Fault-proneモジュール判別におけるサンプリング法適用の効果

亀井靖高[†] 松本真佑[†] 柿元健[†]
門田暁人[†] 松本健一[†]

Fault-proneモジュール（バグを含みやすいモジュール）判別モデルの性能向上を目的として，サンプリング法（オーバ/アンダサンプリング）適用の効果を実験的に評価する．サンプリング法は，2群の判別モデル構築に用いるフィットデータに対する前処理であり，2群のケース数の偏りを解消することにより，少数派の群の判別精度の向上が期待される．ただし，従来，ソフトウェア工学分野への適用事例は報告されていない．実験では，日本のある企業で開発されたレガシーソフトウェアから得られた2つのモジュール群（バグモジュール含有率5.67%，および13.16%）を題材として，4種類のサンプリング法（ROS，SMOTE，RUS，TLUS）適用の効果を，4種類の判別モデル（線形判別分析，ロジスティック回帰分析，ニューラルネット，分類木）について評価した．実験の結果，分類木以外の判別モデルに対してはサンプリング法の効果があること（F1値の向上），および，TLUSは他の3つのサンプリング法よりも効果が小さいことが分かった．また，分類木以外の判別モデルにTLUS以外のサンプリング法を適用した場合，F1値の向上は最小0.078，最大0.303，平均0.146であった．

Applying Sampling Methods to Fault-prone Module Detection

Yasutaka Kamei,[†] Shinsuke Matsumoto,[†] Takeshi Kakimoto,[†]
Akito Monden[†] and Ken-ichi Matsumoto[†]

To improve the prediction performance of fault-proneness models, this paper experimentally evaluates the effect of over and under sampling methods, which are preprocessing procedures for a fit dataset. The sampling methods are expected to improve the prediction performance when the fit dataset is imbalanced, i.e. there exists a large bias between the number of fault-prone modules and the number of non-fault-prone modules. There is, however, no research that reports the effects of applying sampling methods to fault-proneness models. In the experiment, we evaluated the effects of four sampling methods (ROS, SMOTE, RUS, TLUS) applied to four fault-proneness models (linear discriminant analysis, logistic regression analysis, neural network and classification tree) by using two module sets (ratios of fault-prone modules are 5.67% and 13.16%) derived from large-scale legacy software that has been developed and maintained by a Japanese software company. The results showed that all four sampling methods improved the prediction performance of every fault-proneness model except the classification tree. However, the improvement by TLUS was less than that of the other sampling methods. The improvements in F1-Values were 0.079 at minimum, 0.303 at maximum and 0.146 at the mean when three sampling methods (ROS, SMOTE and RUS) were applied to three fault-proneness models (linear, logistic and neural network).

1. はじめに

ソフトウェアテストおよび保守工程において，fault-proneモジュール（バグを含みやすいモジュール）[16]を特定することは，テストの効率化，および，信頼性を向上させるうえで重要である[13]．そのために，従来，モジュールから計測されたメトリクス（プログラム行数，サイクロマティック数，変更行数など）を説明変数とし，モジュールのバグの有無を目的変数とするfault-proneモジュール判別モデルが多数提案されており，線形判別分析，ロジスティック回帰分析，ニューラルネットなど様々な種類のモデルが存在する[4),8),14),17)]．これらの判別モデルの構築は，過去のソフトウェア開発における各モジュールのメトリクス値とバグの有無を記したデータセット（以降，フィットデータ）を用い

† 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology

て行われる．なお本論文では，バグという用語を fault（モジュール中に存在する正しくない処理）の意味で用いる．

しかし，fault-prone モジュール判別モデルを開発・保守の現場で利用する場合に，モデル構築に適したフィットデータが必ずしも用意できるとは限らない．特に，フィットデータに含まれるバグありモジュールの個数が（バグなしモジュールと比べて）著しく少ない場合がしばしば見られ，性能の良いモデル構築の妨げとなる[9]．たとえば，Khoshgoftaar らが用いたデータセットでは，全モジュールに対する（保守工程で発見された）バグありモジュールの割合が約 6%であった[10]．また，NASA IV&V Facility Metrics Data Program[15] が公開しているデータセットの場合，11 件中 8 件のデータセットはバグありモジュールの割合が 15%未満であった．

従来，偏りのあるデータセットを対象として，fault-prone モジュール判別モデルを拡張する試みが行われている[10]．通常の判別モデルでは，多数派の群であるバグなしモジュールを正確に判別することが，（モデル構築時の目的関数である）判別誤差の最小化に大きく寄与するため，少数派の群であるバグありモジュールの判別の正確さが相対的に犠牲となる[19]．そこで，拡張モデルでは，少数派の群の判別精度を向上させるために，誤分類のコストを考慮したモデル構築が行われる．しかし，6 章で後述するように，拡張可能なモデルの種類が限られることが問題となる．一般に，採用すべき（高い判別精度が見込まれる）モデルの種類は，データセットごとに異なる[4]．

本論文では，判別モデル自身を拡張するのではなく，フィットデータに前処理を加えて偏りをあらかじめ解消する方法（サンプリング法）に着目する．この手法は，判別モデルの種類に依存しないため，多くのモデルに適用できるという特長を持つ．この手法はさらに，(1) バグありモジュール（数が少ない方の群）を人工的に増やすオーバサンプリング，(2) バグなしモジュール（数が多い方の群）のいくつかを削除するアンダサンプリング，の 2 通りに分類される[3),11]．ただし，従来，fault-prone モジュール判別モデルにおいて，サンプリング法を適用し，その効果を評価した事例はなく，また，いずれのサンプリング法が最も適しているかも定かではない．さらに，数ある判別モデル（線形判別分析，ロジスティック回帰，ニューラルネットなど）のそれぞれについて，サンプリング法がどの程度有効であるかも定かではない．

そこで，本論文では，日本のあるソフトウェア開発企業で開発された大規模レガシーソフトウェアを対象とし，fault-prone モジュール判別によく用いられている 4 つのモデル（線形判別分析，ロジスティック回帰分析，ニューラルネット，分類木）におけるサンプリング法の効果を実験的に評価した．実験では，4 種類のサンプリング手法（ランダムオーバサンプリング，SMOTE，ランダムアンダサンプリング，Tomek Links を用いたアンダサンプリング）を用い，それぞれサンプリングする割合を変化させて，各 fault-prone モジュール判別モデルの予測精度に与える影響を評価した．実験では，保守におけるあるリリース前の 3 年間に発見されたバグを記録したフィットデータ（バグモジュール含有率 5.67%，および 13.16%）を用い，リリース後 3 年間に検出されたバグ（バグモジュール含有率 2.15%，および 3.36%）の予測（バグあり/なしの判別）を試みた．モデルの説明変数として，ソースコードの複雑さメトリクス 16 個と変更履歴メトリクス 3 個を各モジュールについて計測したものを用いた．

