

Keyword Extraction From Specification Documents for Planning Security Mechanisms

Jeffy Jahfar Poozhithara
Computer & Software Systems
University of Washington
Bothell, USA
jeffyj@uw.edu

Hazeline U. Asuncion
Computer & Software Systems
University of Washington
Bothell, USA
hazeline@uw.edu

Brent Lagesse
Computer & Software Systems
University of Washington
Bothell, USA
lagesse@uw.edu

2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE),
pp.??-??

発表者：井町 瑠貴

背景 | セキュリティメンテナンス

ソフトウェアプロジェクトは、稼働前の環境に配備後、継続的なセキュリティテストを受ける

- 多大な時間と費用を要する
- この段階で脆弱性が発見されると、遅延が発生する可能性がある

プロアクティブ・ソフトウェア・メンテナンス・エンジニアリング^[2]

- リリースの前にセキュリティ上の欠陥を排除する
解決策を提供
- コストと潜在的な損害を削減

[2] N. Hanebutte and P. W. Oman, "Software vulnerability mitigation as a proper subset of software maintenance," Journal of Software Maintenance and Evolution: Research and Practice, vol. 17, no. 6, pp. 379–400, 2005.

既存研究 | 脆弱性予測ツール

多くの脆弱性予測ツールは、ソースコードを必要とする（＝設計段階で利用できない）

- － コードチャーン, 複雑度, 依存関係, コードカバレッジ, 生のソースコード（テキストとして） など

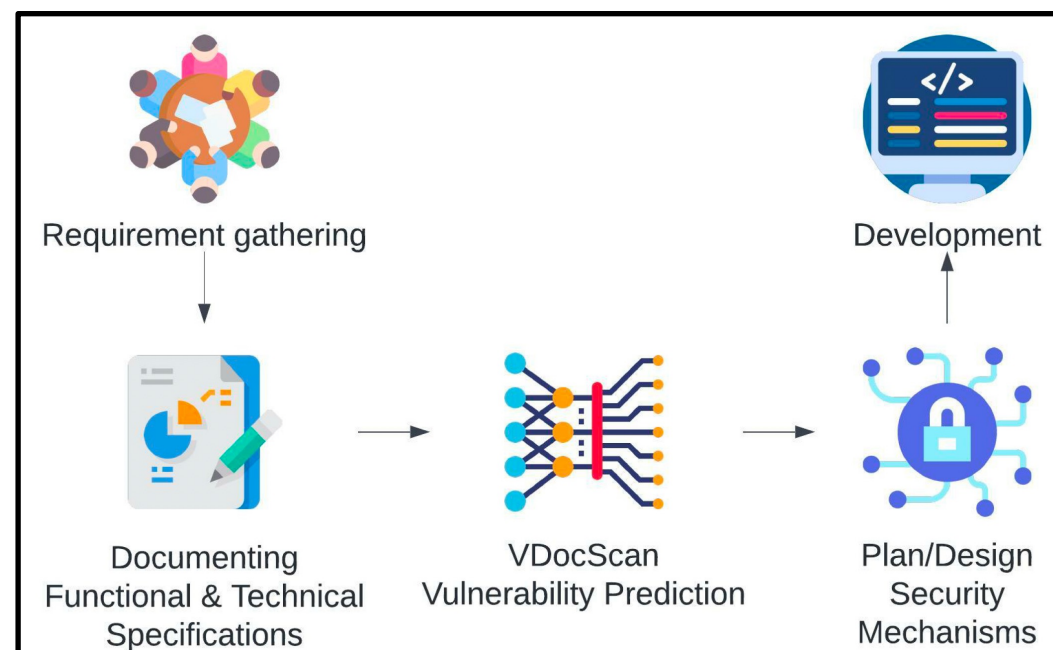
ドキュメントに依存する脆弱性予測ツールは少ない

- － 厳密には、設計段階での予測を可能にはしていない
 - コードから生成されたグラフ/フロー図を利用
 - 特定の言語, ドメイン, 脆弱性のタイプに限定されている

