

深層学習を応用したバグトラッキングシステム支援ツールの開発

05001446 東京都市大学大学院 *柏原行伸[†] KASHIHARA Yukinobu
01308725 山口大学大学院 田村慶信^{††} TAMURA Yoshinobu
01702425 鳥取大学大学院 山田茂^{†††} YAMADA Shigeru

E-mail: [†]g2181424@tcu.ac.jp, ^{††}tamuray@yamaguchi-u.ac.jp, and ^{†††}yamada@tottori-u.ac.jp

1 はじめに

今日では、様々なオープンソースソフトウェア (Open Source Software, 以下 OSS と略す) が盛んに開発されている。OSS には、OSS 自体にフォールトが存在すると、OSS に対して攻撃が行われてしまう可能性があり、脆弱性が大きくなるという問題がある。この問題を管理するために、バグトラッキングシステム (Bug Tracking System, 以下 BTS と略す) が活用されている。修正が必要と判断されたフォールトは、修正が行われ、修正した OSS を公開する。一方、BTS には、修正が行われていない未修正のフォールトも存在する。本研究では、BTS が正しく管理されているのか評価するために、未修正のフォールト割合の増減傾向を推定する。また、この推定を容易に行えるアプリケーションを開発する。BTS の利用者は、このアプリケーションを通し、未修正フォールトの増減傾向がわかることで、BTS におけるバグ管理が正しく行われているかについての評価が可能となる。

2 先行研究と本研究の位置付け

OSS については、様々な観点から研究が行われている。例えば、OSS 開発プロジェクトをお金の価値に定量化する EVM の観点からプロジェクトの進捗状況を評価する研究や、OSS 開発プロジェクトの投入開発工数の予測などといった研究が存在する [1] [2]。このように、OSS に対する様々な研究が行われている。本研究では、BTS 上の大量のフォールトデータを扱い、脆弱性に影響を及ぼすフォールトである未修正フォールトの割合の増減傾向を推定する。その際、未修正フォールトを独自に定義する。

3 未修正フォールト

バグが報告された OSS は BTS 上で様々な処理を受け、様々な状態で登録されている。その状態にはバグの処理が終了し、解決済み状態のもの、処理が途中のもの、および終了していない未解決状態のものがある。また、解決済み状態となるバグが受ける処理には、修

正が行われる、バグではないと判断される、他のバグと同じ内容と判断される、などといったものが存在する。修正が行われていない処理は、処理の判断が誤りであり、解決済み状態とされている可能性を含む。そこで本研究では、バグの状態が解決済み状態とされているが修正の処理がされていないフォールトを未修正フォールトと定義する。

4 研究方法

本研究では、Red Hat Bugzilla の BTS から Tomcat 製品のフォールトデータを入手する。このデータに対し、分析を行う際、深層学習を用いて分析を行う。未修正フォールトの割合の増減傾向を推定するため、深層学習における目的変数を未修正フォールトの割合の変化率とする。この変化率は選択したデータの範囲に応じて変化する一時的な脆弱性に影響を及ぼすフォールトの割合から算出している。使用するアルゴリズムは順伝搬ニューラルネットワークとする。説明変数は製品名、バージョン、公開日時、更新日時、重要度、OS の 6 つである。フォールトデータのうち 9000 件を学習データ、999 件をテストデータとした。

5 深層学習の考察とアプリケーションの実装

5.1 区間ごとの推定値と実データ

BTS 上の Tomcat 製品のフォールトデータを用いて、未修正フォールトの割合の変化率について推定を行った結果を図 1 に示す。横軸がテストデータ数、縦軸が未修正フォールトの割合の変化率である。赤色が変化率の実データ、青色が変化率の推定結果である。推定した変化率はデータの選択範囲に応じて変化する未修正フォールトの割合から求めた変化率であるため、一時的な変化率を表す。そこで、区間ごとに一時的な変化率の累計を求め、その値を各区間における変化率の推定値とする。その各区間における変化率の推定値をまとめたものが表 1 である。

