

令和 6 年 6 月 20 日現在

機関番号：34419

研究種目：基盤研究(C) (一般)

研究期間：2021～2023

課題番号：21K11840

研究課題名(和文) ソフトウェア品質向上に向けた欠陥予測モデルのオンライン最適化の検討

研究課題名(英文) Online Optimization of Defect Prediction Models Towards High Quality Software

研究代表者

角田 雅照 (Tsunoda, Masateru)

近畿大学・情報学部・准教授

研究者番号：60457140

交付決定額(研究期間全体)：(直接経費) 3,100,000円

研究成果の概要(和文)：ソフトウェア品質向上に向け、欠陥予測モデルのオンライン最適化を検討した。成果は3種類に分類され、具体的には(1)欠陥予測モデルの精度向上、(2)オンライン最適化の欠陥予測以外への適用、(3)オンライン最適化自体の改善である。(1)については変数選択方法、アンサンブル学習方法などの選択に関して、オンライン最適化を行った。(2)についてはコードクローン検出、コード自動生成、ソフトウェアレビューなどに対してオンライン最適化を適用した。(3)については、オンライン学習方法の問題点とその改善方法、ソフトウェアテストの再実施方法の2つを新たに提案した。これらより、ソフトウェア品質向上が期待される。

研究成果の学術的意義や社会的意義

ソフトウェア欠陥予測において汎用的なモデルや手法は存在しない。このため、平均的に性能の高い予測方法などを事前に評価する研究が広く行われてきた。このような従来のアプローチでは(1)事前に様々な手法を評価する必要があり、評価のための時間とコストが掛かる、(2)平均的に性能の高い手法が、適用対象のプロジェクトで性能が高いとは限らない、という問題点があった。本研究のオンライン最適化により(1)の事前評価が必須ではなくなり、新しい手法を積極的に利用可能とした、(2)適用対象のプロジェクトにおいて性能が低下するリスクを避けることができ、予測方法などを実プロジェクトに積極的に導入することを可能にした。

研究成果の概要(英文)：To enhance the quality of software, we have focused on the online optimization of defect prediction models. The achievements of this project are classified into three types: (1) accuracy improvement of defect prediction models, (2) application of online optimization besides defect prediction, and (3) improvement of online optimization. Specifically, (1) involves online optimization to select better methods such as variable reduction methods and ensemble learning methods. (2) applies online optimization to activities such as code clone detection, code generation, and software review. (3) clarifies the problem of online learning and its improvement, and proposes a new approach to software testing. The achievements of our project are expected to bring high-quality software.

研究分野：ソフトウェア工学

キーワード：多腕バンディット問題 機械学習 ソフトウェア開発プロジェクト 性能評価

1. 研究開始当初の背景

近年、ソフトウェアは社会のあらゆる場面で利用されており、ゆえにソフトウェアに障害が発生した場合、社会に与える悪影響は広く大きなものとなる。ソフトウェアの利用時に障害が発生することを防ぐためには、ソフトウェア開発時にテストを実施して動作を確認し、障害の原因となる欠陥が発見されればこれを除去しておく必要がある。ただし、近年ソフトウェアは非常に大規模化しており、ソフトウェアに含まれる全てのモジュールを厳密にテストするには非常に大きな労力が必要となっている。そのため、欠陥発生の可能性が高いモジュールを重点的にテストするアプローチがしばしば取られる。

重点的にテストすべきモジュールを特定するために有力な方法の一つは、数学的予測モデルを用いることである。すなわち、モジュールの規模や複雑度を説明変数、欠陥の有無を目的変数として判別モデルを構築し、これを利用して重点的にテストすべきモジュールを特定することである。判別モデルは結合テスト時に用いられることが前提となっており、モジュールごとに欠陥の有無を判別している。この目的のために、ソフトウェア工学分野では欠陥モジュール判別モデルに関する研究が現在まで多数行われている。