VDocScan

ドキュメントに含まれるキーワードと, CWE, CVEで報告された脆弱性との相関を調べる

- 入力：技術仕様書, 機能仕様書
- 出力：将来特定される可能性の高い脆弱性のリスト
- セキュリティ計画など, 予防的なソフトウェア開発やソフトウェア保守に利用できる



研究目的・アプローチ

研究目的

- リリース後に検出される脆弱性の数を少なくする

アプローチ

- 製品の実装が決まる前に作成される仕様書から脆弱性を予測するVDocScanを開発・評価

RQ | VDocScanの評価

RQ1：決定木のようなホワイトボックスモデルで脆弱性を予測できるのか

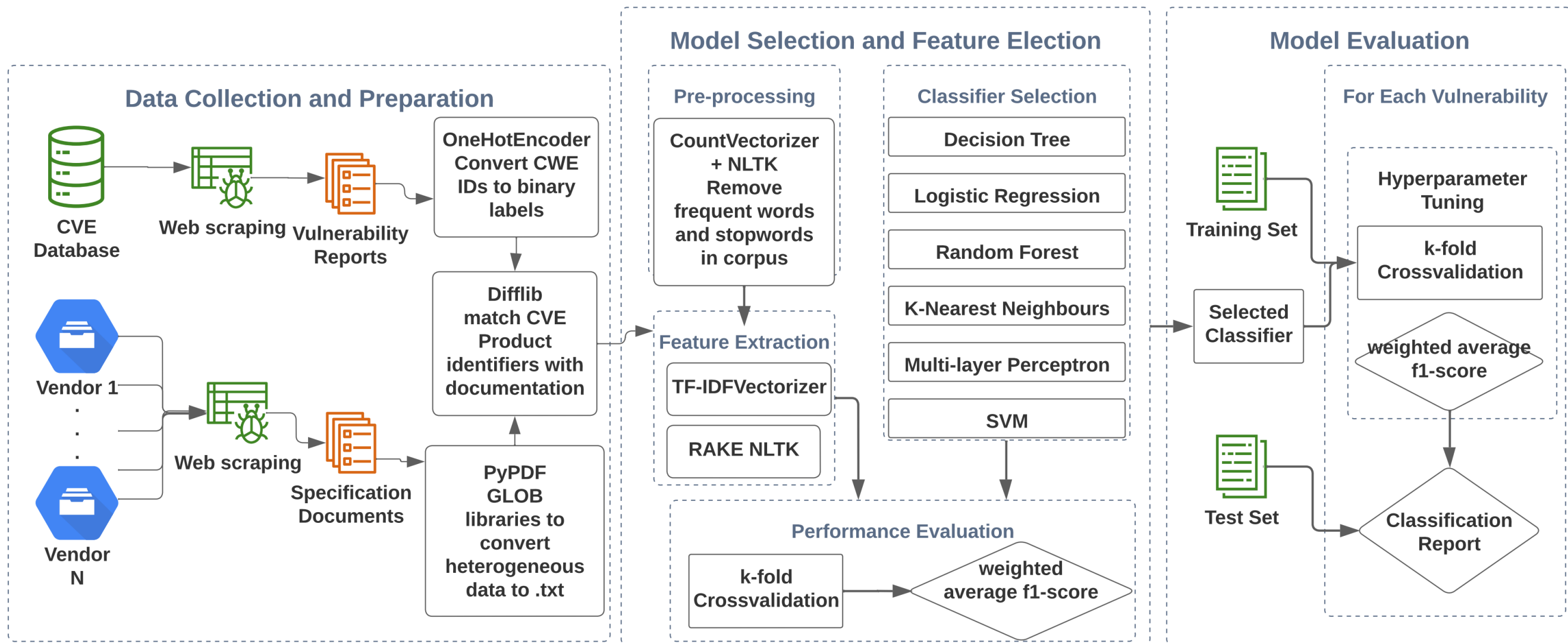
RQ2：ブラックボックスモデルは予測精度を改善できるのか

RQ3：カテゴリ分類による予測への影響

- － 分類後も十分なサンプル数があれば向上

RQ4：相関分析により，潜在的に共起している脆弱性の関係を特定することはできるのか

方法 | VDocScanの概要



方法 | データ収集

スクレイピング1（脆弱性レポート）

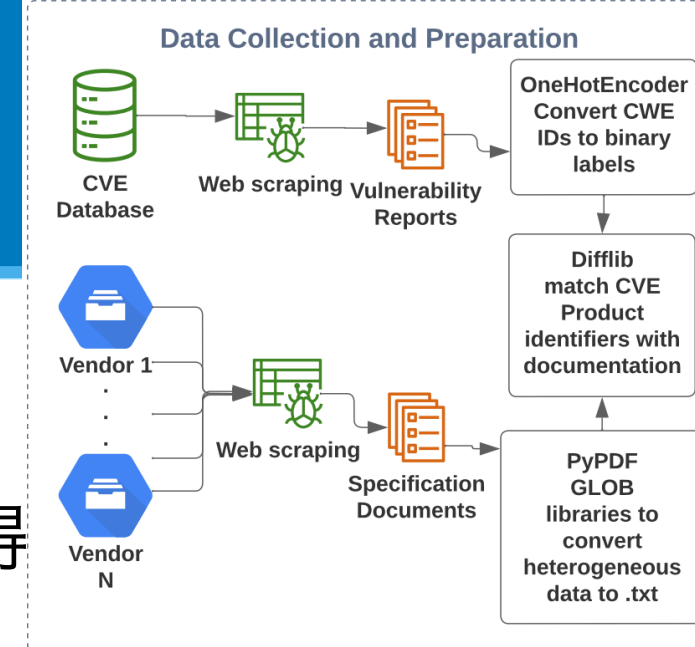
- CVEからベンダーごとに製品のメタデータを取得
 - 23,971ベンダー, 52,110製品
- さらに, 296,931件のレポートを取得

スクレイピング2（仕様書）

- 1で製品の多いベンダーから, HTML, JSON, .txt, .pdf形式ファイルを取得
 - 7,000件のユニークな製品のドキュメント

