

深層学習を用いた組込み OSS の移植可能性評価に関する一考察

05001443 東京都市大学大学院 * 上木僚太 † UEKI Ryota
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: †g2181407@tcu.ac.jp, ††tamura@yamaguchi-u.ac.jp, ††† yamada@tottori-u.ac.jp

1. 研究の背景および目的

現代社会では、iPhone や TV などの製品によって生活の質が向上し、発展を続けている。同時に、多種多様に変化するニーズに適応するために、ワイヤレス通信機能など、ある製品に未導入であったソフトウェアを搭載する事例が増えている。これを実現可能としている技術がオープンソースソフトウェア（以下、OSS と略す）である。OSS はソースコードが公開されている上に、自由にソースコードを変更することができる。しかし、第三者が記述したソースコードに追記することは比較的難しく、自ら修正を行う場合、プログラムの不具合による脆弱性も大きくなりやすい。加えて、品質管理方法にも課題を残しており、OSS には明確なテスト工程が存在していないという観点から、総合テスト工程を対象とした一般的なソフトウェア信頼性評価法は適切ではない。また、組込み OSS の信頼性評価に関する研究も大方行われていないことが現状である [1]。これは、組込み OSS について、ハードウェアに依存するコンポーネントが含まれていることが原因とされている [2]。このため、近年 OSS の信頼性評価や組込み OSS を別のシステムに移植した際、一定期間経っても不具合が発生せずに問題なく稼働する可能性（以下、移植可能性と略す）は評価尺度として有益である [3]。

本研究では、OSS の移植可能性について評価し、考察を行う。

2. 先行研究と本研究の位置付け

先行研究では、ソフトウェアの開発から運用・保守に及ぶ工程において信頼性を評価する試みが行われていないことを背景に、ソフトウェアコンポーネントと組込み OSS の特性を考慮したハザードレートモデルを提示し、その適用可能性について議論されている [4]。また、別の研究では、フォールト発見率に着目したベイジアンネットワークに基づくモデルも提案されている。これらは、バグトラッキングシステム（以下、BTS と略す）上のデータに対して、OSS の信頼性を定量的に評価することで、移植可能性を評価していた [3]。しかし、評価を行う上で、BTS 上のその他のフォールト要因は含まれていない。

本研究では、移植可能性について、深層学習に基づき BTS に登録されている Assignee などの複数のフォールト要因により評価し、考察を行う。

3. 研究方法

3.1. 使用データ

本研究で使用する BTS を「Red Hat Bugzilla」とし、約 18000 件に及ぶ Red Hat Enterprise Linux のデータを入手した [5]。なお、これらのデータは、いずれも現在サポート中である Red Hat Enterprise Linux 6, Red Hat Enterprise Linux 7, Red Hat Enterprise Linux 8 である。入手したデータの概要を表 1 に示す。

3.2. 使用モデル

本研究では、Deep Neural Networks（以下、DNN と略す）に基づく学習アルゴリズムを用いる。DNN とは、4 層以上の

表 1: 変数一覧.

使用変数	定義
Product	Linux のバージョン
Component	コンポーネント
Assignee	BTS への登録者
Status	バグの状況
Resolution	フォールトの詳細状態
Summary	バグの内容
Changed	変更日
目的変数	
Freq (Product) × Freq (Component)	独立事象に基づく 乗法定理

Error transition

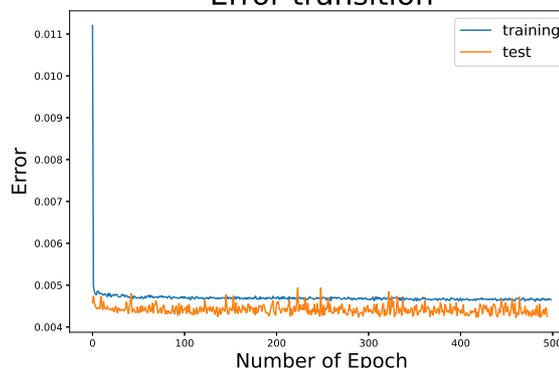


図 1: 損失関数の推定結果.

ニューラルネットワークであり、Keras に用意されているモデルが 2 つ存在する。本研究においては、そのうちの 1 つである Sequential モデルを適用する。

入力層には約 18000 件に及ぶ表 1 の Product と Component を除外した 5 つの説明変数を入力し、出力層には目的変数の数値を出力させることで、組込み OSS の移植可能性を評価する。深層学習の構造を後述する 3.4 節にて示す。

3.3. 過学習対策

DNN において発生しやすい問題は、訓練データに対してのみ過剰に適合してしまう「過学習」である。過学習が発生した場合、未知データに対して予測を行うことが困難になり、現場での使用は不可能になる。これを回避するために、本研究では、過学習に対して有効である「ドロップアウト」と「L2 正則化」の処理を行った。この結果、損失関数の遷移をグラフ化した図 1 を得た。

3.4. パラメータ調整

本研究では、深層学習を取り扱う上で必要となるハイパーパラメータの調整を、バイズ最適化のライブラリである「hyper opt」を用いて行った。hyper opt とは、パラメータ調整中の記録を保存・参照しながら、次に探索すべきパラメータをベ