以降，2 章では本論文で用いたサンプリング法について説明する．3 章では判別モデルについて，4 章では評価実験の方法と手順について説明する．5 章では評価実験の結果と，結果に対する考察について議論する．6 章では関連研究を説明する．最後に 7 章でまとめと今後の課題について述べる．

2. サンプリング法

サンプリング法とは，判別モデルを構築する前に，フィットデータに含まれるケース（本論文ではモジュールに該当する）を増加もしくは減少させることで，データセットの偏りを補正する前処理であり，オーバサンプリングとアンダサンプリングに分類される．オーバサンプリングは，少ない方の群のケース（以降，少数派ケース）を人工的に増やすことで多い方の群のケース（以降，多数派ケース）の数に近づけ，データセットの偏りを補正する．一方，アンダサンプリングは，多数派ケースのいくつかを削除することで少数派ケースの数に近づけ，データセットの偏りを補正する．

一般に，判別モデルの構築において，多数派ケースを正確に判別することが判別誤差の最小化につながるため，少数派ケースの判別精度が犠牲となる[19]．サンプリング法を用いることで，群間の偏りがなくなり，結果として，少数派ケースの判別精度の向上が見込まれる．ただし，オーバサンプリングでは，人工的に増やしたケースに過度に適合（オーバフィッティング）するという危険性も否定できない．また，アンダサンプリングでは，母集団の性質を表すのに必要なケース

が削除される危険性がある．

以降では，本論文の評価実験で用いたサンプリング手法について述べる．

2.1 オーバサンプリング

2.1.1 ランダムオーバサンプリング

ランダムオーバサンプリング（以降，ROS）は，少数派ケースをランダムに複製することで少数派ケース数を増加させる．ROS では追加するケースの回数だけ以下の手順を繰り返す．

手順 1. ケースの選択 少数派ケースをランダムに 1 つ選択する．

手順 2. ケースの複製・追加 手順 1 で選択されたケースを複製し，新たなケースとしてデータセットに追加する． □

2.1.2 SMOTE（Synthetic Minority Over-sampling TEchnique）

Chawla らは，ケースの複製によるオーバフィッティングの問題を緩和するために，選択したケースを複製するのではなく，k 最近傍のケースを基に新たなケースを生成する方法である SMOTE を提案している[3]．SMOTE の手順を以下に示す．

手順 1. 説明変数の値の正規化 SMOTE では，k 最近傍のケースを求めるために，各ケースの説明変数の値の組に基づいて，ケース間の類似度を算出する．ただし，fault-prone モジュール判別においては，説明変数（ソースコードの複雑さメトリクスや変更履歴メトリクス）ごとに値域が異なるため，何らかの正規化が必要となる．そこで，本論文では，Chawla らが提案した手法に説明変数の値を正規化する手順を追加し，各説明変数の値域を $[0, 1]$ に統一した．ケース m_i の説明変数 f_j の値 $v_{i,j}$ の正規化された値 $norm(v_{i,j})$ は式 (1) によって求められる．

$$norm(v_{i,j}) = \frac{v_{i,j} - min(f_j)}{max(f_j) - min(f_j)} \quad (1)$$

ここで，$min(f_j)$ と $max(f_j)$ はそれぞれ説明変数 f_j の最小値と最大値である．

手順 2. ケース間の類似度算出 新たなケースを作成するために，各少数派ケース間の類似度を求める．ただし，2 つのベクトル（説明変数の値の組）の類似度の算出方法は多数考えられる．本論文では，ベクトル間の類似度指標として一般的に用いられているユークリッド距離を用いた．ケース m_a とケース m_i との類似度 $sim(m_a, m_i)$ は式 (2) で定義される．

$$sim(m_a, m_i) = \sqrt{\sum_{j=1}^{n} (v_{a,j} - v_{i,j})^2} \quad (2)$$

ここで，$v_{i,j}$ はケース m_i の説明変数 f_j の値で，n はデータセットの説明変数の数である．

手順 3. ケースの追加 手順 2 で求めた類似度を基に k 最近傍を求める．本論文では，k の値として Chawla らが用いた $k = 5$ を採用した[3]．

各少数派ケース m_a について，まず，k 個の最近傍からランダムに 1 つのケース m_r を選択する．次に，ケース m_a と m_r の特徴ベクトルの頂点を結ぶ直線上のランダムな場所に新たなケースを追加する．この手続きを，増加させるケース数となるまで繰り返す． □

2.2 アンダサンプリング

2.2.1 ランダムアンダサンプリング

ランダムアンダサンプリング（以降，RUS）は，データセットから多数派ケースをランダムに選択し，削除することで多数派ケースの数を減少させてデータセットの偏りを解消する手法である．RUS では以下の手順を削除するケースの数だけ繰り返す．

手順 1. ケースの選択 多数派ケースをランダムに 1 つ選択する．

手順 2. ケースの削除 手順 1 で選択されたケースをデータセットから削除する． □

2.2.2 Tomek links を用いたアンダサンプリング

Kubat らは，母集団の性質を表すのに必要なケースを削除する可能性があるアンダサンプリングの問題点を解決するために，Tomek links を用いたアンダサンプリング（以降，TLUS）を提案している[11]．$\delta(x, y)$ を少数派ケース x と多数派ケース y の距離と定義すると，Tomek links とは $\delta(x, z) < \delta(x, y)$ もしくは $\delta(y, z) < \delta(y, x)$ を満たすケース z がない，x と y の組である[20]．つまり，Tomek links である x と y はそれぞれ最も距離が近いケースの組である．

Kubat らは，Tomek links であるケースの組が多数派と少数派の境界付近，もしくは境界の少数派側にある少数派ケースと多数派ケースの組である可能性が高いと考えている．Kubat らは，そのようなケースの組が精度の高い判別モデル構築の妨げになっていると考え，データセットから削除する．さらに Kubat らは，多数派と少数派の境界から距離が遠い多数派の類似したケース（**図 1** の右上の多数派ケース）が判別モデル構築の際に冗長であると考え，データセットから削除する．TLUS の手順を以下に示す．

手順 1. 説明変数の値の正規化 本論文では，

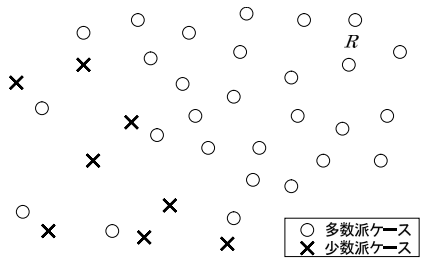

図 1 多数派ケースと少数派ケースの分布の例（データセット S）
Fig. 1 An example of distribution of major cases and minor cases (Dataset S).