判別モデルに基づいて重点的にテストするモジュールを決定する場合、モデルの予測精度の高低がソフトウェアの品質(ソフトウェアリリース後の障害発生確率)に強く影響する。予測精度が高い場合、欠陥が含まれているモジュールを重点的にテストできることが増え、実際に欠陥を発見できる可能性が高まる。逆に予測精度が低い場合、実際には欠陥が含まれているモジュールに対し、欠陥が含まれていないと予測することが増えるため、それらのモジュールに対するテストが不十分となり、欠陥を見逃してしまう可能性が高まる。

予測精度を高めるため、様々な予測方法や変数選択法などが判別モデルに適用され、評価されてきた。ただし、ある予測方法(ロジスティック回帰など)があるソフトウェア A において欠陥モジュール予測精度が高くても、別のソフトウェア B においては必ずしも判別精度が高くないということが発生する。これは欠陥予測の外的妥当性の問題と呼ばれる。主な原因は、説明変数(ソフトウェアの特徴量)と目的変数(欠陥の有無)の関係が、ソフトウェアによって大きく異なりうるためであると考えられる。例えばあるソフトウェアではソースコードの複雑度(説明変数)と欠陥が含まれる確率(目的変数)の関係が線形であるが、別のソフトウェアでは非線形であるなどである。欠陥予測の外的妥当性の問題については、これまでの研究で多く指摘されている[1][2]。

このため従来研究では、複数のソフトウェアにおいて平均的に判別精度が高い予測方法や変数選択法を実験的に明らかにし、それらを判別モデル構築時に利用することを推奨してきた。ただしこのアプローチに従い、開発中のソフトウェアに対して、推奨される方法を利用してモデルを構築しても、上述の外的妥当性の問題が発生し、判別予測精度が低くなる(欠陥見逃しの確率が高まる)リスクを避けることが困難であった。

2. 研究の目的

研究の目的は、欠陥モジュール判別モデルの利用時に、欠陥予測の外的妥当性の問題が発生し、予測精度が低下するリスクを抑えることである。従来アプローチでは(結合)テスト前にモデルの判別精度を評価していた。それに対し本研究では、テスト中にモデルの判別精度を評価し、精度の高いモデルを動的に選択するというアプローチを取る。テスト中に判別精度を評価するために、バンディットアルゴリズムを利用する。

アルゴリズムの説明には、スロットマシンの比喩がしばしば用いられる。複数の当たり確率の異なるスロットマシンが存在し、100 コインを使って最大の累積報酬を得ることを目的とする(多腕バンディット問題)。最も単純な方法は、どれかのスロットマシンを1つだけ選び、そのマシンに100 コインすべてを使うことである。ただしこの場合、当たり確率の低いスロットを選び、累積報酬が低くなる可能性がある。バンディットアルゴリズムでは、1 コインずつ使い、その結果から各マシンの報酬の期待値を更新していき、それぞれの場合で最も報酬の期待値が高いマシンを選んでいく方法である。

ここで判別モデルをスロットマシンとみなし、モデルの予測精度(欠陥ありと予測されたモジュールで、実際にテスト時に欠陥が発見されるか)を報酬とみなす。図1に判別モデルの選択に用いた例を示す。図ではモデル A, B, C が予測モデルの候補であり、図はテスト対象モジュールのうち t3, t1, t5 をテストした直後を示す。4 つ目のモジュールのテストでは、最も精度が高いモデル C が選択される。

本研究の最も大きな利点は、判別モデルの予測精度評価を繰り返しながら順次モデルを選択するため、精度評価とモデル利用の繰り返しが一定数以上ならば、原理的に低い予測精度にならないことである。例えば、モデル A, B, C が存在し、A, B, C の順に予測精度

テストモジュール	t3	t1	t5	t7	t9	...
欠陥あり/なし	P	N	N			...
モデルAによる予測						
予測	P	N	P	N	N	...
精度評価	TP	TN	FP			...
モデルBによる予測						
予測	N	P	N	P	N	...
テストモジュール	FN	FP	TN			...
モデルCによる予測						
予測	P	N	N	N	N	...
テストモジュール	TP	TN	TN			...

