

The Importance of Accounting for Real-World Labelling When Predicting Software Vulnerabilities

Matthieu Jimenez

University of Luxembourg, Luxembourg
matthieu.jimenez@uni.lu

Renaud Rwemalika

University of Luxembourg, Luxembourg
renaud.rwemalika@uni.lu

Mike Papadakis

University of Luxembourg, Luxembourg
michail.papadakis@uni.lu

Federica Sarro

University College London, UK
f.sarro@ucl.ac.uk

Yves Le Traon

University of Luxembourg, Luxembourg
yves.lettraon@uni.lu

Mark Harman

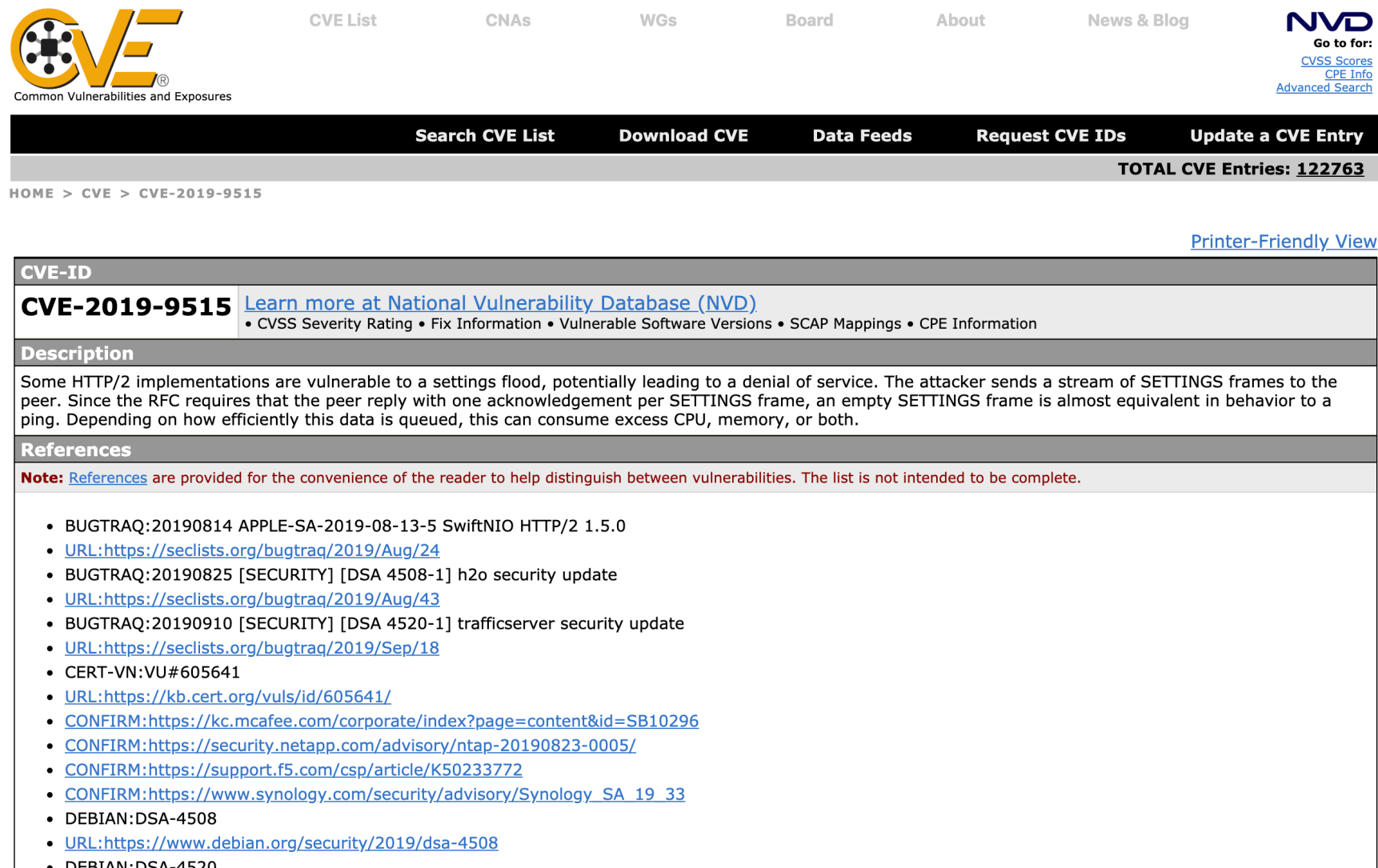
University College London and Facebook, UK
mark.harman@ucl.ac.uk

