

Detecting False Alarms from Automatic Static Analysis Tools: How Far are We?

Hong Jin Kang

Singapore Management University
Singapore, Singapore
hjkang.2018@phdcs.smu.edu.sg

Khai Loong Aw

Singapore Management University
Singapore, Singapore
klaw.2020@scis.smu.edu.sg

David Lo

Singapore Management University
Singapore, Singapore
davidlo@smu.edu.sg

44th International Conference on Software Engineering (ICSE 2022)

発表者 花田

背景

- Javaプログラムのバグを検出するために自動静的解析ツール（ASAT）が導入
- 現在では多くのプロジェクトが，ASATを採用している

問題点

- 警告が本当のバグであることを保証できていない
- ASATの警告には
 - 高い誤報率
 - 実質的にコードに影響する可能性が低いもの } が存在する
- そのため、多くの開発者はASATによる警告を適切なものであると認識していません^[19, 43, 52]

[19] BrittanyJohnson,YoonkiSong,EmersonR.Murphy-Hill,andRobertW.Bow- didge. 2013. Why don't software developers use static analysis tools to find bugs?. In 35th International Conference on Software Engineering, (ICSE 2013). IEEE ComputerSociety,672–681. <https://doi.org/10.1109/ICSE.2013.6606613>

[43] Caitlin Sadowski, Edward Aftandilian, Alex Eagle, Liam Miller-Cushon, and Ciera Jaspan. 2018. Lessons from building static analysis tools at Google. Commun. ACM61,4(2018),58–66. <https://doi.org/10.1145/3188720>

[52] Carmine Vassallo, Sebastiano Panichella, Fabio Palomba, Sebastian Proksch, Harald C Gall, and Andy Zaidman. 2020. How developers engage with static analysis tools in different contexts. Empirical Software Engineering (EMSE) 25, 2 (2020), 1419–1457. <https://doi.org/10.1007/s10664-019-09750-5>

背景

問題解決の提案

- 研究者たち^[15, 53]は,問題点に対処するため,
誤報を除去し,開発者が有効だと考える警告を識別する技術を提案
- 研究者たちが文献で提案した特徴をWangら^[53]は,系統的評価を行い,
対処可能なFindbugs警告を検出するための最も重要な特徴を特定
→23の特徴からなる『Golden Features』

[15] Sarah Heckman and Laurie Williams. 2011. A systematic literature review of actionable alert identification techniques for automated static code analysis. Information and Software Technology (IST) 53, 4 (2011), 363–387. <https://doi.org/10.1016/j.infsof.2010.12.007>

[53] Junjie Wang, Song Wang, and Qing Wang. 2018. Is there a "golden" feature set for static warning identification?: an experimental evaluation. In 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, (ESEM2018).ACM,17:1–17:10. <https://doi.org/10.1145/3239235.3239523>

背景

- 最近の研究では、『Golden Features』を用いた機械学習非常に効果的,完璧な性能を達成する
- Golden Features を用い,SVMなどの機械学習を実行
→最大96%のRecallと98%のPrecision,99.5%のAUCを達成
- 完全予測器はRecall,Precision,AUCが100%であり,
Golden Featuresを用いた機械学習技術はほぼ完全

Table 1: The Golden Features studied in prior work [53, 56, 58]. A *warning context* is defined [53] as the difference of the number of actionable warnings and false alarms divided by the total number of warnings reported in a given method/file, or for a warning pattern. We provide more descriptions of each feature in our replication package [1].

Feature type	Feature
Warning combination	warning context in method warning context in file warning context for warning type defect likelihood for warning pattern discretization of defect likelihood average lifetime for warning type
Code characteristics	comment-code ratio method depth file depth # methods in file # classes in package
Warning characteristics	warning pattern warning type warning priority package
File history	file age file creation developers
Code analysis	parameter signature method visibility
Code history	LOC added in file (last 25 revisions) LOC added in package (past 3 month)
Warning history	warning lifetime by revision

目的

- Golden Featuresは高い性能を示すことが分かっていますが,なぜそのような効果があるのかは分かっていません.
- 「 Golden Features 」 の強力な性能をより良く理解するために,詳細な分析を行った.

RQ

RQ1:なぜ, Golden Featuresは機能するのか？

- Wangら^[53]とYangら^[56]の実験に使用されたデータセットを使用
- SVMを用いて先行研究の結果を再現
 - Golden Featuresによる予測力を検証
- より単純な分類器に切り替え,実験データを分析
 - なぜ分類器の選択が先行研究の結果に影響を与えなかったか理解

RQ2:closed-warning heuristicは警告の発見方法として適切なのか？

[53] Junjie Wang, Song Wang, and Qing Wang. 2018. Is there a "golden" feature set for static warning identification?: an experimental evaluation. In 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, (ESEM2018).ACM,17:1–17:10. <https://doi.org/10.1145/3239235.3239523>

[56] Xueqi Yang, Jianfeng Chen, Rahul Yedida, Zhe Yu, and Tim Menzies. 2021. Learning to recognize actionable static code warnings (is intrinsically easy). Empirical Software Engineering (EMSE) 26, 3 (2021), 56. <https://doi.org/10.1007/s10664-021-09948-6>

RQ1 結果（１） なぜ, Golden Featuresは機能するのか？

Wangら^[53]とYangら^[56]の実験に使用されたデータセットを使用
→SVMを用いて先行研究の結果を再現

Table 3: Effectiveness of an SVM using the Golden Features after removing the leaked features and removing the duplicate warnings between the training and testing dataset. The numbers in parentheses are the F1 obtained by the baseline classifier that predicts all warnings are actionable.