図1 バンディットアルゴリズムによるモデル選択例

- が高い場合、繰り返し回数が増えるほど、バンディットアルゴリズムによる予測精度は C に近づく(A と同程度の低い予測精度とはならない)。この利点より次の(1)~(3)の効果が期待される。
- (1) 欠陥見逃しのリスクを抑えてソフトウェアの品質を高める(リリース後の障害発生確率を低下させる)。
 - (2) これまでのアプローチでは欠陥予測の外的妥当性の問題から、リスクを抑えるために予測方法などを慎重に選定する必要があった。これに対し本研究では予測方法などを一つに絞り込む必要はなく、複数の候補を用意すればリスクを抑えられる。このため、欠陥モジュール判別予測の適用がより容易になり、ソフトウェア開発での予測モデル利用が促進される。
 - (3) 十分に予測精度が評価されていない、新たな予測方法や変数選択方法を、低いリスクで利用できる。すなわち、新たな方法が予測精度を低下させるものであっても、その影響を抑えられ、かつ新たな方法が精度を改善するものであればその効果を得ることができる。

3. 研究の方法

(1) 多様なソフトウェア欠陥予測におけるバンディットアルゴリズムの有用性: ソフトウェア欠陥予測の研究は幅広く行われている。欠陥判別モデルのモデル構築方法や変数選択方法に関する研究だけでなく、例えば(テスト中ではなく)モジュールの修正直後に欠陥の有無を判別する Just-in-time defect prediction に関する研究や、ラーニングデータとテストデータでソフトウェアが異なる(例えば前者ではソフトウェア A、後者ではソフトウェア B)から得られたデータを用いて、欠陥の判別予測を行うクロスプロジェクト欠陥予測に関する研究などがある。これらにおけるバンディットアルゴリズムの有用性と、それぞれの特徴を考慮した、新たなバンディットアルゴリズムの適用方法も明らかにする。

(2) オンライン学習とバンディットアルゴリズムのハイブリット: オンライン学習(予測中に得られたデータでモデルを再構築する)と、バンディットアルゴリズム組み合わせると、予測精度が高まる可能性がある。例えばテスト対象のモジュールが 100 個あるとする。最初の 10 個のテスト結果を用いて高精度の予測モデルを構築することは難しいが、半数の 50 個のテストを終えた段階では、ある程度精度の高いモデルを構築できる可能性がある。そこで、オンライン学習とバンディットアルゴリズムの組み合わせ方法を明らかにする。

(3) ソフトウェアテストにおけるバンディットアルゴリズムの適用範囲: バンディットアルゴリズムは下記の条件を満たす場合に適用可能である。

予測精度評価がモデル利用中に可能であること。すなわち、予測結果に対応する実際の結果が記録されており、かつモデル利用が繰り返し行われること。

最適な予測モデルがモデルの利用中に変化しない。

結合テストの場合、多数のモジュールのテストが順次行われるため が当てはまる。また、テスト対象のソフトウェアは結合テスト中に大幅に修正されないため、ソフトウェアの性質は変わらず が当てはまる。従って、ソフトウェアテストに関連して、複数の手法の中から最適な手法を選ぶ必要がある場合、バンディットアルゴリズムが有効である可能性がある。そこで、テストケース生成やソフトウェア信頼度成長曲線などへのバンディットアルゴリズムの適用可能性を明らかにする。

4. 研究成果

成果は 3 種類に分類される。具体的には(1) 欠陥予測モデルの精度向上、(2) オンライン最適化の欠陥予測以外への適用、(3) オンライン最適化自体の改善である。(1)については変数選択方法、アンサンブル学習方法などの選択に関して、オンライン最適化を行った。(2)についてはコードクローン検出、コード自動生成、ソフトウェアレビューなど対してオンライン最適化を適用した。(3)については、オンライン学習方法の問題点とその改善方法、ソフトウェアテストの再実施方法の 2 つを新たに提案した。これらより、ソフトウェア品質向上が期待される。以降では、上記のいくつかを取り上げて詳説する。