背景

- ソフトウェアシステムに含まれる脆弱性を手動で探し出すことは非現実的.
- 脆弱性を自動検出するための技術 “脆弱性予測” は広く研究されている
 - 予測には欠陥予測で使用するコードメトリクスと開発者メトリクスが使用される

セキュリティ脆弱性

- 脆弱性はMITRE社が脆弱性ごとにIDで管理している
 - CVE -Common Vulnerabilities and Exposures
- バグ報告の, CVE-IDが記述されているものを脆弱性と判定



The screenshot displays the CVE website interface. At the top, there is a navigation bar with links for CVE List, CNAs, WGs, Board, About, News & Blog, and NVD. Below this is a search bar and a table of links: Search CVE List, Download CVE, Data Feeds, Request CVE IDs, and Update a CVE Entry. The total number of CVE entries is shown as 122763. The main content area shows the details for CVE-2019-9515, including a description of a settings flood vulnerability in HTTP/2 implementations and a list of references.

CVE-ID
CVE-2019-9515 [Learn more at National Vulnerability Database \(NVD\)](#)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description
Some HTTP/2 implementations are vulnerable to a settings flood, potentially leading to a denial of service. The attacker sends a stream of SETTINGS frames to the peer. Since the RFC requires that the peer reply with one acknowledgement per SETTINGS frame, an empty SETTINGS frame is almost equivalent in behavior to a ping. Depending on how efficiently this data is queued, this can consume excess CPU, memory, or both.