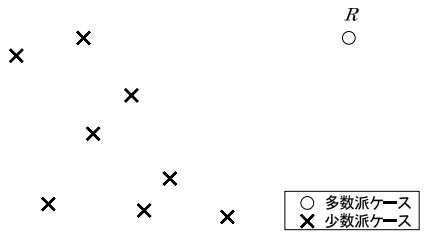

図 2 ケース R とすべての少数派ケース（データセット C）
Fig. 2 The case R and all minority cases (Dataset C).

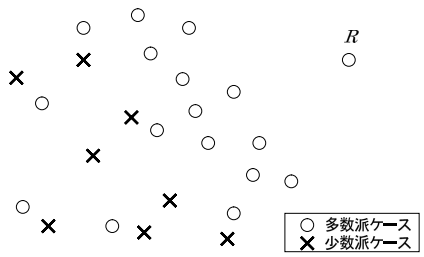

図 3 冗長な多数派ケースを削除して得られたデータセット C'
Fig. 3 The dataset C' derived by eliminating redundant majority cases.

SMOTE と同様に，Kubat らが提案している手法に説明変数の値の正規化という手順を追加する．正規化の手順も SMOTE と同様である．

手順 2. 冗長なケースの削除 データセット S（図 1）から，多数派ケースからランダムに選択した 1 つのケース R とすべての少数派ケースを取り出し，データセット C（**図 2**）を作成する．データセット S の各多数派ケースについて，データセット C のケースの中から k 最近傍（$k = 1$）を求め，ケース R が最近傍として選ばれたケースを冗長なケースとして削除し，データセット C'（**図 3**）を作成する．

本論文では，k 最近傍を求めるための類似度指標としてユークリッド距離を用いた．各ケース間の類似度は，SMOTE と同様に式 (2) で定義される．

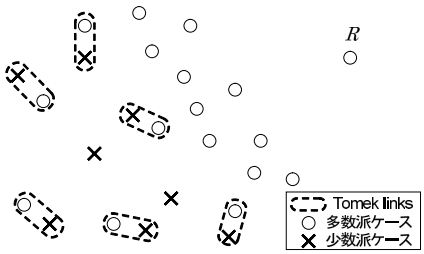

図 4 データセット C' に含まれる Tomek links
Fig. 4 Tomek links in the dataset C'.

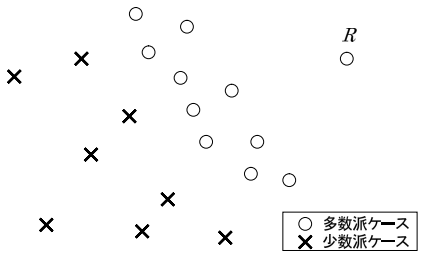

図 5 TLUS の出力結果
Fig. 5 The output of TLUS.

手順 3. Tomek links の削除 冗長な多数派ケースを削除したデータセット C' から，Tomek links の定義を満たす多数派ケースと少数派ケースの組の多数派ケース（**図 4** の破線で囲まれた多数派ケース）を削除する．その結果，アンダサンプリングされたデータセット（**図 5**）が出力される．

□

3. Fault-prone モジュール判別モデル

Fault-prone モジュール判別では，過去に開発されたモジュールのメトリクス値とバグの有無から判別モデルを構築する．構築したモデルに対して，バグあり/なしを判別したいモジュールから計測したメトリクス値を入力することで，当該モジュールがバグありであるかバグなしであるかが判別される．判別モデルの種類は多数あるが，fault-prone モジュール判別においては，下記の 4 つのモデルがよく用いられており，本論文の評価実験においても採用する．

3.1 線形判別分析

線形判別分析では，判別モデル（判別関数）はケースの特性を表す説明変数の線形結合により表される．各ケースに対して p 個の説明変数の値が観測されているとき，線形判別関数は式 (3) の形となる．

$$Z = \alpha_1 x_1 + \cdots + \alpha_p x_p \qquad (3)$$

ここで，x_i は説明変数，α_i は判別係数と呼ばれる定数である．本論文では，判別値 Z が 0 以上の値をとる場合，バグありモジュールであると判定する．

3.2 ロジスティック回帰分析

ロジスティック回帰分析では，判別関数はロジスティック関数の形で表現される．各ケースに対して p 個の説明変数の値が観測されているとき，判別関数は式 (4) の形となる．

$$P(y|x_1,\cdots,x_p)=\frac{1}{1+e^{-(\alpha_1 x_1+\cdots+\alpha_p x_p+\beta)}} \tag{4}$$

ここで，$y\in\{0,1\}$ は群を表す目的変数，x_i は説明変数，α_i は判別係数であり，$P(y|x_1,\cdots,x_p)$ は，説明変数の値の組 $x_1,\cdots,x_p$ に対して，y が 1 の値をとる確率である．本論文では $P(y|x_1,\cdots,x_p)$ が 0.5 以上の値をとる場合，バグありモジュールであると判定する．

3.3 ニューラルネットワーク

ニューラルネットワークモデルは，人間の脳内の神経細胞間の信号伝達を模したネットワーク構造により説明変数と目的変数の関係を表現する[1)]．モデルの出力に閾値を設定することで，2 群判別に用いることができる．本論文では，ニューラルネットワークモデルに 3 層のパーセプトロン（中間層は 3 個[12)]）を，学習アルゴリズムに誤差逆伝播法[18)] を用い，2 群を判別するための出力値が 0.5 以上の場合，バグありモジュールであると判定する．

3.4 分 類 木

分類木は，説明変数と目的変数の関係を木構造で表現したモデルである．木の各ノードは 2 つ以上の子ノードを持っており，説明変数の値によっていずれかの子ノードへと分岐する．リーフノードは，いずれかの群に割り当てられている．与えられたケースは，説明変数の値に従ってルートノードから木をたどることで，リーフノードにおいて群の判別が行われる．本論文では，分類木を構築するためのアルゴリズムとして fault-prone モジュール判別で一般的に用いられている CART（Classification And Regression Trees）[8)] を用いた．

