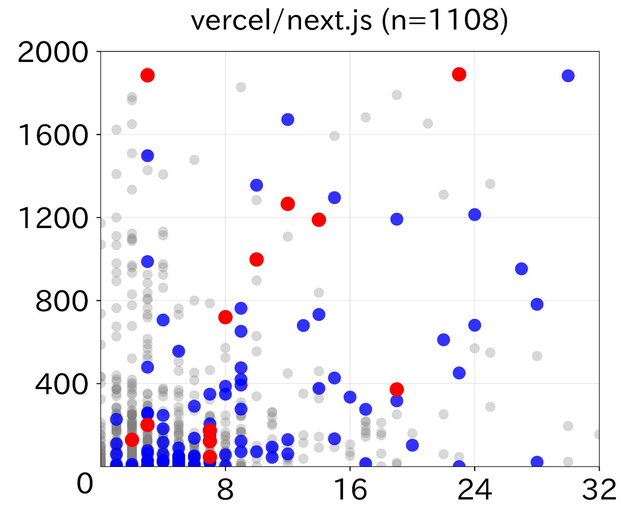
※ データ収集は2026年1月21日まで実施

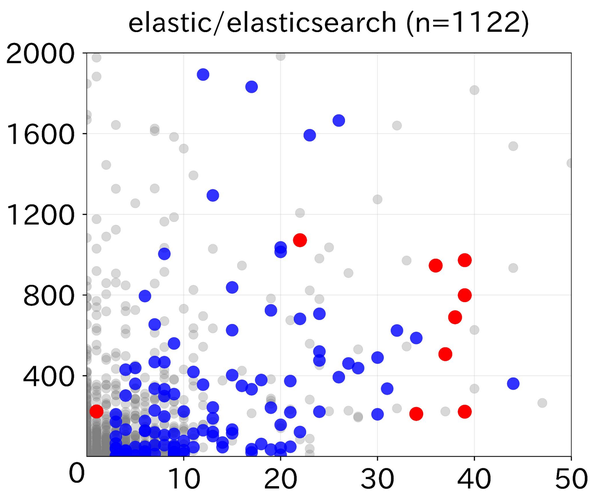
調査

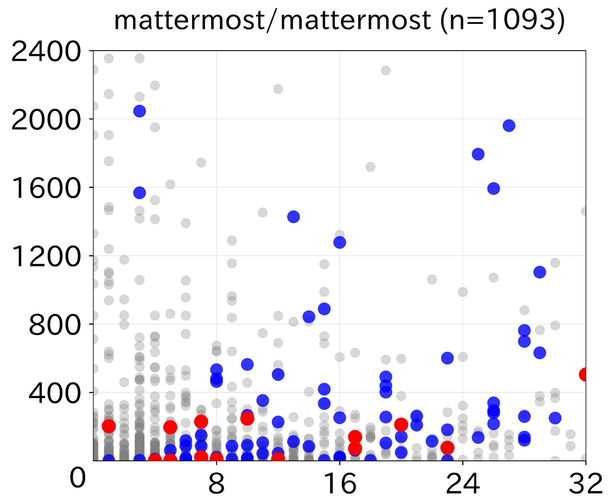
分析1: コードの変更行数とコメント数での分析

ケース（見逃し）
コントロール（発見）
両者に属さないPR

X軸: レビューコメントの数
Y軸: 変更行数
(削除+追加の合計)





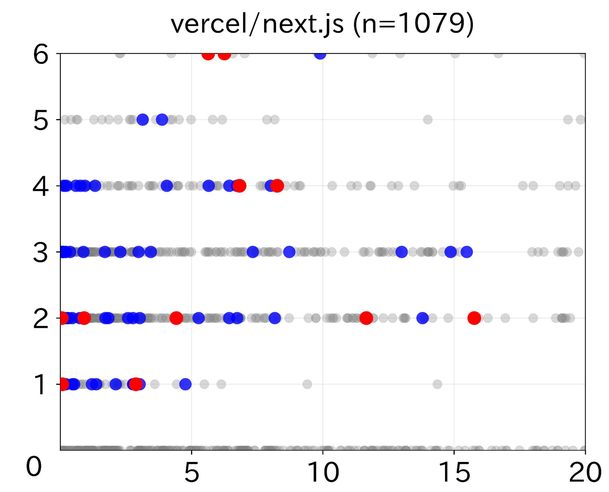


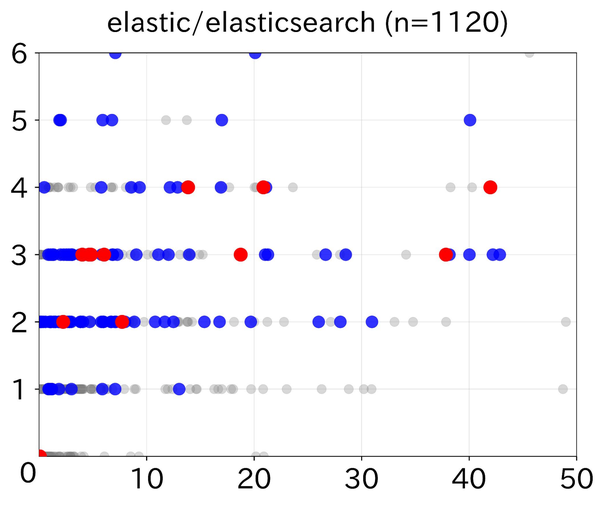
- ✓ 変更行数の多さやレビューコメントの数と脆弱性混入の間に有意な関係は確認されなかった

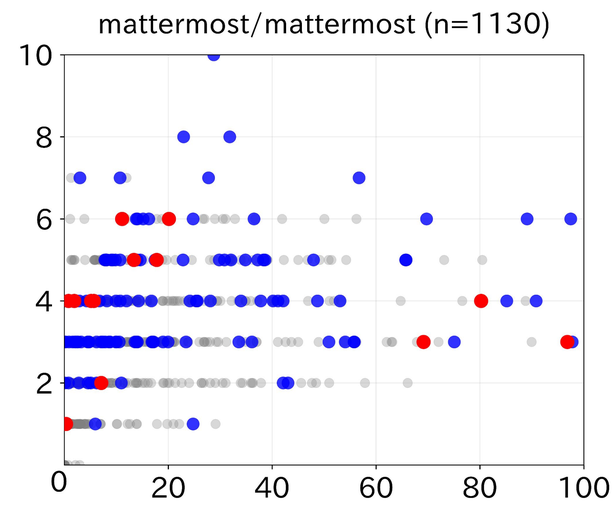
分析2: レビュアーの人数とレビュー期間での分析

ケース（見逃し）
コントロール（発見）
両者に属さないPR

X軸: レビュー期間（日）
Y軸: レビュアー数（人）







- ✓ 分析1と同様に脆弱性混入との間に有意な関係は確認されず

結果　分析の結果、脆弱性混入と有意な関連を示す
特徴は確認されなかった

課題　ケース群となるデータ数が限られており
明確な傾向を把握しにくい

今後の展望

- ✓ レビュアーやレビュイーの人的属性に着目した分析の実施
- ✓ ケース群となるデータの拡充
- ✓ 分析対象リポジトリの拡大