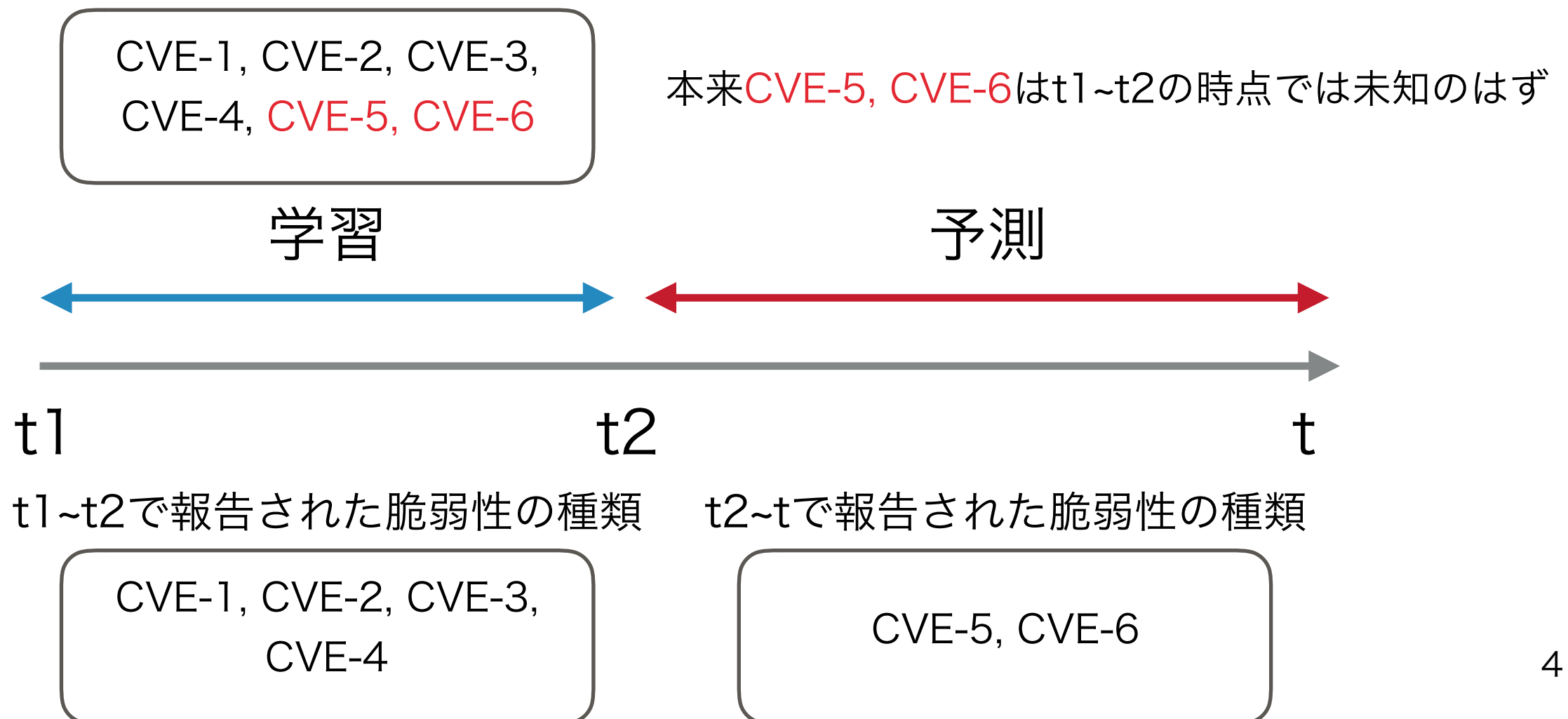
References
Note: [References](#) are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- BUGTRAQ:20190814 APPLE-SA-2019-08-13-5 SwiftNIO HTTP/2 1.5.0
- [URL:https://seclists.org/bugtraq/2019/Aug/24](https://seclists.org/bugtraq/2019/Aug/24)
- BUGTRAQ:20190825 [SECURITY] [DSA 4508-1] h2o security update
- [URL:https://seclists.org/bugtraq/2019/Aug/43](https://seclists.org/bugtraq/2019/Aug/43)
- BUGTRAQ:20190910 [SECURITY] [DSA 4520-1] trafficserver security update
- [URL:https://seclists.org/bugtraq/2019/Sep/18](https://seclists.org/bugtraq/2019/Sep/18)
- CERT-VN:VU#605641
- [URL:https://kb.cert.org/vuls/id/605641/](https://kb.cert.org/vuls/id/605641/)
- CONFIRM:https://kc.mcafee.com/corporate/index?page=content&id=SB10296
- CONFIRM:https://security.netapp.com/advisory/ntap-20190823-0005/
- CONFIRM:https://support.f5.com/csp/article/K50233772
- CONFIRM:https://www.synology.com/security/advisory/Synology_SA_19_33
- DEBIAN:DSA-4508
- [URL:https://www.debian.org/security/2019/dsa-4508](https://www.debian.org/security/2019/dsa-4508)
- DEBIAN:DSA-4520

先行研究の課題

- 先行研究では、時間的制約に関係なくラベル付けが情報が利用可能であることが暗黙的に想定されている。
 - 時間の経過とともに新しい脆弱性が明らかになることを考慮していない

先行研究で学習時に脆弱性と判定するラベル



先行研究の課題

- まだ評価されていないアプローチが存在する[38]
 - モデル学習時に脆弱性だけでなく、欠陥も脆弱性とラベルをつけ学習を行う.
 - バグは脆弱性と似たメトリクスの分布ではないかという仮説

[38] Yonghee Shin and Laurie Williams. 2013. Can traditional fault prediction models be used for vulnerability prediction? Empirical Software Engineering 18, 1 (Feb. 2013), 25–59. <https://doi.org/10.1007/s10664-011-9190-8>