4. 適 用 実 験

4.1 概 要

本実験では，バグありモジュールの個数が（バグなしモジュールと比べて）著しく少ないデータセットを対象とし，サンプリング法適用の効果を実験的に評価する．実験では，4 種類のサンプリング法（ROS, SMOTE, RUS, TLUS），4 種類のモデル（線形判別分析，ロジスティック回帰分析，ニューラルネット，分類木）のすべての組合せについて，それぞれ判別精度を評価する．

なお，ROS, SMOTE, RUS の 3 つについては，サンプリングを行う度合いを調整することが可能である．たとえば，ROS では，追加するケースの数を変化させることで，バグあり/バグなしモジュールの割合（以降，モジュール比）を自由に調整可能である．この割合は，偏りをなくす観点からは 1 : 1 に設定すべきであるが，2 章で述べたようにケースを追加/削除しすぎると，オーバフィッティングや必要なケースが削除されるという問題があるため，1 : 1 が必ずしも最適とは限らない．

そこで本実験では，まず，フィットデータだけを用いて，最適なモジュール比を決定するための事前実験を行う．次に，事前実験で決定したモジュール比に基づいて，フィットデータに対してサンプリング法を適用し，判別モデルを構築する．各サンプリング法はランダム要素を含むため，各手法で 5 回ずつサンプリング法を適用し，それぞれ判別モデルを構築する．最後に，テストデータを用いて判別モデルの予測性能（5 回の平均値）を，サンプリング法を適用しない場合と比較対照することで評価する．

本実験では，4 種類の fault-prone モジュール判別モデルの構築に，SPSS 社のデータマイニングソフトである Clementine を用いた．線形判別分析とロジスティック回帰分析のモデル構築では，変数選択（変数減少法）を行った．ニューラルネットのモデル構築では，サンプリング法を適用した場合モデルの学習が 85%未満で収束しないため，予測精度が 80%以上であることを学習の停止条件とした．

4.2 データセット

本実験で対象としたデータセットは，日本のあるソフトウェア開発企業で開発された，大規模レガシーソフトウェアの保守期間中に収集された各モジュールのメトリクス値を記録したものである．開発されたソフトウェアは，Capers Jones の分類[7)] における MIS（Management Information System）に該当する，企業，官公庁，銀行などにおける業務の基盤となるアプリケーションソフトウェアであり，主に 2 種類の開発言語で記述されている．本ソフトウェアは，初期リリースから約 20 年間保守され続け，ある期間（数年ごと）で新たな要求による機能追加や変更，および多くのバグが修正されリリースされている．

本実験では，保守におけるあるリリース前の 3 年間

表 1 各データセットのバグあり/なしモジュールの個数とバグモジュール含有率
Table 1 The number of modules and ratio of modules in each dataset.

	バグありモジュール数（個）	バグなしモジュール数（個）	バグモジュール含有率（%）
A_{fit}	103	1,815	5.67
A_{test}	42	1,950	2.15
B_{fit}	210	1,596	13.16
B_{test}	61	1,815	3.36

表 2 適用実験に用いたデータセットのメトリクスの名称
Table 2 Metrics of the experiment data.

	メトリクス
m1	バグの有無
m2	プログラムの総行数（Lines of Code）
m3	コメント行数/LOC
m4	モジュール内手続き数/LOC
m5	オペレータの種類
m6	オペランドの種類
m7	オペレータ総出現数/LOC
m8	オペランド総出現数/LOC
m9	Halstead 尺度の Volume
m10	Halstead 尺度の Difficulty
m11	最大ネストレベル
m12	サイクロマティック数/LOC
m13	ネストレベル/LOC
m14	ジャンプ文数（GOTO 文）/LOC
m15	外部変数参照数/LOC
m16	内部手続き呼び出し数/LOC
m17	外部モジュール呼び出し数/LOC
m18	改訂数
m19	モジュールが生成された日からの日数
m20	モジュールが生成された日から最終改訂日までの日数

に発見されたバグ（バグモジュール含有率 5.67%，および 13.16%）の記録をフィットデータとして用い，リリース後 3 年間に検出されたバグ（バグモジュール含有率 2.15%，および 3.36%）をテストデータとして用いた．モデルの説明変数として，ソースコードの複雑さメトリクス 16 個と変更履歴メトリクス 3 個を各モジュールについて計測したものを用いた．

本実験では，一方の言語で実装されたモジュールの集合をデータセット A，もう一方の言語のモジュールの集合をデータセット B と呼ぶ．また，A，B それぞれのデータセットにおけるフィットデータを A_{fit}，B_{fit}，テストデータを A_{test}，B_{test} と記す．各データセットのバグありモジュールの割合を**表 1** に示す．

フィットデータとテストデータでモジュール数が異なる理由は，リリースにおいてモジュールが新たに追加されたためである．それぞれのデータセットの 20 個のメトリクスの名称を**表 2** に示す．語尾に「/LOC」がついているものは，行数で正規化されたメトリクスである．表 2 に示すメトリクスのうち，モジュールのバグの有無を目的変数とし，それ以外のメトリクスを説明変数の候補とした．

表 3 判別結果の分類
Table 3 Classification of detection results.

	バグがないと予測	バグがありと予測
実際にバグなし	n_{11}	n_{12}
実際にバグあり	n_{21}	n_{22}

4.3 評価基準

バグモジュール判別の評価基準として，再現率，適合率，および F1 値[5] を用いた．再現率 R は，すべてのバグを含むモジュールのうち，バグを含むモジュールと予測した割合を表し，**表 3** で示す記号を用いて式 (5) で定義される．

$$R = \frac{n_{22}}{n_{21} + n_{22}} \tag{5}$$

また，適合率 P は，バグありモジュールと予測したうち，実際にバグありモジュールである割合を表し，表 3 で示す記号を用いて式 (6) で定義される．

$$P = \frac{n_{22}}{n_{12} + n_{22}} \tag{6}$$

再現率と適合率はトレードオフの関係にあるため，再現率と適合率だけで判別精度を評価することは難しい．そこで，本論文では適合率と再現率の調和平均で与えられる F1 値を評価基準として用いた．F1 値 F は，式 (7) として定義され，値が大きいほど判別精度が高いことを表す．

$$F = \frac{2 \times R \times P}{R + P} \tag{7}$$

4.4 実験の手順

4.4.1 最適なモジュール比の決定（事前実験）

サンプリング法適用時の最適なモジュール比を決定するための事前実験を，フィットデータだけを用いて（交差検証法により）行った．最適なモジュール比は，TLUS 以外の 3 種類のサンプリング法（ROS, SMOTE，RUS）と 4 種類のモデル（線形判別分析，ロジスティック回帰分析，ニューラルネットワーク，分類木）の組合せ計 12 通りについて求めた．具体的には，12 通りの組合せそれぞれについて下記の手順を実施した．