マッチング

- 製品名とファイル名は必ずしも一致しないため, テキスト類似度により, マッチング
 - 3,602製品が含まれている



方法 | 不均衡なデータへの対処

1 (脆弱性あり) ラベルは, 0 (脆弱性なし) ラベルに比べて著しく少ない

- 多いものでも50%未満

そのため, 以下の対策を講じている

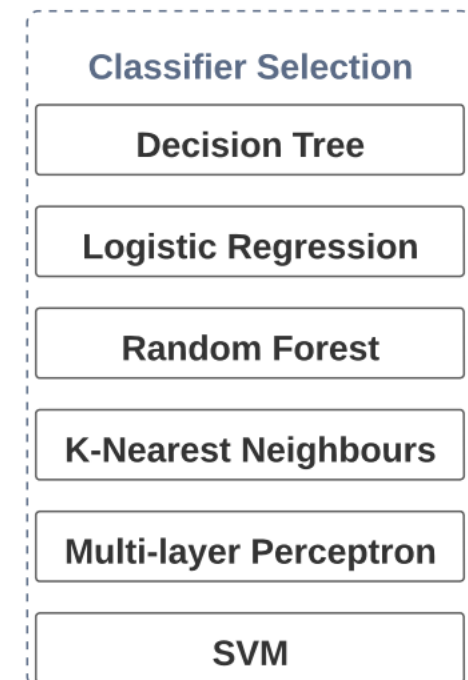
- 重点サンプリング
 - 0ラベルのダウンサンプリング
- 代替メトリクスの利用
 - 本研究では, 重み付きf値を指標として利用
- アルゴリズムのサポートを利用
 - 自動的に重みを調整するモードの利用 (SVM, RF)

方法 | モデル選択と特徴抽出

データがまばらなため、最も抽象度の高い8ラベルに付け直す（ラベルの詳細は補足資料）

モデル選択

- 右図に示されているものを用いて実験を行う



特徴の抽出

- 特徴は、ドキュメントから抽出されたキーワードで構成
 - RAKE^{[43],[44]}を用いて抽出

[43] S. J. Rose, V. L. Crow, N. O. Cramer et al., "Rapid automatic keyword extraction for information retrieval and analysis," Pacific Northwest National Lab.(PNNL), Richland, WA (United States), Tech. Rep., 2012.

