

発生頻度の少ないコーディング規約違反データ統合による 検出精度向上への試み

亀岡 令[†]
和歌山大学[†]

伊原 彰紀[‡]
和歌山大学[‡]

南 雄太[§]
和歌山大学[§]

大森 楓木[¶]
和歌山大学[¶]

1 はじめに

ソフトウェアを共同開発するプロジェクトでは、他者が実装したソースコードでも保守を容易にするために、コーディング規約を遵守する可読性の高いソースコードの実装が期待される。具体的には、Python の PEP8 や Java の Code Conventions for the Java Programming Language など、各言語が標準的な規約を公開している。各プロジェクトは、保守性や可読性を高めるためにインデントや括弧などの記法 **TODO: [記法だけだっけ? エラー, プログラムの特徴量 (複雑度) なども含んでいなかった?]** を標準的な規約に加えて個々に定めることもある。

多くのプロジェクトでは静的解析ツールを用いることで規約を違反するソースコードの検出を自動化している。一方で、開発者は **TODO: [XX のため]** 静的解析ツールが検出する膨大な違反の中から、修正すべき違反の選択が余儀なくされる。従来研究では、機械学習を用いて優先的に修正すべき規約違反を推薦する手法を提案している [?]. しかし、

規約違反の種類ごとの発生頻度が低く、修正率 **TODO: [発生率?]** が低い規約は、推薦精度に影響する。特に、新規プロジェクトでは過去に発生した違反修正履歴を持たないため、規約違反の推薦は容易でない。

本研究では、**TODO: [何のために?]** 連合学習を用いた **TODO: [何を? 特にクロスプロジェクトのことを書くべきでは?]** 予測するモデルを提案する。連合学習は、学習データから得られたモデルに加えて、別の学習データから得られたモデルも利用してモデルを構築するため、それぞれの学習データセットの特徴を考慮することができる。**TODO: [だから何?]** 本研究では、プログラムの編集履歴を特徴量とし、違反が修正された編集の特徴をもとにし、連合学習によって構築されたモデルを利用することで、**TODO: [従来研究よりどのように良い方法であるのかを一言で書くべきでは?]** 規約違反が修正させるかを予測する。

2 分析

2.1 データセット

本研究では、GitHub から Python で実装されており Pylint を用いて開発している transitions, schematics, schema_salad, python-bugzilla, python-cloudant, pyscard, pynput, OWSLib, howdoi, hickle の 10 リポジトリを対象とする。これらのリポジトリからそれぞれ 1,000 日間のコミット (**TODO: [中央値で**

Toward improving detection accuracy by integrating occasional coding convention violations datasets

[†] Ryo Kameoka, Wakayama University

[‡] Akinori Ihara, Wakayama University

[§] Yuta Minami, Wakayama University

[¶] Fuki Omori, Wakayama University

XX コミット分) を分析対象とする。

2.2 分析手法

規約違反が修正されるかの予測には、**TODO: [従来研究で使用された??]** 変更行数や複雑度などソースコードや編集についての 43 個の特徴量を抽出し、FedAvg アルゴリズムを用いた連合学習を行う [?]. 目的変数は、ある **TODO: [「ある」とは?]** コミットにおいて規約違反が修正有無の 2 クラス分類とし、説明変数には規約違反の種類やソースコードの複雑度などの 44 個 **TODO: [上の 43 ではない?]** のパラメータを用いる。 **TODO: [規約違反検出数に対して修正された違反数が極端に少ないため?]** 学習データとテストデータは、コミットの古い順から 7 割を学習データとし、残り 3 割を予測データとすることで、時系列を考慮した予測を行う。

図 1 は、FedAvg を用いた連合学習の概要を示し、個別 **TODO: [この研究ではプロジェクトごとってこと?]** の深層学習によって得られたモデルを集約し平均を取り、再度学習する際に出力されたモデルと、先ほど **TODO: [先ほどとは?]** 得られたモデルとの平均 **TODO: [中央値ではなく? 普通は平均を使うものなの?]** とする。モデルの構築、得られたモデルの集約、再度学習を繰り返すことで、より精度の高いモデルを得る手法である。実装には、オープンソースフレームワークの OpenFL を用いる。

本論文で使用する FedAvg において、深層学習のレイヤーは 3 層で、1, 2 層目は ReLU 層であり、3 層目にシグモイド関数を用いた出力層となっている。最適化アルゴリズムには Adam を利用し、学習率は 0.01 に設定した。

2.3 結果

分析の結果から

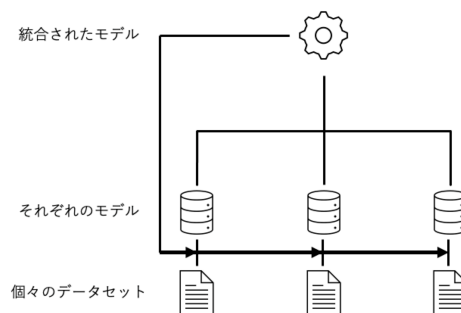


図 1 連合学習の概略図

表 1 分析結果

	適合率	再現率	F1 値
連合学習モデル	0.03	0.52	0.06
深層学習モデル	0.03	0.32	0.05

3 おわりに

本研究では、静的解析ツールによって検出された規約違反が予測地点において修正されるかを、連合学習によって精度向上を試みた。今後の方針として、データセットの拡張、違反頻度の低い規約違反への重み付けなどを用いて予測精度の高いモデルの構築を目指す。

謝辞

～～～

参考文献

- [1] 福地ユキ, 伊原彰紀, 山本豪志朗, 橋谷直樹, オブジェクトの動作に基づく Scratch 作品の直感的検索手法, **TODO: [会議名]**, 2021
- [2] J. Moreno-Le'on, G. Robles, and M. Rom'an-Gonz'alez, "Dr. scratch: Automatic analysis of scratch projects to as-

sess and foster computational thinking,”
RED.Revista de Educaci'on a Distancia,
vol.15, no.46, pp.1-23, 2015.

- [3] E. Aivaloglou, F. Hermans, J. Moreno-Le'on, Gr. Robles, “A dataset of scratch programs: scraped, shaped and scored,”
Proceeding of the International Conference on Mining Software Repositories (MSR'17), pp.511-514, 2017.