(1) アンサンブル学習方法の選択に対するオンライン最適化

欠陥予測モデルの精度を高めるために、アンサンブル学習が適用される場合がある。本研究では、アンサンブル学習による欠陥予測のモデルの精度向上に取り組んだ。

アンサンブル学習の概要: 予測時に 1 つの機械学習モデルを用いるのではなく、複数の機械学習モデルの予測結果を組み合わせる、アンサンブル学習と呼ばれるアプローチが存在する。最も単純なアンサンブル学習である Hard Voting を用いて説明する。Hard Voting ではあらかじめ複数の機械学習モデル、例えば 3 つのモデルを作成しておき、それらのうちの過半数の予測結果を採用する。具体的にはあるモジュールに対し、2 つ以上のモデルが欠陥ありと予測している場合、そのモジュールの予測を欠陥ありとする。逆に 1 つ以下のモデルしか欠陥ありと予測していない場合、予測を欠陥なしとする。

アンサンブル学習をソフトウェア欠陥予測に適用した研究は数多く存在する[2]。アンサンブル学習として、主に、バギング、ブースティング、スタッキングがあげられる。

アンサンブル学習の問題点: アンサンブル学習において、どのようなデータセット、すなわちソフトウェアに対しても高い精度を示す方法は存在しない。このため予測モデル構築時に、最適、すなわち最も予測精度が高くなるアンサンブル学習方法を選択する必要がある。平均的に予測精度の高い方法は存在しうるが、実際に適用した場合、その方法が必ずしも高精度になる保証は

ない。従って、欠陥予測の精度が低くなるリスクが常に存在することとなる。

バンディットアルゴリズムとアンサンブル学習との差異: バンディットアルゴリズム (BA) は近年、Web ページの最適化などさまざまな場面で利用されている[5]。複数の予測モデルを扱っている点に着目すると、BA とアンサンブル学習は類似している。ただし、BA はオンライン最適化のアプローチを取っており、ソフトウェア開発中のテスト結果を収集・活用することにより予測精度を高めている。すなわち、テストデータ上で最適化を行っているともみなせる。これに対しアンサンブル学習はモデル構築のために学習データのみを用いており、テストデータを扱うことは前提としていない。

実験: BA によるアンサンブル学習方法の選択効果を確認するために実験を行った。実験に用いたデータセットは、NASA Metrics Data Program において収集された KC4 と呼ばれるデータセットであり、モジュールの複雑度や欠陥有無が記録されている。

アンサンブル学習法として、バギング、XGBoost、ブレンディング、ランダムフォレストを用いた。BA として ϵ -greedy 法を用い、パラメータ ϵ は 0, 0.1, 0.2, 0.3 を設定した。予測精度の評価指標として AUC (Area Under the Curve) を用い、交差検証はホールドアウト法により行った。

BA と、各アンサンブル学習法の予測精度を図 2 に示す。実験に用いたデータセットではバギングが最も高く、BA はパラメータに関わらず、2 番目に精度の高いランダムフォレストと同程度であった。このことより、BA により、最適でない方法を選ぶことによる、予測精度の低下を避けることができるといえる。

(2) コード生成方法の選択に対するオンライン最適化

本研究では、バンディットアルゴリズム (BA) に基づき、コード生成ツールをコーディング中に評価することを提案する。提案方法の手順は以下の通りとなる。

選択中のツールを利用してコードを生成する

ツールにより生成されたコードの正確性を判定する

判定されたコードの正確性を記録して蓄積する

選択中のツールの平均正確性が低ければ、他のツールを選択し、手順 2 に戻る

図 3 に提案方法の適用例を示す。選択対象のツールが 2 種類存在し、それぞれの正確性の初期値は 0 とする。最初はランダムに選択される。適用結果が正しくないため、正確性が -1 と評価される。次に事例 2 において、正確性の平均値が高い ChatGPT が選択され、結果が正しいため正確性が 1 と評価される。以降この手順を繰り返す。

提案方法を評価するために、公開されているデータセット[6]を用いて実験を行った。このデータセットでは 164 件のケースが含まれており、それぞれのケースにおいて、コード生成ツールの正確性が記録されている。正確性はテストケースの通過割合に基づいて [0, 1] の値を取る。

コード生成ツールとして、GitHub Copilot (Ver.1.7.4421, 1.70.8099)、Amazon CodeWhisperer (Ver. Nov.22, Jan.23)、ChatGPT の 5 つを BA ($\epsilon=0.1, 0.2, 0.3$) により選択し、ランダムにツールを選択した場合を比較対象 (ベースライン) とした。表 1 に実験データでの各ツールの正確性を示す。