手順 1. フィットデータの分割 フィットデータを 2 等分し，一方のデータをデータセット a，もう一方のデータをデータセット b とした．

手順 2. サンプリング法の適用 フィットデータ a に対して，モジュール比が (0.25 : 1.00)，(0.50 : 1.00)，(1.00 : 1.00)，(1.50 : 1.00)，(2.00 : 1.00)

となるようにサンプリングを行い，フィットデータ $a_1, \cdots, a_5$ を作成した．

手順 3. 判別モデルの構築 フィットデータ $a_1, \cdots, a_5$，および，サンプリング法を適用しない場合の判別モデル a_0 それぞれを用いて，判別モデルを 6 つ構築する．

手順 4. 判別モデルの精度の評価 テストデータ b を用いて，各判別モデルの精度を求めた．

手順 5. モジュール比の決定 最も精度の良い（F1 値の大きい）判別モデルが得られたモジュール比を採用する． □

4.4.2 サンプリング法の効果の評価

4.4.1 項の事前実験で決定したモジュール比に基づいて，次の手順によりサンプリング法の効果を評価した．

手順 1. サンプリング法の適用 事前実験で決定したモジュール比になるように，フィットデータに対してサンプリング法を適用した．

手順 2. 判別モデルの構築 サンプリング法を適用したフィットデータを用いて，4 種類の判別モデルを構築する．

手順 3. モジュールの判別 構築した各 fault-prone モジュール判別モデルを用いて，テストデータの各モジュールのバグの有無を予測した．

手順 4. 判別モデルの精度の評価 予測したモジュールのバグの有無と，実際のモジュールのバグの有無から 4.3 節で説明した各評価基準で各判別モデルの精度を求めた． □

4.4.3 対 照 実 験

サンプリング法の効果を調べるために，対照実験としてサンプリング法を適用しない場合の判別モデルを構築し，精度評価を行った．手順は次のとおりである．

手順 1. 判別モデルの構築 フィットデータを用いて，4 種類の判別モデルを構築した．

手順 2. モジュールの判別 構築した各判別モデルを用いて，テストデータの各モジュールのバグの有無を予測した．

手順 3. 判別モデルの精度の評価 予測したモジュールのバグの有無と，実際のモジュールのバグの有無から各評価基準で各判別モデルの精度を求めた． □

5. 結果と考察

5.1 モジュール比に関する結果と考察

各サンプリング法と判別モデルの組合せについて，サンプリング法適用時のモジュール比を変化させたときの F1 値の変化を**図 6** に示す．グラフの横軸はモジュール比を，縦軸は F1 値を示す．グラフの左端の値は，サンプリング法を適用しなかった場合の F1 値を示す．また，事前実験において決定された最適なモジュール比を，各サンプリング手法，および各判別モデルについてまとめたものを，データセット A については**表 4** に，データセット B については**表 5** に示す．

図 6 と表 4，表 5 より，モジュール比 1 : 1 が必ずしも最適とならない（むしろ最適でないことの方が多い）ことが分かった．また，モジュール比と F1 値の関係に着目すると，データセットにかかわらず，判別モデルごとに次の特徴が見られた．

- **線形判別分析** サンプリング未適用またはモジュール比が小さい（0.25）ときに最も精度が低く，モジュール比が大きい（1.50 または 2.00）ときに最も精度が高くなっている．
- **ロジスティック回帰分析** モジュール比が 0.25〜0.50 のときに最も精度が高くなり，モジュール比をさらに上げると精度が下がっていく傾向にある．これは，線形判別分析と逆の傾向にあるといえる．
- **ニューラルネットワーク** サンプリング未適用では F1 値がゼロ（全モジュールをバグなしと予測）であった．モジュール比が 0.50 付近のときに最も精度が高くなり，モジュール比をさらに上げると精度が下がっていく傾向にある．ロジスティック回帰分析と似た傾向にあるといえる．
- **分類木** サンプリング法の適用によって判別精度が低下する場合がある．また，精度が向上する場合でも，その効果は小さい．

以上のことから，モデルによって最適なモジュール比は異なることが分かった．また，いずれのデータセット，サンプリング法を用いた場合においても，平均的な傾向として，(1) 線形判別分析ではモジュール比 2.00 付近が良い，(2) ロジスティック回帰分析およびニューラルネットではモジュール比 0.50 付近が良い，(3) 分類木ではサンプリング法の効果はほとんど見られない，ことが分かった．

(1)，(2) の結果から，線形判別分析はモジュール比が 1.0 を超えるほどの強いオーバ/アンダサンプリングに対してもロバスト（精度の低下を招きにくい）であり，逆にロジスティック回帰分析とニューラルネットワークは強すぎるオーバ/アンダサンプリングに対してロバストでないことがうかがえる．1 つの解釈として，線形判別分析は（他の 2 つのモデルよりも）モデル自身が単純であるために，追加/削除されたケースに過度に適合する（オーバフィッティング）ことが