研究の貢献

これまでの脆弱性予測に関する研究を包括した実証的研究を行う

1. 脆弱性ラベルの時間的制約を考慮した場合の予測モデルへの影響を調査する
2. 未評価のアプローチの調査を行う

先行研究との違い

Study	Systems	No. of Vulnerabilities	Granularity	Method	Evaluation Method	Results	Mislabelling Noise
Neuhaus et al. [28]	Mozilla	134	Component	Imports and Function calls	Cross-validation	Precision 70, Recall 45	Does not consider
Zimmermann et al. [46]	Windows Vista	66	Binary	Code Metrics & Dependencies	Cross-validation	Precision 66.7 & 60, Recall 20 & 40	Not specified
Shin et al. [37]	Mozilla Firefox, Red Hat Enterprise, Linux kernel	389	File	Code Metrics	Cross-validation & Release-based	Precision 3 - 5 & 3, Recall 87 - 90 & 79 - 85	Not specified
Shin et al. [38]	Mozilla Firefox	363*	File	Code Metrics	Cross-validation	Precision 9, Recall 91	Does not consider
Scandariato et al. [35]	20 Android apps	N/A	File	Bag Of Words	Cross-validation & Release-based	Precision 90** & 86**, Recall 77** & 77**	Does not consider and uses artificial data
Walden et al. [43]	Drupal, Moodle, PHPMyAdmin	223	File	Bag Of Words	Cross-validation	Precision 2-57, Recall 74-81	Does not consider
Zhang et al. [45]	Drupal, Moodle, PHPMyAdmin	223	File	Code Metrics & Text Features	Cross-validation	Precision 25-67, Recall 4-69	Does not consider
Jimenez et al. [22]	Linux Kernel	743	File	Imports and Function calls, Code Metrics, Bag Of Words	Cross-validation & Release-based	Precision 65-76 & 39-93, Recall 22-64 & 16-48	Does not consider
This paper ("Ideal" World) ("Real" World)	Linux Kernel, OpenSSL, Wireshark	1593	File	Imports and Function calls, Code Metrics, Bag Of Words	Release-based	Precision 44.7-83.3, Recall 33.5-76.5 Precision 19.4-44.6, Recall 1.6-26.2	Evaluates the impact of mislabelling noise

*Number of vulnerable files. **Estimated from the graphs and reported data of the paper.

リサーチクエスチョン

1. ラベル付けの時間的制約を考慮した場合、予測モデルはソフトウェアリリース間の脆弱なコンポーネントをどの程度識別するか？
2. ラベル付の時間的制約を考慮しない場合、予測モデルはソフトウェアリリース間の脆弱なコンポーネントをどの程度識別するか？
3. 学習データに脆弱性以外の欠陥を含むと精度が向上するか？（先行研究[38]のアプローチ）

データセット

Software System	Vulnerabilities	Vulnerable Components
Linux Kernel	1,202	1,508
Wireshark	265	221
OpenSSL	126	164
Total	1,593	1,898

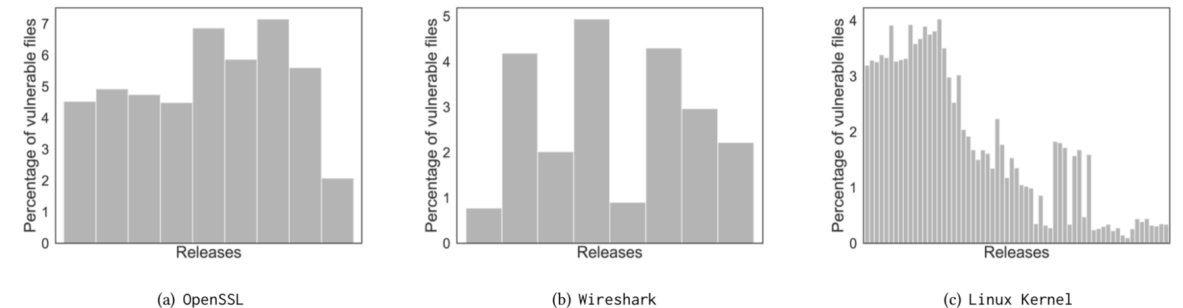


Figure 1: Ratio of vulnerable components per considered release in the three systems we study. We consider 64 versions of Linux Kernel, (versions from 2.6.12 to 4.15), 9 versions of Wireshark (all major releases) and 10 versions of OpenSSL (all major releases).