[44] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic keyword extraction from individual documents," Text Mining: Applications and Theory, vol. 1, pp. 1–20, 2010.

実験1 | 決定木による分類

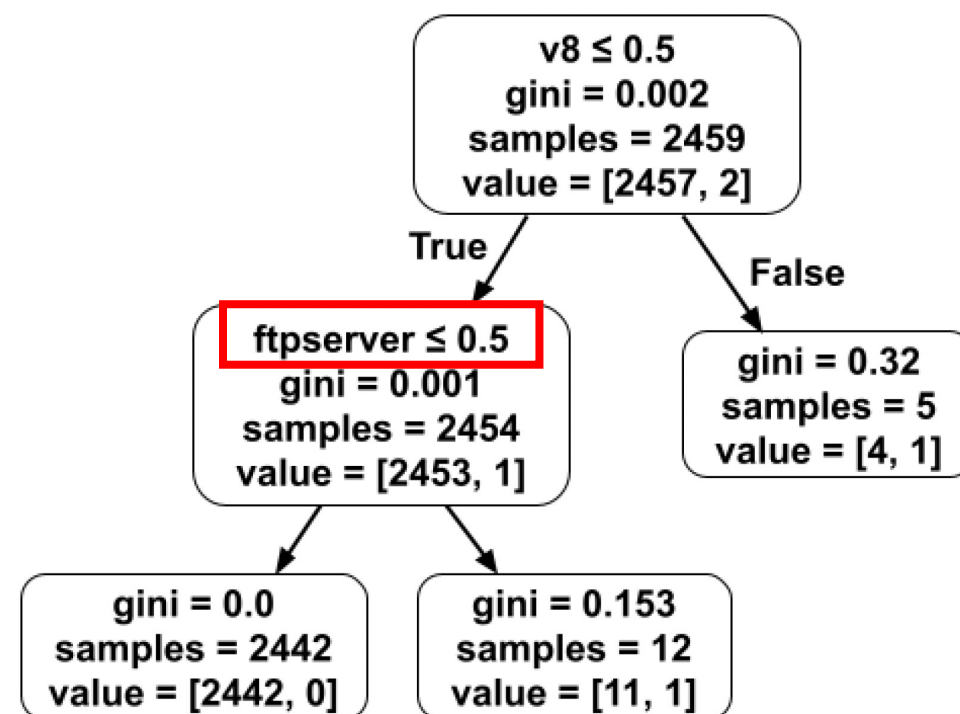
データセットの8つの脆弱性クラスを予測する
決定木モデルの性能評価

- データセットは、学習データとテストデータで4:1に分割
- 43,130の特徴量（キーワード）を利用
 - 279の頻出単語を除外済
- 5分割交差検証を行う
- ラベルの不均衡を考慮するために、重み付きf値を評価指標とする
 - ラベルの不均衡：1（脆弱性あり）ラベルが、0（脆弱性なし）ラベルに比べて著しく少ない

結果1 | 決定木による分類

決定木の重要な特徴が、脆弱性の説明と一致する

- 以下の例では、ftpserver が重要な特徴
 - ftpserver : FTPを使用してファイルの送受信を行う
- CWE-435 (相互作用エラー)
 - 正しく振る舞う複数のエンティティ間の不適切な相互作用



(a) CWE 435

結果1 | 決定木による分類

十分なサンプル数があれば、決定木を用いることで、脆弱性を予測することができる

- 1,000件以上のサンプルで学習する場合、高い水準で予測できることが示唆

CWEID	# s	P	R	f1	A
435	2	0.000	0.000	0.000	0.998
682	15	0.000	0.000	0.000	0.992
703	62	0.333	0.083	0.133	0.974
691	210	0.200	0.075	0.109	0.900
284	400	0.389	0.171	0.237	0.817
693	424	0.350	0.171	0.230	0.809
664	1166	0.553	0.780	0.647	0.630
707	1174	0.611	0.579	0.595	0.626

