IoT システムの欠陥に対するコードクローン検出の有効性の調査

On the Effectiveness of Clone Detection for Detecting IoT-related Buggy Clones

大野 堅太郎 * 吉田 則裕 † 朱 文青 ‡ 高田 広章 §

あらまし コードクローン検出とは，ソースコード中での類似または一致した部分を検出することで，ソースコード上に含まれる欠陥検出に対して有効性を持つ．本研究では，IoT システムの欠陥に対するコードクローン検出の有効性を調査した．

1 概要

IoT システムは複雑なため，欠陥が含まれる可能性が高い．コードクローン検出とは，ソースコード中での類似または一致した部分を検出することである．欠陥を含むソースコードを収集したデータセットとの比較により，欠陥を検出することが可能である．我々の研究グループは，IoT システムの欠陥を収集したデータセットを作成し，コードクローン検出を行うことにより，IoT システムの欠陥検出の有効性について調査を行った．本論文では，データセットの作成方法と，データセットに対してコードクローン検出を行った結果の一部を示す．

2 欠陥を含むソースコードの収集

欠陥を含むソースコードを，欠陥に対して修正される前のソースコードとして定義し，**CVE を元に収集する方法**と **GitHub のイシューを元に収集する方法**で欠陥を含むソースコードを収集した．収集したデータセットは `https://zenodo.org/record/5090430#.YSe4mI77Q2w` に記載した．

2.1 CVE を元に収集する方法

最初に，CVEList のリポジトリ内で対象キーワードを検索することにより，IoT システムにおける脆弱性に関する CVE を取得する．CVE は，一般公開されている脆弱性の識別子である．対象キーワードは SQL injection，CSRF (cross-site request forgeries)，XSS (cross-site scripting)，weak password の 4 つである．これらのキーワードは [1] より，IoT システムに対する主な攻撃分類の中で，ソースコード上に表れやすい脆弱性であるウェブ管理システムを介した攻撃分類における攻撃方法である．次に，CVEList のリポジトリでの検索で得られた対象キーワードを含む CVE を GitHub 上で検索にかける．その検索結果のコミットメッセージ，イシュー，プルリクエストなどを目視し，対象キーワードに関するソースコードを探索する．これにより，対象キーワードを含む CVE に関するソースコードを収集する．最後に，パッチファイルや Blame 機能を使用して修正前のソースコードを取得する．

2.2 GitHub のイシューを元に収集する方法

最初に，[2] からリポジトリのオーナー名，リポジトリ名，イシュー番号を取得する．[2] は IoT システムの欠陥の分類のために，IoT の欠陥に関連したイシューを

*Kentaro Ohno, 名古屋大学
†Norihriro Yoshida, 名古屋大学
‡Wenqing Zhu, 名古屋大学
§Hiroaki Takada, 名古屋大学

まとめたデータセットである．次に，REST API [3] を使用して，オーナー名，リポジトリ名，イシュー番号からプルリクエスト番号を検索する．これは，イシュー番号から直接パッチファイルとパッチファイルにより変更されたファイル群を取得することはできないためである．イシューとプルリクエストの関係は一様ではないので，すべてのプルリクエスト番号を取得することはできない．最後に，取得したオーナー名，リポジトリ名，プルリクエスト番号から GitHub の REST API [3] を使用して，パッチファイルとパッチファイルにより変更されたファイル群を取得する．そして，パッチファイルから修正される前のファイル群を取得する．

3　コードクローン検出による IoT システムの欠陥検出の調査

2 で収集したデータセットに対して，コードクローン検出をかけることにより，IoT システムの欠陥検出におけるコードクローン検出の有意性の調査を行った．ソースコード内に同じような欠陥が複数個所含まれていなければ，欠陥を検出することは不可能である．したがって，データセットからランダムに同じような欠陥を複数個所含むソースコードを選択し，コードクローン検出を行った．データセットが多言語を含み，使用するコードクローン検出ツールは多言語対応している必要があるため，NiCad [4] と CCFinderSW [5] を用いた．GitHub のイシューを元に収集したデータセットに対する NiCad を用いたコードクローン検出では，IoT システムの欠陥を検出できた割合は 0.32(関数単位，しきい値 70%，タイプ 3 の場合 [6]) であった．さらに，CVE を元に収集したデータセットに対する検出や，NiCad や CCFindeSW の設定値を変更して検出を行った．この結果から，コードクローン検出により，IoT システムの欠陥を部分的に検出できたが，半数以上の欠陥は検出できないことが分かった．

4　今後の課題

IoT システムの複雑さからコードクローン検出が難しい欠陥が存在することが分かった．そのため，IoT システムの特徴を考えることで，IoT システムの欠陥検出に有効なコードクローン検出ツールの設定値などの調査を検討している．

謝辞　本研究は JSPS 科研費 JP20K11745 の助成を受けた．

参考文献

[1] Xingbin Jiang, Michele Lora, and Sudipta Chattopadhyay. An experimental analysis of security vulnerabilities in industrial IoT devices. *ACM Transactions on Internet Technology (TOIT)*, Vol. 20, No. 2, pp. 1–24, 2020.

[2] Amir Makhshari and Ali Mesbah. IoT bugs and development challenges. In *Proc. of ICSE 2021*, pp. 460–472, 2021.

[3] GitHub. GitHub REST API - GitHub Docs. `https://docs.github.com/en/rest`. (Accessed on 08/25/2021).

[4] James R Cordy and Chanchal K Roy. The NiCad clone detector. In *Proc. of ICPC 2011*, pp. 219–220, 2011.

[5] Yuichi Semura, Norihiro Yoshida, Eunjong Choi, and Katsuro Inoue. CCFinderSW: Clone Detection Tool with Flexible Multilingual Tokenization. In *Proc. of APSEC 2017*, pp. 654–659, 2017.

[6] Katsuro Inoue. Introduction to code clone analysis. In *Code Clone Analysis*. Springer, 2021.