表 2 に提案方法による正確性 (164 回のツール選択後) の平均値を示す。表に示すように、ベースラインよりも正確性が高く、 $\epsilon=0.3$ のときに最も正確性が高かった。また、提案方法による正確性は、2 番目に正確性の高いツール (表 1 GitHub Copilot Ver.1.70.8099) よりも高かった。

(3) オンライン学習方法の問題点とその改善方法

ソフトウェア欠陥予測の精度を高めるために、オンライン学習により予測モデルを構築することが取り組まれている[4]。オンライン学習では新たなケースが追加された場合、そのケースを学習データに加えて予測モデルを再構築することを繰り返す。

ただし、予測結果が陰性の場合に欠陥見逃しが生じ、オンライン学習の予測精度が低下するという問題がある。本研究ではこの問題に対処することにより、オンライン学習による欠陥予測モデルの精度を高める。

欠陥見逃しの影響: 欠陥予測モデルにより、あるモジュールが陰性 (欠陥無) と予測された場合、人的資源を節約するために、そのモジュールに対して少数のテストケースが作成される。その結果、実際には欠陥があるモジュールにおいても欠陥見逃しが生じ、「欠陥無」モジュールとみなされうる。

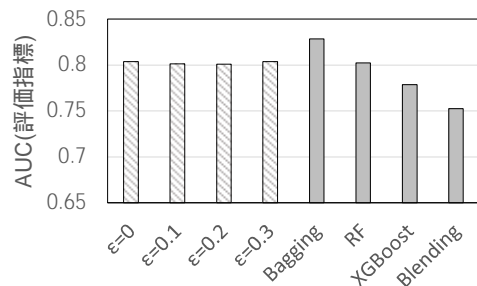


図 2 提案方法と従来方法の予測精度比較

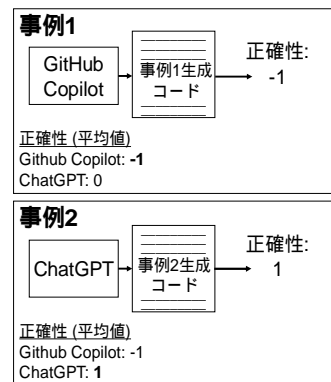


図 3 提案方法の適用例

表 1 各ツールの平均正確性

ツール	GitHub Copilot V.1.7.4421	GitHub Copilot V.1.70.8099	Amazon Code Whisperer Ver. Nov.22	Amazon Code Whisperer Ver. Jan.23	ChatGPT
正確性	0.54	0.60	0.37	0.52	0.78

表 2 提案方法の平均正確性

方法	ランダム	$\epsilon=0.1$	$\epsilon=0.2$	$\epsilon=0.3$	
正確性		0.56	0.63	0.65	0.68

陰性予測による欠陥見逃しが予測モデルに与える影響について説明する。図4において各モジュールは上から順次テストされる。「テスト結果」列はテスト工程で発見された欠陥のみを示し、「テスト後欠陥」はソフトウェアのリリース後など、テスト工程後も含めた欠陥を示す。例ではテストケース数が少ない場合、常に欠陥見逃しが生じるとしている。すなわち図で「予測」が「陰性」の場合、確率100%で「テスト結果」を「欠陥無」としている。

テスト結果に基づき、モジュール t1 と t5 は「欠陥無」とみなされる。t5 のテスト後（すなわち t7 のテスト前）に予測モデル OM が構築される。学習データにはモジュール t1, t9, t5 のデータが含まれ、コード行数などが説明変数、テスト結果が目的変数となる。OM により t7 の欠陥が予測されるが、学習データが正確でないため、OM の予測精度が低くなっている。そのため、本来「陽性」と予測すべき t7 を誤って「陰性」と予測している。

