

Automated Code-centric Software Vulnerability Assessment: How Far Are We? An Empirical Study in C/C++

Anh The Nguyen
Independent Researcher,
Vietnam
anhnguyenthe.99.06.05@gmail
.com

Triet Huynh Minh Le CREST- The
Centre for Research on Engineering
Software Technologies, The University
of Adelaide Adelaide, Australia Cyber
Security Cooperative Research Centre,
Australia triet.h.le@adelaide.edu.au

M. Ali Babar CREST- The Centre for
Research on Engineering Software
Technologies, The University of Adelaide
Adelaide, Australia Cyber Security
Cooperative Research Centre, Australia
ali.babar@adelaide.edu.au

ESEIW
2024 8.3

発表者: 福井 遥陽

背景

| C/C++言語はソフトウェア工学において重要な位置を占めている.

例) オペレーティングシステム, データベース, ネットワークインフラストラクチャ

しかし, Common Vulnerabilities and Exposures(CVE)データベースにおいて, 最も多くのソフトウェア脆弱性(SV)が報告されている.

→これらは深刻なセキュリティ上の脅威となる.

従来の検出と評価の課題

SVを検出するだけではシステムの堅牢性を確保できない。

→ SVの種類ごとに重要性や緊急性が均等でなく、時間とリソースの制約からすべてを一度に修正できない。

実務では、SV評価を行い脆弱性の特徴と深刻度で優先順位をつけている

データ駆動型アプローチ

- ・ 遅延の問題：従来のSV評価の自動化研究は主にSVレポートを入力で利用してきたが、SVレポートの完成を待つ必要がある

- ・ 関数レベルの未探求：関数レベルのSV評価に適用させた研究はほとんど行われていない。

目的

1 .SV検出で実績のある機械学習モデルと深層学習モデルの性能をSV評価タスクに適用.

- 最適な分類帰と特徴抽出の選択肢を特定することを含む

2 .複数の評価タスクを単一モデルで同時に予測するマルチタスクDL戦略の有効性と効率性を調査

- 従来のマルチクラスアプローチと比較

研究の焦点

本研究では二つの主要な研究ギャップに焦点をあてた。

1. 機械学習(ML)と深層学習(DL)の性能比較の欠如：SV検出ではDLモデルがSOTA(State-of-the-Art)性能を示したが、SV評価においてどのモデルが最適か不明であった。

2. マルチタスク学習の未検証：SV評価は複数の予測タスク（例：悪用可能性，影響，深刻度）で構成されているため，各タスクに個別のモデルを構築するのは高いトレーニングコストを伴う。

データと評価メトリクス

- データセット : big-Vulデータセットから9993件の脆弱なC/C++関数を収集し, 各関数にCVSSv2のメトリクスを対応付けした.
- 評価タスク : CVSSv2のBaseグループからスコア算出に必須な以下の7つのマルチクラス予測タスクを評価対象とした.
 - 悪用可能性 : Access Vector, Access Complexity, Authentication
 - 影響 : Confidentiality, Integrity, Availability
 - 深刻度 : Severity Level (Low, Medium, Highに質的分類)
- 評価尺度 : 不均衡データセットでの評価に適したMatthews Correlation Coefficient (MCC)を主目標とし, Macro F1-Scoreを使用した.

アプローチ

本研究では6つのMLモデルと4つの特徴抽出方法の組み合わせを評価

- Light Gradient Boosting Machine (LGBM)
- Random Forest (RF)
- Extreme Gradient Boosting (XGB)
- Logistic Regression (LR)
- Support Vector Machine (SVM)
- K-Nearest Neighbor (KNN)

Bag-of-Subtokens (BoST): トークンを2～6文字のサブトークンに分割し、頻度を計測

Bag-of-Tokens (BoT): コードの構文とセマンティクスを保持したトークンの頻度

fastText: サブトークンを用いつつ、Word2Vecのようにコンテキストを考慮した分散表現を生成

Word2Vec: トークンとその近傍のトークンのコンテキストを考慮した分散表現を生成

アプローチ

次に、DLモデルは関数レベルSV検出の文献で広く使用されている3つの主要なアーキテクチャを採用

Transformerベース

- CodeBERT

グラフベース

- Gated Graph Neural Networks (GGNN)
- Graph Convolutional Networks (GCN)

シーケンスベース

- Convolutional Neural Network (CNN)
- Long Short-term Memory (LSTM)

アプローチ

- マルチタスクDLモデルへのカスタマイズ
 - 構造の変更：各DLモデルの最終層を修正し，7つのCVSS評価タスクにそれぞれ対応する7つの線形層から構成.
 - 出力定義：特定のタスクの出力taskは関数特徴量 x と学習可能な重み ω 及びバイアス b を用いて $\text{task} = \omega x + b$ で定義される.
 - 損失関数：全てのタスクのクロスエントロピー損失の平均を最小化するようにモデルを学習.

実験結果

RQ1：マルチクラス機械学習(ML) モデルと深層学習(DL) モデルは、C/C++におけるSV評価をどれだけ上手く実行できるか？

モデルカテゴリ	最良モデル/特徴	平均MCC	F1-Score	トレーニング時間
ML	LGBM + BoST	0.68	0.763	3,058秒
DL	CodeBERT	0.673	0.746	12,063秒

- ・それぞれの最良モデルの比較した結果， MLとDLは同程度の性能を発揮し， トレーニング時間はMLが1/4であった.

実験結果

RQ 2 : マルチタスクDLは、C/C++におけるSV評価を改善できるか？

項目	マルチタスクCodeBERT	マルチクラスからのMCC向上	全体的な効率（訓練時間）
平均MCC	0.734	9%	マルチクラスCodeBERTより23%削減

- ・ マルチタスク学習は、全てのDLモデルの性能を向上させ、MCCは平均 8 %～22%増加した.
- ・ マルチタスクCodeBERTは、7つのマルチクラスCodeBERTモデルの合計訓練時間（12,063秒）と比較して、総訓練時間を約23%削減し（9,235秒）、DLモデルの効率性を大幅に改善

考察

本研究はデータ駆動型アプローチが関数レベルSV評価を効率的にサポートできることを実証した。

リソース制約下での最良の選択

- ・限られたリソース下ではマルチクラスMLモデルが最適。これはより複雑なDLモデルと遜色ない性能を達成しつつ、コストを節約できるため。

最高性能追及のための最良の選択

- ・マルチタスクDLモデルが推奨される。知識共有を可能にするマルチタスク学習は、DLモデルの性能を飛躍的に向上させ、複数のモデルを個別に訓練・維持するコストも削減するため