Project	All Golden Features		- leaked features		- data duplication		- leak, duplication	
	F1	AUC	F1	AUC	F1	AUC	F1	AUC
ant	0.94 (0.09)	1.00	0.11 (0.09)	0.67	-	-	-	-
cassandra	0.92 (0.24)	1.00	0.45 (0.24)	0.86	0.9 (0.41)	0.99	0.29 (0.41)	0.54
commons	0.65 (0.10)	0.99	0.16 (0.10)	0.65	0.75 (0.25)	0.97	0.11 (0.25)	0.49
derby	0.95 (0.09)	1.00	0.39 (0.09)	0.93	0.97 (0.28)	0.97	0.30 (0.28)	0.59
jmeter	0.94 (0.38)	0.99	0.53 (0.38)	0.76	1.00 (0.14)	1.00	0.25 (0.14)	1.00
lucene-solr	0.87 (0.51)	0.97	0.59 (0.51)	0.74	0.87 (0.53)	0.98	0.23 (0.53)	0.62
maven	0.86 (0.07)	1.00	0.27 (0.07)	0.9	0.95 (0.24)	0.99	0.27 (0.24)	0.58
tomcat	0.93 (0.37)	1.00	0.48 (0.37)	0.73	0.95 (0.70)	1.00	0.65 (0.70)	0.39
phoenix	0.89 (0.25)	1.00	0.42 (0.25)	0.78	0.83 (0.37)	0.99	0.40 (0.37)	0.63
Average	0.88 (0.23)	1.00	0.38 (0.23)	0.76	0.90 (0.37)	0.99	0.31 (0.37)	0.59

RQ1 結果（2）

Golden Featuresによる予測力を検証

→警告コンテキストの実装に微妙なデータリークがあることが判明

➤警告コンテキストが,実用的かどうかの判定基準に不備

leaked features

Feature type	Feature
Warning combination	warning context in method
	warning context in file
	warning context for warning type
	defect likelihood for warning pattern
	discretization of defect likelihood
	average lifetime for warning type

leaked featuresのみを用いたベースラインが
Precision 0.79, Recall 0.94, F1 0.83

➤Golden Featuresが実験において高い性能を示したのは,
leaked featuresに大きく依存しており,
その有効性を楽観的に見積もっている

Table 4: Average Precision (Prec.), Recall, and F1 of various approaches on the original dataset by Yang et al.

Technique	Prec.	Recall	F1
Golden Features SVM	0.84	0.94	0.88
- leaked features	0.26	0.70	0.38
- data duplication	0.88	0.93	0.90
- data duplication and leaked features	0.27	0.57	0.31
+ reimplemented leaked features	0.32	0.57	0.38
Golden Features kNN (with k=10)	0.91	0.57	0.68
Golden Features kNN (with k=5)	0.86	0.72	0.78
Golden Features kNN (with k=3)	0.87	0.78	0.82
Golden Features kNN (with k=1)	0.87	0.90	0.84
Only leaked features SVM	0.79	0.94	0.83
Repeat label from training dataset	0.72	0.80	0.75

RQ 1 結果 (3)

単純な分類器に切り替え,実験データを分析

→k近傍法アルゴリズムが効果的に機能することがわかった

k=1 の場合が最も良い結果を出し, Precision 0.87, Recall0.90, F1 0.84

➤ 解析の結果

Findbugsが生成した警告のデータ収集プロセスにおいて
データの重複が発生していることがわかった

結論RQ1

データリークとデータ重複を削除後,

Strawman baselineを用いるとAUCが0.52

➤ Golden Featuresが何らかの予測力を持つことを示唆

Table 4: Average Precision (Prec.), Recall, and F1 of various approaches on the original dataset by Yang et al.