- VulData7[23]を使用してGitとCVEデータの紐付けを行う。
- RQ1 & RQ3 : Realistic Data
 - リリース時に使用可能な情報でラベル付を行う。（学習モデル構築の期間までに報告された脆弱性をラベル付け）
- RQ2 : Experimental Data
 - 考慮されている期間全てに渡って報告された，脆弱性をラベル付け

メトリクス

- メトリクス

(先行研究で利用されているメトリクスを使用)

- Imports and Function Calls

- ・ ファイルがどの関数を呼び出しているかのメトリクス（抽象構文木を使って抽出）
 - ・ 脆弱なファイルは特定の関数を呼び出している傾向があるという先行研究に基づく [28]

- Code and Process Metrics

- ・ 詳細は次のスライド

- Bag of Words

- ・ コードをトークンに分割してBag of Wordsを作成
 - ・ 次元数が非常に多いが、減らす方法は先行研究[24]を使っらしい。

[24] Igor Kononenko. 1995. On Biases in Estimating Multi-valued Attributes. In Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI). 1034–1040.

[28] Stephan Neuhaus, Thomas Zimmermann, Christian Holler, and Andreas Zeller. 2007. Predicting vulnerable software components. In CCS'07. 529. <https://doi.org/10.1145/1315245.1315311>

Code and Process Metrics

- コードメトリクス
 - コード行
 - プリプロセッサ行数
 - コードに対するコメントの割合
 - 定義された関数の数
 - 定義された変数の数
 - Cyclomatic複雑度の最小値, 平均, 最大値
 - Strict Cyclomatic複雑度の最小値, 平均, 最大値
 - Essential Cyclomatic複雑度の最小値, 平均, 最大値
 - ネストの深さの最小値, 平均, 最大値
 - 入力数の最小値, 平均, 最大値
 - 出力数の最小値, 平均, 最大値
- Code Churn(git log)
 - ファイルの履歴に追加された行, 変更された行, 削除された行
- Developer Activity Metrics
 - ファイルに影響するコミットの数
 - ファイルを変更した開発者の数
 - 現在, ファイルの開発取り組んでいる開発者の数

分類器, 学習期間, 評価指標

予備実験を実施し分類器, 学習期間, SMOTEの有無を決定
(16,760パターンから以下に決定)

- 分類器
 - ランダムフォレスト
- 学習期間
 - 最後の3つのリリースを評価データとし, 評価データ以前のデータを学習データとする
 - データの不均衡に対処するためSMOTEを使用
- 評価指標
 - Precision, Recall
 - MCC (-1 ~ 1の係数を返す. 1は完全な予測, -1は逆の予測)
 - ・ True Negativeを考慮に入れた指標. 不均衡なデータの評価に用いられる