実験2 | ブラックボックスモデルの比較

| 以下の5種類を比較

- k-最近傍
- ロジスティック回帰
- サポートベクターマシン
- 多層パーセプトロン (MLP)
- ランダムフォレスト

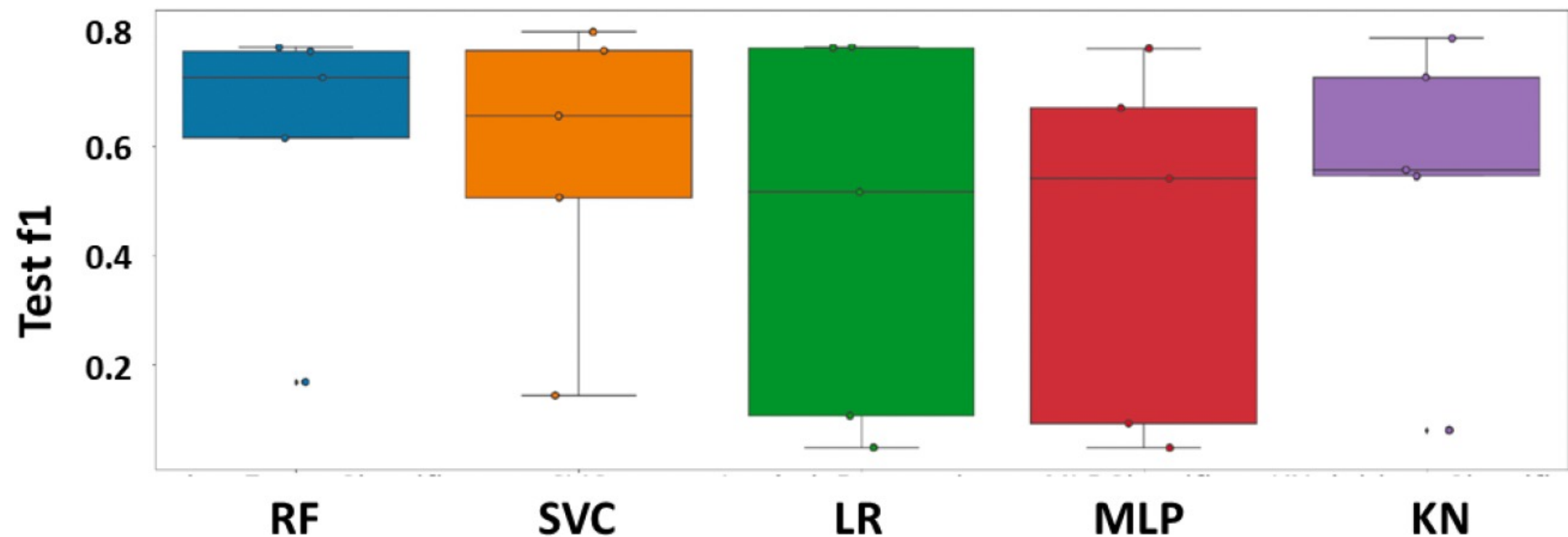
| 5分割交差検証を行い評価

- 重み付きf値により評価
- それぞれ, CWE-664用にチューニングされた最適なハイパーパラメータ値で初期化

結果2 | ブラックボックスモデルの比較

ランダムフォレストが、F値の平均と分散の両方において、最も優れている

- 以降の分析ではランダムフォレストを用いる



実験2 | ランダムフォレストによる分類

下の表のような結果が得られた上位3つの脆弱性クラスの予測性能を評価

- グリットサーチにより, 以下のパラメータを調整
 - 'bootstrap' (各ツリーを構築するためにブートストラップサンプルとデータセット全体のどちらを使用するか)
 - 'max_depth'
 - 'max_features'
 - 'min_samples_leaf'
 - 'n_estimators' (フォレスト内のツリー数)