Technique	Prec.	Recall	F1
Golden Features SVM	0.84	0.94	0.88
- leaked features	0.26	0.70	0.38
- data duplication	0.88	0.93	0.90
- data duplication and leaked features	0.27	0.57	0.31
+ reimplemented leaked features	0.32	0.57	0.38
Golden Features kNN (with k=10)	0.91	0.57	0.68
Golden Features kNN (with k=5)	0.86	0.72	0.78
Golden Features kNN (with k=3)	0.87	0.78	0.82
Golden Features kNN (with k=1)	0.87	0.90	0.84
Only leaked features SVM	0.79	0.94	0.83
Repeat label from training dataset	0.72	0.80	0.75

RQ2 closed-warning heuristicは警告の発見方法として適切なのか？

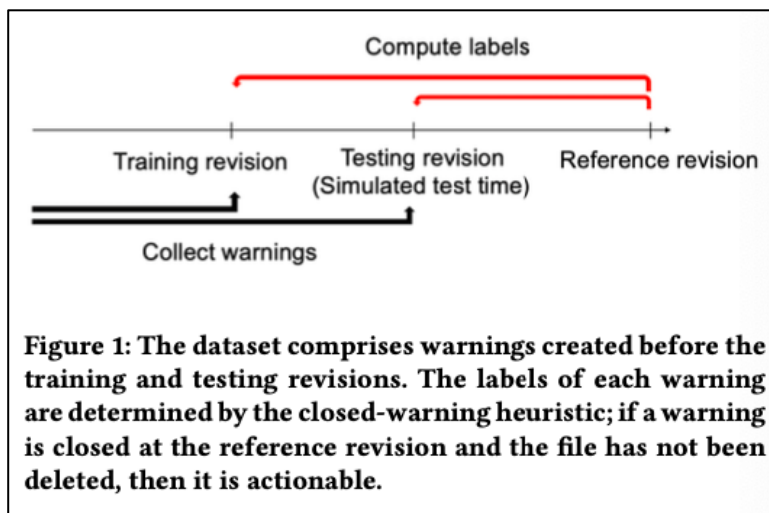
closed-warning heuristic

→警告が実用的かどうかの判定

トレーニング,テストリビジョンの前に作成された警告から構成

各警告が参照リビジョン（警告から二年間）で閉じられ,ファイルが削除されていない場合

➤ 実用的と判断



RQ2 結果（１）

- closed-warning heuristicが判断した警告は信頼できるのか？

本論文の研究員で,新たなclosed-warning 1,357件 を収集

→手動で分析

結果（１）

➤closed-warningのうち,手動によって対処可能とされたのは47%

→多くのclosed-warningが対処可能でない

また,偶然に閉じられただけのclosed-warningも存在

RQ2結果（2）

Findbugs や他のASATには,フィルタファイル が存在

→開発者が警告と認識し,追加する場所（警告の判別）

しかし,フィルタファイルにマッチしない未解決の警告は,開発者によって検査されていない可能性

➤プロジェクトの開発者によって誤報と確認されたopen-warningの割合を調査

結果（2）

平均して,警告の31%のみが開発者によって誤報であると明示

警告のうち誤報であるものは少数派

残りの警告は誤報である可能性があるが,

まだ検査されていない実用的な警告である可能性

Table 6: Number of open warnings in each project matched by their Findbugs filter file. If a warning was filtered, it indicates that the project's developers consider it a false alarm.

Project	# open warnings	# filtered	% filtered
jmeter	710	6	1%
tomcat	1624	9	1%
commons-lang	106	19	18%
flink	4934	4754	96%
hadoop	3053	269	9%
jenkins	1212	178	15%
kudu	1873	464	25%
kafka	4668	2993	64%
morphia	65	0	0%
undertow	347	113	33%
xmlgraphics-fop	949	909	96%
Average (Mean)	1666	818	31%
Average (Median)	1212	178	18%

RQ2結果（3）

- 未確認の実用的な警告をRQ1のopen-warningから削除
→再びSVMで分析

結果（3）

AUCは0.54から0.68に増加,F1も0.39から0.64に増加.

F1が0.57であるStrawman baselineを上回り,よりクリーンなデータが
Golden Features SVMの有効性を高める可能性

Dataset	Act. %	F1	AUC
Original dataset [56, 58]	39.9	0.39 (0.43)	0.54
– unconfirmed actionable warnings	40.0	0.61 (0.57)	0.66
Projects using Findbugs	38.0	0.43 (0.44)	0.62
– unconfirmed false alarms	40.0	0.41 (0.46)	0.60

RQ2結論

- ・ closed-warning heuristicは警告の発見方法として適切とは言えないかもしれない.
- ・ 実験結果から得られる知見に影響を与える可能性
- ・ closed-warning heuristicは,クローズドワーニングを実行可能な警告とし,オープンワーニングを誤報として混同
- ・ 未確認の実用的な警告を削除してよりクリーンなデータにすることで,Golden Features SVMの性能を向上させることを発見

結論

- この研究では,自動静的解析ツールから実用的な警告を検出する問題は,解決にはほど遠いことを示す
- 先行研究では,「Golden Features」の強力な性能は,微妙で検出が困難なデータリークとデータ重複の問題によって寄与されていた

所感

- Golden Featuresがすごいとあれだけ持ち上げていて,蓋を開けてみたらデータなどの不備で実際はあまり大したことがないと論文で発表されていることに驚きました.