欠陥見逃しの抑制方法: 上記で述べた欠陥見逃しを抑制するために、全モジュールの10%がテストされるまで、すなわち、オンライン学習の初期フェーズの間は、欠陥予測の結果を一律に「陽性」とすることを提案する。提案方法では、例えばテスト対象モジュールが30個ある場合、対象のモジュールは3個となる。図5の「固定予測」列に示すように、これら3個のモジュールの予測結果は「陽性」となる。これにより陰性予測による欠陥見逃しを抑制することができ、テスト結果の正確性を高めることができる。

これは予測結果が「陽性」の場合、多数のテストケースが該当モジュールに適用されるためである。そのため図に示すように、欠陥見逃しを抑制できている。一方で全モジュールの10%において、欠陥が存在しないモジュールに対しても「陽性」とみなす可能性もあり、それらの場合では予測精度が低下する。

提案方法では、全モジュールの10%以上がテストされている場合は、欠陥予測モデルの予測結果を利用する。例えば図5では、モジュール t7 に対してはモデル OM の予測結果を利用する。

実験: 欠陥予測モデルの結果が「欠陥無（陰性）」の場合、欠陥見逃しが100%発生すると想定し、提案方法の適用有無により予測精度がどの程度変化するかを評価した。また参考として、欠陥見逃しの確率が0%の場合の予測モデルの精度についても評価した。

モデルを構築するために、3つのオープンソースソフトウェア（ant, prop, synapse）から収集されたデータを用いた。予測モデルの構築手法として、欠陥予測で広く用いられているロジスティック回帰分析を用いるとともに、correlation-based 変数選択法を適用した。モデルの評価指標として AUC と F1 値を用いた。

表3に予測精度の平均値を示す。表に示すように、提案手法を適用しない場合、予測精度は非常に低くなった。提案手法を適用した場合については、欠陥見逃しが起きないことを想定した場合（欠陥見逃率0%）と比較すると、若干の精度低下が見られた。ただし提案手法を適用しない場合と比較すると、AUC, F1 値ともに3つのデータセット全てで大きく精度が改善した。このため、提案方法はオンライン学習による欠陥予測モデルの精度改善に大きく寄与するといえる。

<引用文献>

- [1] M. D'Ambros, M., Lanza, and R. Robbes, "Evaluating defect prediction approaches: a benchmark and an extensive comparison," Empirical Software Engineering, vol.17, no.4-5, pp.531-577, 2012.
- [2] F. Matloob et al., "Software Defect Prediction Using Ensemble Learning: A Systematic Literature Review," IEEE Access, vol.9, pp.98754-98771, 2021.
- [3] B. Ghotra, S. McIntosh, and A. Hassan, "A Large-Scale Study of the Impact of Feature Selection Techniques on Defect Classification Models," Proc. of International Conference on Mining Software Repositories (MSR), pp.146-157, 2017.
- [4] S. Tabassum et al., "An Investigation of Cross-Project Learning in Online Just-In-Time Software Defect Prediction," Proc. of International Conference on Software Engineering (ICSE), pp.554-565, 2020.
- [5] J. White, Bandit Algorithms for Website Optimization: Developing, Deploying, and Debugging, O'Reilly Media, 2013.
- [6] B. Yetiştir, I. Özsoy, M. Ayerdem and E. Tüzün, "Evaluating the Code Quality of AI-Assisted Code Generation Tools: An Empirical Study on GitHub Copilot, Amazon CodeWhisperer, and ChatGPT," arXiv, 2304.10778, 2023.

表3 提案方法の有無による予測精度

ソフト	欠陥見逃し 0% (参考)		提案 方法無		提案 方法有	
	AUC	F1 値	AUC	F1 値	AUC	F1 値
ant	0.74	0.56	0.55	0.46	0.71	0.53
prop	0.63	0.26	0.49	0.11	0.60	0.24
synapse	0.70	0.61	0.51	0.26	0.64	0.53