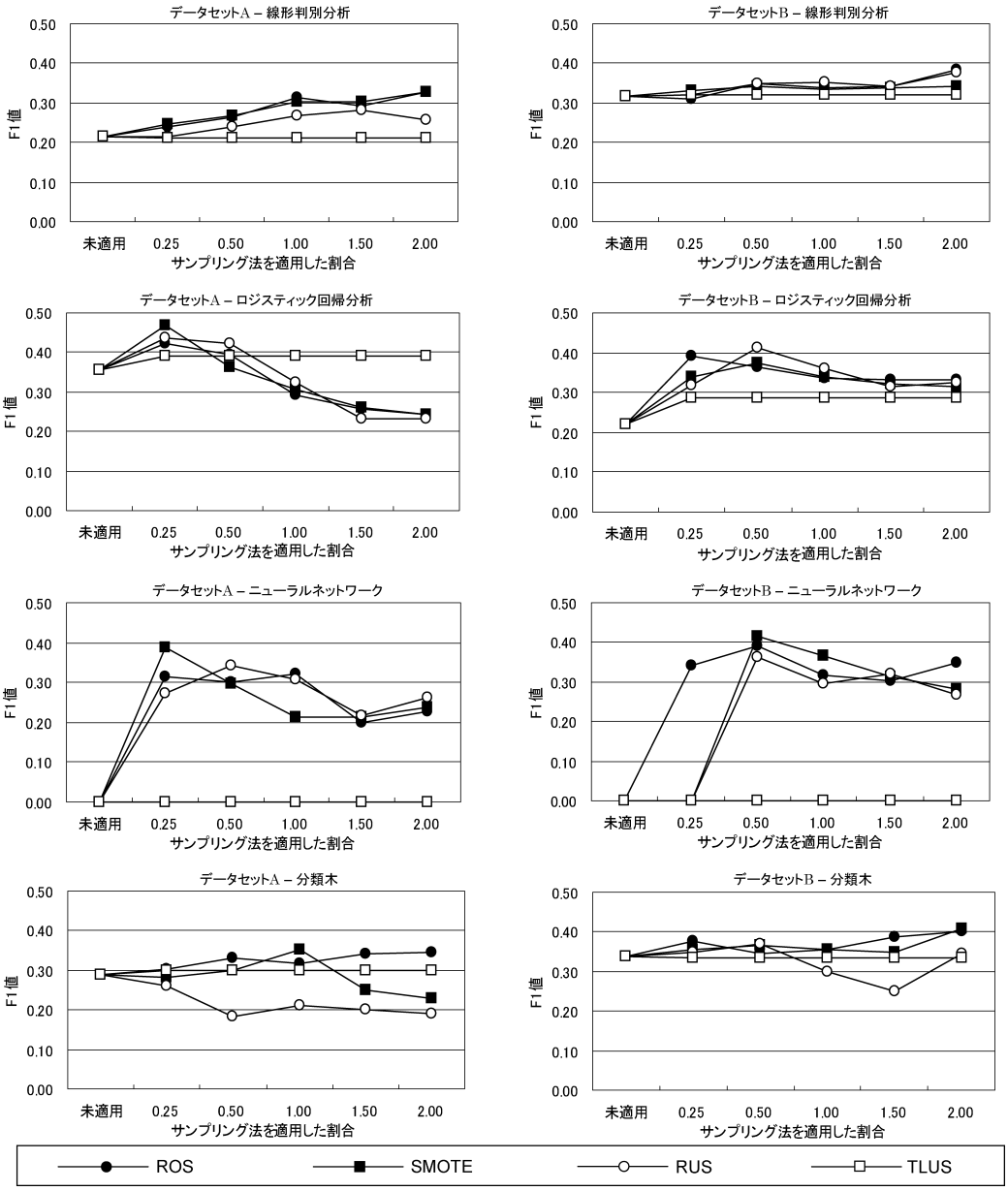

図 6 構築する判別モデルごとのサンプリング法の効果
Fig. 6 Effects of sampling methods in each prediction model.

起こらず，結果としてロバストであったことが考えられる．

また，(3) の分類木で効果がほとんどなかったことの原因として，本論文は分類木の構築アルゴリズムに CART を用いており，その分岐基準として Gini 係数を用いていたことがあげられる．このアルゴリズムは，2 クラス間のケース数の偏りを考慮して木を構築するため[2]，オーバ/アンダサンプリングによる偏り補正の恩恵をほとんど受けなかったと考えられる．ただし，分類木の構築にはほかにも多数のアルゴリズムが存在し（CHAID，C4.5 など），特に CHAID については Khoshgoftaar ら[10] の研究で，拡張を行わないと少数派クラスに対する精度が大幅に低下することが示されていることから，オーバ/アンダサンプリングが有効となる可能性がある．CART 以外の分類木構築アルゴリズムに対して，オーバ/アンダサンプリング適用の効果を評価することは，残された重要な課題である．

表 4 事前実験により決定されたモジュール比（データセット A）
Table 4 Ratio of bug modules to non-bug modules of dataset A.

	線形判別分析	ロジスティック回帰分析	ニューラルネットワーク	分類木
ROS	2.00：1.00	0.25：1.00	0.25：1.00	2.00：1.00
SMOTE	2.00：1.00	0.25：1.00	0.25：1.00	1.00：1.00
RUS	1.50：1.00	0.25：1.00	0.50：1.00	未適用
TLUS	未適用	適用	適用	適用

表 5 事前実験により決定されたモジュール比（データセット B）
Table 5 Ratio of bug modules to non-bug modules of dataset B.

	線形判別分析	ロジスティック回帰分析	ニューラルネットワーク	分類木
ROS	2.00：1.00	0.25：1.00	0.50：1.00	2.00：1.00
SMOTE	2.00：1.00	0.50：1.00	0.50：1.00	2.00：1.00
RUS	2.00：1.00	0.50：1.00	0.50：1.00	0.50：1.00
TLUS	適用	適用	適用	未適用

表 6 判別モデルとサンプリング法の各組合せの判別精度（データセット A）
Table 6 Detection accuracy on dataset A.

		線形判別分析	ロジスティック回帰分析	ニューラルネットワーク	分類木
再現率	未適用	0.857	0.238	0.000	0.762
	ROS	0.752	0.629	0.586	0.648
	SMOTE	0.791	0.595	0.448	0.733
	RUS	0.833	0.624	0.662	0.762
	TLUS	0.857	0.286	0.019	0.729
適合率	未適用	0.057	0.357	0.000	0.062
	ROS	0.146	0.266	0.159	0.094
	SMOTE	0.137	0.282	0.244	0.080
	RUS	0.117	0.257	0.129	0.062
	TLUS	0.057	0.375	0.050	0.088
F1 値	未適用	0.106	0.286	0.000	0.115
	ROS	0.244	0.374	0.247	0.162
	SMOTE	0.233	0.382	0.303	0.143
	RUS	0.205	0.364	0.211	0.115
	TLUS	0.106	0.324	0.028	0.156

5.2 サンプリング法適用の結果と考察

事前実験において決定された最適なモジュール比に従って，サンプリング法の適用，判別モデルの構築，および予測精度の評価を行った結果を，データセット A については**表 6** に，データセット B については**表 7** に示す．表 6，表 7 中の「未適用」は，サンプリング法を適用しなかった場合を示す．