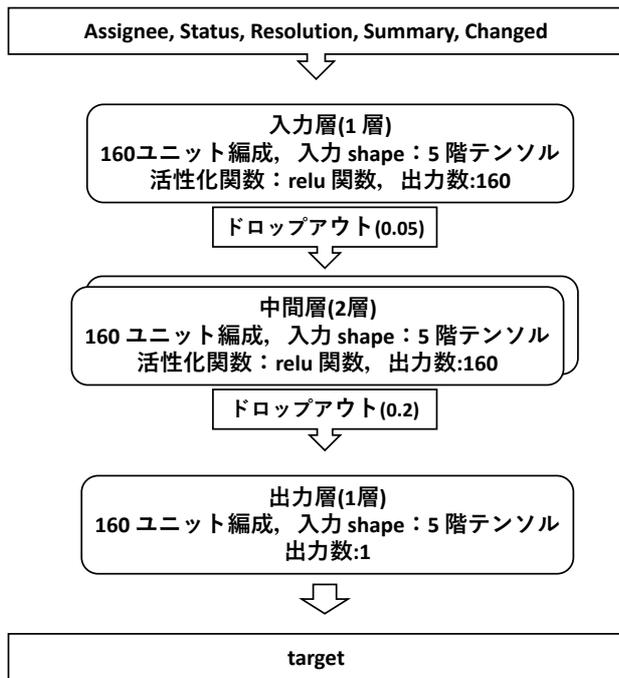


図 2: DNN の構造.

表 2: hold-out 法による本モデルの評価結果.

train データの割合	MAE
50%	0.00443
60%	0.00429
70%	0.00420
80%	0.00422
90%	0.00424

イズ探索に基づいて探索を行う方法である。このため、ランダムサーチよりも探索効率が優れていると言われている。本研究で取り扱う、パラメータ調整後の深層学習の構造を図 2 に示す。

4. 組込み OSS の移植性評価

OSS プロジェクトにおいてテスト工程がないことを背景に、ソフトウェアの ISO/IEC における品質管理だけでは、近年の OSS に対応できない状況がある。また、発生したバグを BTS に登録していることから、プログラムに潜む未発見バグが拾いきれていない傾向にある。このことを考慮すると組込み OSS を移植する際、BTS のデータに加えて脆弱性についても考慮する必要があることが分かる。ここで、脆弱性の原因であるプログラムのバグ等は、BTS に登録されているほど少なくなることから、本研究ではバグ要素を示す Product や Component に注目する。さらに、これら 2 変数の出現頻度の確率から乗法定理に基づき、予測出力値が高ければ脆弱性が低くなる傾向にある、移植可能性が高い傾向にあるとして、OSS の移植可能性を評価する。表 1 に示すデータから、深層学習に基づいて推定した結果を図 3 に示す。また、hold-out 法を用いた Mean-Absoluted-Error による本モデルの評価結果を表 2 に示す。

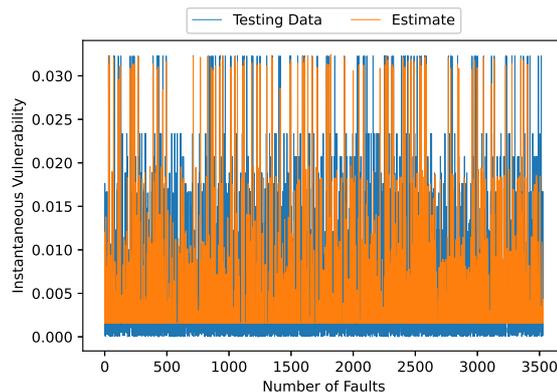


図 3: Product および Component に基づくフォールト出現頻度の推定結果.

5. まとめと今後の課題

本研究では、深層学習を用いて BTS 上のデータから移植可能性を評価した。本研究における提案手法により、今まで組込み OSS の導入を躊躇していたプロジェクトにおいて、移植可能性の判断が容易になる。

今後の課題として、移植可能性を評価するにあたり、各 Red Hat Enterprise Linux のバージョンごとにおいて初期段階のバグ数が異なるため、重みづけ等を行う必要があると考える。また、移植可能性を評価するにあたり、残存バグ数を推定した後、変数「バグ修正量 / 残存バグ数」を作成し、フォールト修正率に着目する必要があると考える。また、精度面に関しては、説明変数である Summary について TREE TAGGER 等を用いた形態素解析などの適用による新しい特徴量の生成や、LightGBM 等の決定木を作成し、本研究の DNN とのアンサンブルを行うべきであると考えられる。

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] 吉田祐貴, 田村慶信, 山田茂, “組込み OSS に対する一般化ハザードモデルに基づく開発コスト削減量に関する比較”, 数理解析研究所講究録, Vol. 1734, pp. 280–287, 2011.
- [2] Y. Tamura and S. Yamada, *Software Rrliability assessment and optimal version-upgrade problem for open source software*, Proc. of the 2007 IEEE Intern. Conf. on Systems, Man, and Cybernetics, Montreal, Canada, October 7-10, 2007, pp. 1333-1338.
- [3] 田村慶信, 竹原英秀, 山田茂, “組込みオープンソースソフトウェアに対する移植性評価法に関する一考察”, 電子情報通信学会誌, Vol. 109, no. 67, pp. 13-17, 2009.
- [4] 吉田 祐貴, 田村慶信, 山田 茂, “組込み OSS に対する一般化ハザードレートモデルに基づく 開発コスト削減量に関する比較”, 数理解析研究所講究録, Vol. 173, pp. 280-287, 2011.
- [5] “Red Hat Bugzilla – Search for bugs”, Red Hat Bugzilla, <https://bugzilla.redhat.com/query.cgi?format=advanced>, (参照 2020-06-12).