結果2 | ランダムフォレストによる分類

下の表のような結果が得られた

- 決定木 (RQ1) と比較すると, CWE-693においては, 再現率は132.8%, f1-scoreは88.0%向上

CWEID	P	R	f1
664	0.553	0.780	0.647
707	0.611	0.579	0.595
693	0.350	0.171	0.230

決定木 (RQ1)

Vulnerability	precision	recall	f1-score
CWE-664	0.668	0.667	0.667
CWE-707	0.639	0.704	0.669
CWE-693	0.811	0.839	0.8

ランダムフォレスト (RQ2)

結果4 | 脆弱性間の相関分析

30以上の相関の高い脆弱性のペア（相関 >0.65 , p 値 <0.001 ）を特定することができた

- これらのペアが直接関係していないことは手動で確認済
- 以下は相関の高いペアの例
 - CWE-317：機密データ保護の脆弱性
 - CWE-778：ロギングに関する脆弱性

Vulnerability 1	Vulnerability 2
317 - Cleartext Storage of Sensitive Information in GUI	778 - Insufficient Logging

- このような結果から，一方の脆弱性を検出することで，相関のある脆弱性が検出される前に，修正することができる可能性がある

まとめ

仕様書に基づいて脆弱性を予測する技術である
VDocScanを開発・評価

- 8つのCWE脆弱性で評価した結果, 決定木では最大61.1%の精度と78%の再現率で脆弱性を予測
- さらに以下の行うことで精度は96%, Recallは91.1%に向上
 - 製品をOS, アプリケーション、ハードウェアのカテゴリに分類
 - ランダムフォレスト分類器などのブラックボックス・アンサンブル・モデルを使用

この結果は, VDocScanが将来的に検出される脆弱性を設計段階で事前に予測し, 必要な対処ができることを示している

所感

| Importance Sampling（重点サンプリング）に対する理解が不足していた

- 本来の確率分布に従うのではなく、重要だと考えられる領域に重みを置いてサンプリングする手法
- 具体的なアルゴリズムについての理解が不足

| RQ4の相関分析の方法が理解できなかった

- どの変数で相関を取っているのかよくわからなかった

補足

補足 | 脆弱性規格

CWE (Common Weakness Enumeration)

- 日本語では, 「共通脆弱性タイプ」
- ソフトウェアおよびハードウェアにおける脆弱性タイプをリスト化している

CVE (Common Vulnerabilities and Exposures)

- 日本語では, 「共通脆弱性識別子」
- 一般公開されている脆弱性をデータベース化したもの

補足 | 使用したラベル

以下の8種類を使用

- CWE-435 正しく振る舞う複数のエンティティ間の不適切な相互作用
- CWE-682 不適切な計算
- CWE-703 例外条件の不適切なチェックまたは処理
- CWE-691 制御フロー管理の不備
- CWE-284 不適切なアクセス制御
- CWE-693 保護機構の不具合
- CWE-664 リソースのライフタイムを通しての不適切な制御
- CWE-707 不適切な中和 (Neutralization)

補足 | データセットのサンプル

	Vendor	Product	Documentation	435	682	703	691	284	693	664	707
1	GNU	Gzip	gzip reduces the size of the named files using Lempel–Ziv coding (LZ77). Whenever possible, each file is replaced by one with the extension ‘.gz’, while keeping the same ownership ...	0	0	0	0	0	0	0	1
2	Ffmpeg	Libavcodec	The libavcodec library provides a generic encoding/decoding framework and contains multiple decoders and encoders for audio, video and subtitle streams, and several bitstream ...	0	0	0	1	0	0	1	1
3	Ffmpeg	Libswresample	The libswresample library performs highly optimized audio resampling, rematrixing and sample format conversion operations. Specifically, this library performs the following ...	0	0	1	0	0	0	0	0
4	Debian	APT	Advanced package tool, or APT, is a free-software user interface that works with core libraries to handle the installation and removal of software on Debian, and Debian-based Linux...	0	0	0	0	0	0	1	1
5	Microsoft	Xamarin	Xamarin is an open-source platform for building modern and performant applications for iOS, Android, and Windows with .NET. Xamarin is an abstraction layer that manages...	0	0	0	0	1	0	1	0