RQ1

- ラベル付けの時間的制約を考慮した場合、予測モデルはソフトウェアリリース間の脆弱なコンポーネントをどの程度識別するか？

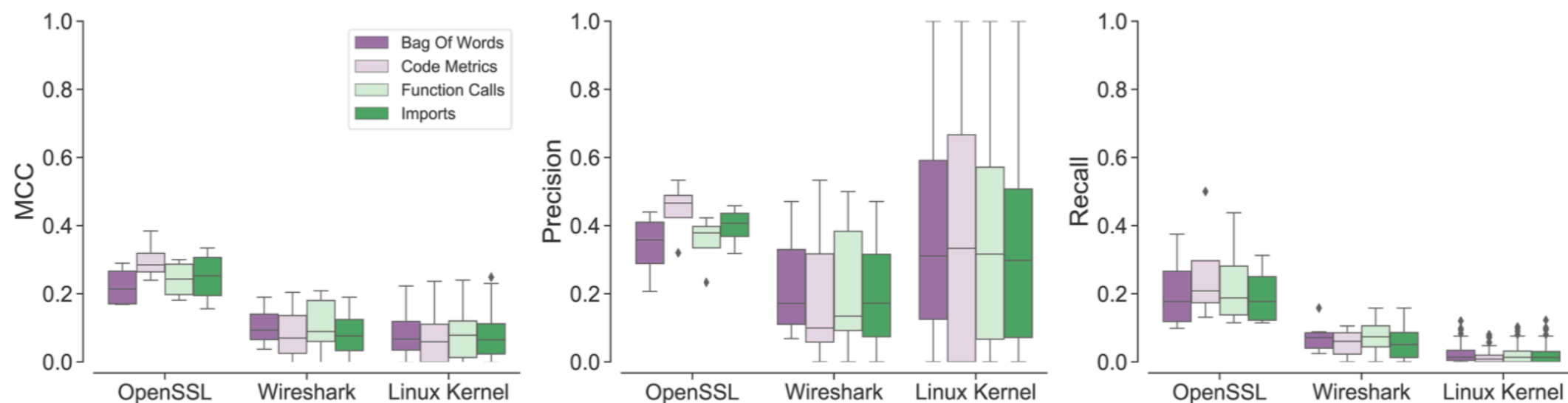


Figure 2: Prediction performance in a “real” world setting - with data labelled only with information available at model-building time(RQ1). We observe a relatively low performance for all methods. Also, all approaches show little performance differences (all yield similar MCC values).

MCCにおいてCodeMetricsが他のメトリクスに比べてわずかに優れているが殆ど変わらない。予測に使用するメトリクスに関する影響はない

全体的に精度は低い。実用に耐えうる精度は（Precision, Recall 0.7以上と言われている）

RQ2

- ラベル付の時間的制約を考慮しない場合、予測モデルはソフトウェアリリース間の脆弱なコンポーネントをどの程度識別するか？

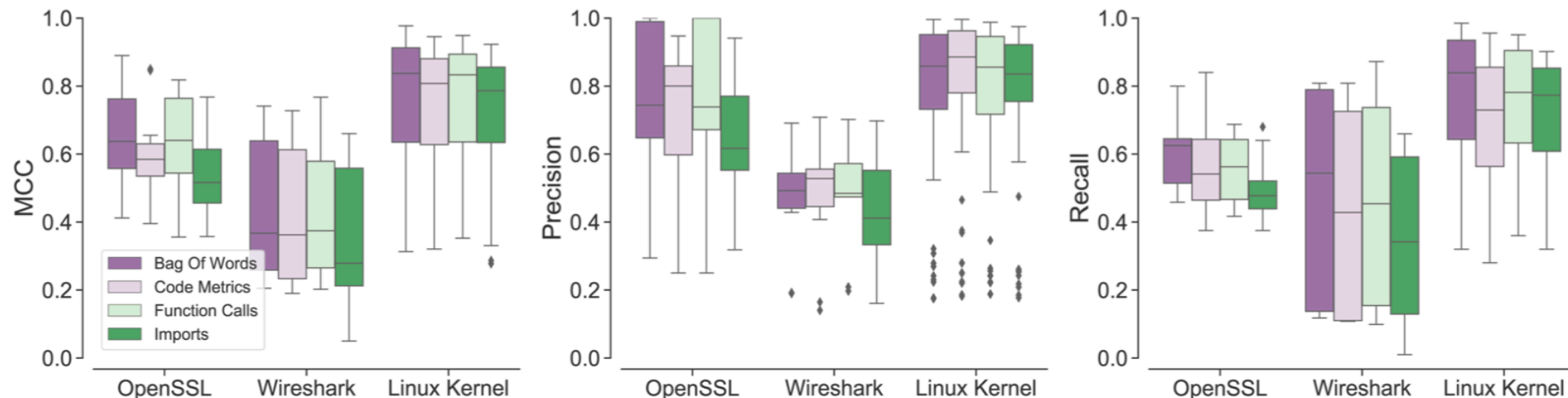


Figure 3: Prediction performance in an “ideal” world setting - with unrealistic vulnerability labelling information (RQ2). We observe a relatively good performance for all methods. *Bag of Words* and *Function Calls* methods achieve the best results, i.e., yield the highest MCC, but overall with negligible differences. There are notable performance differences from the “real” world results presented in Figure 2.