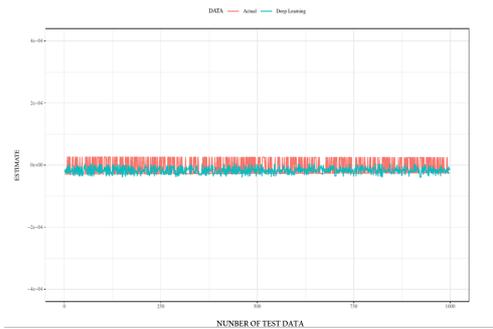


図 1 未修正フォルトの割合の変化率の推定結果.

表 1 区間ごとの未修正フォルトの割合の変化率の推定結果と実データ.

区間	データ	推定したフォルトの割合の変化率	実データ (フォルトの割合)
1	1~250	-4.06×10^{-3}	5.24×10^{-1}
2	251~500	-4.28×10^{-3}	5.20×10^{-1}
3	501~750	-2.15×10^{-3}	5.16×10^{-1}
4	751~999	-2.69×10^{-3}	5.12×10^{-1}

5.2 推定値と実データとの比較

表 1 より, 実データである未修正フォルトの割合は各区間ごとに減少していることから未修正フォルトの割合は減少傾向にあることが分かる. また, 表 1 の推定した各区間における未修正フォルト割合の変化率は全区間において負の値を示すことから, 全区間で未修正フォルトの割合が減少傾向であることを意味している. 以上より実データと推定結果が示す傾向は一致している.

5.3 アプリケーションの実装

アプリケーションはフレームワーク NW.js を用いて実装した. 図 2 は, 未修正フォルトの割合を区間ごとに推定したものである. この推定に用いるデータは図 1, 表 1 と同様のデータを用いた. 図 2 の推定結果から区間ごとに未修正フォルトの割合は減少傾向にあることがわかる.

6 今後の課題

本研究より, Red Hat Bugzilla の BTS における Tomcat 製品のフォルトデータから推定した未修正フォー

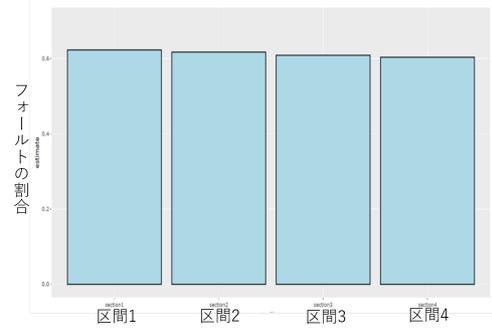


図 2 アプリケーション実行画面 (区間ごとの未修正フォルトの割合の推定量).

ルトの割合の増減傾向は実データと一致することが確認できた. 未修正フォルトは減少傾向であることが分かり, バグ管理について信頼できるという評価ができる. しかし, 図 2 における各区間ごとの未修正フォルトの割合の推定は, 実データである表 1 の未修正フォルトの割合と比較すると精度が十分でないといえる. 今後は推定精度を向上させる必要がある.

謝辞

本研究の一部は, JSPS 科研費基盤研究 (C) (課題番号 20K11799) の援助を受けたことを付記する.

参考文献

- [1] Y. Tamura, K. Tada, and S. Yamada, "Statistical EVM analysis based on deep learning for open source fault big data," Proceeding of the IEEE International Conference on Reliability, Infocom Technology and Optimization, Amity University, Uttar Pradesh, Noida, India, pp. 1-5 (Virtual mode), June 4-5, 2020.
- [2] Y. Tamura, K. Sugisaki and S. Ymada, "Jump diffusion process model considering component dependency in open source project for development effort management," Proceeding of the IEEE International Conference on Industrial Engineering Management, Singapore, December 14-17, 2020, CD-ROM (Reliability and Maintenance Engineering 2).