- **判別モデル間の比較** 分類木を除いて，すべてのデータセットおよび判別モデルにおいて，4 種類のサンプリング法のいずれを適用した場合にも予測精度が向上した．
  サンプリング法を適用した場合としなかった場合の分類木を比較したところ，木の構造において各ノードの分岐の条件に大きな違いは見られなかった．
  なお，最も精度（F1 値）の高い判別モデルは，データセット A ではロジスティック回帰分析，データセット B では線形判別分析となった．この結果は，最適なモデルはデータセットによって異なるという Gray らの結果[4] と反しない．
  また，表 6，表 7 の F1 値に着目すると，サンプリング法を適用しない場合に最も F1 値が高いモデルは，サンプリング法を適用した場合においても最も F1 値が高いモデルであることが分かった．たとえば，表 6 ではサンプリング法を適用しない場合にロジスティック回帰分析が最も精度の高いモデルであり，サンプリング法を適用した場合においても最も F1 値が高い．このことから，開発・保守の現場で利用する際に，あらかじめどのモデルの精度が高いかが判明している場合，そのモデルとサンプリング法を組み合わせることで，より性能の高い判別モデルを構築できると期待される．
- **サンプリング手法間の比較** 最も効果の高いサンプリング手法は，判別モデルごと，および，データセットごとに異なっていた．たとえば，データセット A では，SMOTE とロジスティック回帰分析の組合せにおいて最も F1 値が高くなった（0.382）．データセット B では，RUS と線形判別分析の組合せが最も高くなった（0.324）．平均的な傾向としては，TLUS はいずれのデータセットおよび判別モデルにおいても効果が小さく（平均 F1 値は 0.126），残りの 3 つ（ROS，SMOTE，RUS）は互いに優劣がつけられなかった（それぞれ平均 F1 値は 0.217，0.222，0.216 であった）．この傾向は，F1 値そのものではなく，（サンプリング法

表 7 判別モデルとサンプリング法の各組合せの判別精度（データセット B）
Table 7 Detection accuracy on dataset B.

		線形判別分析	ロジスティック回帰分析	ニューラルネットワーク	分類木
再現率	未適用	0.590	0.033	0.000	0.393
	ROS	0.420	0.138	0.154	0.325
	SMOTE	0.443	0.220	0.144	0.220
	RUS	0.482	0.328	0.194	0.377
	TLUS	0.669	0.033	0.006	0.393
適合率	未適用	0.118	0.200	0.000	0.068
	ROS	0.234	0.299	0.103	0.061
	SMOTE	0.249	0.210	0.092	0.043
	RUS	0.244	0.246	0.108	0.052
	TLUS	0.129	0.182	0.025	0.068
F1 値	未適用	0.196	0.056	0.000	0.116
	ROS	0.300	0.187	0.123	0.101
	SMOTE	0.319	0.215	0.112	0.072
	RUS	0.324	0.280	0.138	0.090
	TLUS	0.217	0.056	0.010	0.116

適用による）F1 値の上昇幅に着目した場合においても，同様であった．

TLUS によって削除されたモジュール数について調べると，データセット A_{fit} では 1,815 個のバグなしモジュールのうち，145 個のモジュールしか削除されておらず，バグモジュール含有率は 5.67%から 6.17%にしか変化していなかった．また，データセット B_{fit} では 1,596 個のバグなしモジュールのうち，147 個のモジュールしか削除されておらず，バグモジュール含有率は 13.16%から 14.49%にしか変化していなかった．すなわち，TLUS を適用してもデータセットの偏りが解消されていない．そのため，予測精度が向上しなかったと考えられる．TLUS では削除するモジュール数を指定することができないため，今回の結果のようにデータセットの偏りが解消されない可能性がある．今後の改善策として，フィットデータに TLUS を繰り返し適用することで，モジュール比を制御することが考えられる．

6. 関 連 研 究

モデル構築手法を改良することでデータセットの偏りの問題の解決を目指した研究として，Khoshgoftaar らの研究[10] がある．Khoshgoftaar らは，誤分類のコストに基づいて，分類木のリーフノードにおけるバグあり/なしモジュールの判別基準を変更する方法を提案している．バグモジュール含有率約 6%のフィットデータを用いてバグモジュール含有率約 5%のテストデータの予測を行った評価実験において，従来の分類木よりも再現率が向上したことが示されている．ただし，この方法は，分類木以外には適用できない．また，フィットデータとテストデータとの間でバグモジュール含有率に大きな差がある場合に，この方法がうまく働くかどうかは定かでない．

偏りのあるフィットデータを人工的に作成し，サンプリング法の効果を調べた研究として，Japkowicz らの研究[6] がある．Japkowicz らは，データセットの大きさと複雑さとモジュール比を調整して人工的に作成したフィットデータを用いて，サンプリング法の効果を実験的に評価している．ただし，この研究では，説明変数が 1 つのみであるなどソースコードから収集可能なデータセットが想定されておらず，fault-prone モジュール判別モデルに対してサンプリング法の適用の効果については定かではない．

今後，誤分類のコストを考慮した分類木と，サンプリング法（と線形判別分析，ロジスティック回帰分析，ニューラルネットの組合せ）との精度比較を行うことは，残された重要な課題の 1 つである．

7. お わ り に

本論文では，fault-prone モジュール判別モデルの構築において，2 群のケース数の偏りを解消するためにサンプリング法を用いた場合の効果を実験的に評価した．実験では，日本のあるソフトウェア開発企業で開発された大規模レガシーソフトウェアを対象とし，fault-prone モジュール判別によく用いられている 4 つのモデル（線形判別分析，ロジスティック回帰分析，ニューラルネット，分類木）と 4 種類のサンプリング法（ROS，SMOTE，RUS，TLUS）を用いた．

実験により得られた主な結果および知見は，下記のとおりである．

- 分類木以外の判別モデルに対してはサンプリング法の効果が見られた．分類木以外の判別モデルに TLUS 以外のサンプリング法を適用した場合，F1 値の向上は最小 0.078，最大 0.303，平均 0.146 であった．なお，ROS，SMOTE，RUS を適用した判別モデルの平均 F1 値は，それぞれ 0.217，0.222，0.216 であり，優劣がつけられなかった（TLUS を適用した判別モデルの平均 F1 値は 0.126）．
- TLUS は他の 3 つのサンプリング法よりも効果が小さかった．TLUS によって削除されたモジュール数について調べると，それぞれのデータセットのバグモジュール含有率は 5.67%から 6.17%，13.16%から 14.49%にしか変化していなかった．
- モデルによって最も高い判別精度が得られるモジュール比に違いがあった．線形判別分析ではモジュール比 2.00 付近で最も良い精度が得られ，ロジスティック回帰分析とニューラルネットではモジュール比 0.50 付近で最も良い精度が得られた．分類木ではサンプリング法の効果はほとんど得られなかった．
- サンプリング法を適用しない場合に最も F1 値が高いモデルは，サンプリング法を適用した場合においても最も F1 値が高いモデルであった．