Bag of WordsとFunction CallsはMCCでは他のメトリクスよりも優れている

この結果は全体的に精度が高いが、RQ2は理想的なラベル付けに対する評価のため、RQ1の現実的なラベル付けとのギャップが大きい

RQ3

- 学習データに脆弱性以外の欠陥を含むと精度が向上するか？（先行研究[38]のアプローチ）

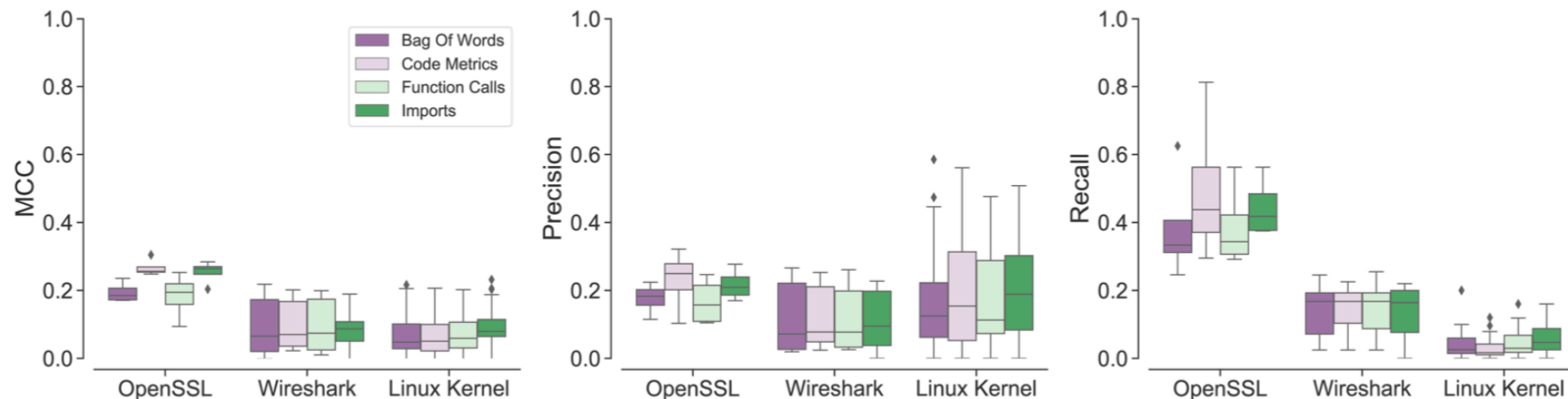


Figure 4: Performances for training set augmentation in a "real" world setting (RQ3). We observe that training set augmentation does not noticeably help achieving a reasonable performance. There are no significant performance differences from the non-augmented results that were presented in Figure 2.

学習データに欠陥データを含むことで劇的な精度の向上はおこらない。若干 Recallが向上し、Precisionが低下する（RQ1のFigure2との比較）

まとめ

- データのラベル付けに関する知見（RQ1 & RQ2）
 - 完全なラベル付けを行ったデータセットでモデルを構築すると、高いパフォーマンスを示す
 - しかし、現実的なラベル付けを行うとパフォーマンスが低下する.
- 未評価のアプローチの評価に関する知見（RQ3）
 - 脆弱性のデータに欠陥のデータも加えることでわずかに精度が向上する
- この結果は、コミュニティが脆弱性予測の評価を方法を改善して、より実用的なものにする必要があることを示す.

所感

- 研究の観点が面白い.
 - ICSEでも似た観点の論文があったけど，流行ってるんだろうか. . . .
- 予測モデルを実用的な環境で使うための評価は，他の分野でも探せばできそう