なお，本論文では，2 つのデータセットを用いた評価しか行っていない点に注意する必要がある．結果の信頼性を向上させるために，他のデータセットを用いたり，データセットの偏りを変化させたりするといった実験を行うことが今後の課題となる．

謝辞 本研究の一部は，文部科学省「e-Society 基盤ソフトウェアの総合開発」の委託に基づいて行われた．

参 考 文 献

1) 麻生英樹：ニューラルネットワーク情報処理—コネクショニズム入門，あるいは柔らかな記号に向けて，p.198, 産業図書 (1988).
2) Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J.: *Classification And Regression Trees*, p.358, Chapman & Hall/CRC (1998).
3) Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-sampling Technique, *Journal of Artificial Intelligence Research*, Vol.16, pp.321–357 (2002).
4) Gray, A.R. and MacDonell, S.G.: Software Metrics Data Analysis—Exploring the Relative Performance of Some Commonly Used Modeling Techniques, *Empirical Software Engineering*, Vol.4, No.4, pp.297–316 (1999).
5) Herlocker, J.L., Konstan, J.A., Terveen, L.G. and Riedl, J.T.: Evaluating Collaborative Filtering Recommender Systems, *ACM Trans. Information Systems*, Vol.22, No.1, pp.5–53 (2004).
6) Japkowicz, N. and Stephen, S.: The Class Imbalance Problem: A Systematic Study, *Intelligent Data Analysis*, Vol.6, No.5, pp.429–450 (2002).
7) Jones, C.: *Applied Software Measurement*, *2nd Edition*, p.861, McGraw-Hill, New York (1996).
8) Khoshgoftaar, T.M. and Allen, E.B.: Modeling Software Quality with Classification Trees, *Recent Advances in Reliability and Quality Engineering*, pp.247–270, World Scientific, Singapore (1999).
9) Khoshgoftaar, T.M., Gao, K. and Szabo, R.M.: An Application of Zero-inflated Poisson Regression for Software Fault Prediction, *Proc. 12th Int'l Symposium on Software Reliability Engineering* (*ISSRE'01*), Hong Kong, China, pp.66–73 (2001).
10) Khoshgoftaar, T.M., Yuan, X. and Allen, E.B.: Balancing Misclassification Rates in Classification-Tree Models of Software Quality, *Empirical Software Engineering*, Vol.5, No.4, pp.313–330 (2000).
11) Kubat, M. and Matwin, S.: Addressing the Curse of Imbalanced Training Sets: One-Sided Selection, *Proc. 14th Int'l Conf. on Machine Learning* (*ICML'97*), Nashville, USA, pp.179–186 (1997).
12) 栗田多喜夫：情報量基準による 3 層ニューラルネットの隠れ層のユニット数の決定法，電子情報通信学会論文誌，Vol.J73-D-II, No.11, pp.1872–1878 (1990).
13) Li, P.L., Herbsleb, J., Shaw, M. and Robinson, B.: Experiences and Results from Initiating Field Defect Prediction and Product Test Prioritization Efforts at ABB Inc, *Proc. 28th Int'l Conf. on Software Engineering* (*ICSE'06*), Shanghai, China, pp.413–422 (2006).
14) Munson, J.C. and Khoshgoftaar, T.M.: The Detection of Fault-prone Programs, *IEEE Trans. Softw. Eng.*, Vol.18, No.5, pp.423–433 (1992).
15) NASA/WVU IV&V Facility: Metrics Data Program. http://mdp.ivv.nasa.gov/
16) Ohlsson, N. and Alberg, H.: Predicting Fault-Prone Software Modules in Telephone Switches, *IEEE Trans. Softw. Eng.*, Vol.22, No.12, pp.886–894 (1996).
17) Pighin, M. and Zamolo, R.: A Predictive Metric Based on Discriminant Statistical Analysis, *Proc. 19th Int'l Conf. on Software Engineering* (*ICSE'97*), Boston, USA, pp.262–270 (1997).
18) Rumelhart, D.E., Hinton, G.E. and Williams, R.J.: Learning Representations by Back-propagating Errors, *Nature*, Vol.323, pp.533–536 (1986).
19) 鈴木英之進：正確な学習よりも得する学習—学習コストを考慮する分類学習（1）評価編，情報処理，Vol.45, No.4, pp.395–401 (2004).
20) Tomek, I.: Two Modifications of CNN, *IEEE Trans. Systems, Man and Cybernetics*, SMC-6, pp.769–772 (1976).

(平成 18 年 12 月 5 日受付)
(平成 19 年 5 月 9 日採録)

亀井 靖高（学生会員）

平成 17 年関西大学総合情報学部卒業．平成 19 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了．現在，同大学博士後期課程在籍．エンピリカルソフトウェア工学，特にソフトウェア信頼性の研究に従事．電子情報通信学会，IEEE 各会員．

松本 真佑（学生会員）

平成 18 年京都産業大学理学部卒業．現在，奈良先端科学技術大学院大学情報科学研究科博士前期課程在籍．エンピリカルソフトウェア工学，特にソフトウェアメトリクスの研究に従事．電子情報通信学会，IEEE 各会員．

柿元　健（学生会員）

平成 15 年神戸市立工業高等専門学校専攻科電気電子工学専攻修了．平成 17 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了．現在，同大学博士後期課程在学中．ソフトウェア信頼性/開発工数予測等の研究に従事．電子情報通信学会，IEEE 各会員．

門田　暁人（正会員）

平成 6 年名古屋大学工学部電気学科卒業．平成 10 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了．同年同大学同研究科助手．平成 16 年同大学助教授．平成 19 年同大学准教授．平成 15〜16 年 Auckland 大学客員研究員．博士（工学）．ソフトウェアメトリクス，ソフトウェアセキュリティ，ヒューマンインタフェースの研究に従事．電子情報通信学会，日本ソフトウェア科学会，教育システム情報学会，IEEE，ACM 各会員．

松本　健一（正会員）

昭和 60 年大阪大学基礎工学部情報工学科卒業．平成元年同大学大学院博士課程中退．同年同大学基礎工学部情報工学科助手．平成 5 年奈良先端科学技術大学院大学助教授．平成 13 年同大学教授．工学博士．エンピリカルソフトウェア工学，特に，プロジェクトデータ収集/利用支援の研究に従事．電子情報通信学会，日本ソフトウェア科学会，ACM 各会員，IEEE Senior Member．