図4 欠陥見逃しの例



図5 提案方法による欠陥見逃しの考慮

5. 主な発表論文等

〔雑誌論文〕 計11件（うち査読付論文 11件 / うち国際共著 0件 / うちオープンアクセス 8件）

1. 著者名 NAKASAI Keitaro, KOMEDA Shin Komeda, TSUNODA Masateru, KASHIMA Masayuki	4. 巻 -
2. 論文標題 Measuring Mental Workload of Software Developers Based on Nasal Skin Temperature	5. 発行年 2024年
3. 雑誌名 IEICE Transactions on Information and Systems	6. 最初と最後の頁 -
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 -
1. 著者名 角田 雅照、松本 健一、大岩 佐和子、押野 智樹	4. 巻 41
2. 論文標題 システム構成を考慮したソフトウェア保守種別と工程比率の作業効率への影響分析	5. 発行年 2024年
3. 雑誌名 コンピュータ ソフトウェア	6. 最初と最後の頁 -
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 -
1. 著者名 西浦 生成、廣瀬 椋一、角田 雅照、門田 暁人	4. 巻 41
2. 論文標題 過去と現在進行中のテスト結果を組み合わせたソフトウェアバグ予測	5. 発行年 2024年
3. 雑誌名 コンピュータ ソフトウェア	6. 最初と最後の頁 -
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 -
1. 著者名 KOMEDA Shin, TSUNODA Masateru, NAKASAI Keitaro, UWANO Hidetake	4. 巻 E107.D
2. 論文標題 Prediction of Residual Defects after Code Review Based on Reviewer Confidence	5. 発行年 2024年
3. 雑誌名 IEICE Transactions on Information and Systems	6. 最初と最後の頁 273 ~ 276
掲載論文のDOI（デジタルオブジェクト識別子） 10.1587/transinf.2023MPL0002	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 -

1. 著者名 YAMAGATA Ikuto, TSUNODA Masateru, NAKASAI Keitaro	4. 巻 E107.D
2. 論文標題 The Influence of Future Perspective on Job Satisfaction and Turnover Intention of Software Engineers	5. 発行年 2024年
3. 雑誌名 IEICE Transactions on Information and Systems	6. 最初と最後の頁 268 ~ 272
掲載論文のDOI (デジタルオブジェクト識別子) 10.1587/transinf.2023MPL0001	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

1. 著者名 Toda Koji, Tsunoda Masateru	4. 巻 18
2. 論文標題 Outlier Elimination Technique Using Deletion Imputation Iteration for Fault Prone Module Detection	5. 発行年 2023年
3. 雑誌名 IEEJ Transactions on Electrical and Electronic Engineering	6. 最初と最後の頁 1653 ~ 1663
掲載論文のDOI (デジタルオブジェクト識別子) 10.1002/tee.23889	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 角田 雅照、松本 健一、大岩 佐和子、押野 智樹	4. 巻 64
2. 論文標題 開発体制がソフトウェアプロジェクトのコストと納期に与える影響	5. 発行年 2023年
3. 雑誌名 情報処理学会論文誌	6. 最初と最後の頁 846 ~ 859
掲載論文のDOI (デジタルオブジェクト識別子) 10.20729/00225493	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 角田 雅照、松本 健一、大岩 佐和子、押野 智樹	4. 巻 63
2. 論文標題 ユーザによるソフトウェア価格算出を考慮した技術者単価の推定支援	5. 発行年 2022年
3. 雑誌名 情報処理学会論文誌	6. 最初と最後の頁 960 ~ 972
掲載論文のDOI (デジタルオブジェクト識別子) 10.20729/00217600	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 NAKASAI Keitaro、TSUNODA Masateru、MATSUMOTO Kenichi	4. 巻 E105.D
2. 論文標題 Analyzing Web Search Strategy of Software Developers to Modify Source Codes	5. 発行年 2022年
3. 雑誌名 IEICE Transactions on Information and Systems	6. 最初と最後の頁 31 ~ 36
掲載論文のDOI (デジタルオブジェクト識別子) 10.1587/transinf.2021MPL0004	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

1. 著者名 角田 雅照、松本 健一、大岩 佐和子、押野 智樹	4. 巻 39
2. 論文標題 プログラミング言語の使用割合とソフトウェア開発の生産性との関係	5. 発行年 2022年
3. 雑誌名 コンピュータ ソフトウェア	6. 最初と最後の頁 4_24 ~ 4_30
掲載論文のDOI (デジタルオブジェクト識別子) 10.11309/jssst.39.4_24	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

1. 著者名 MURAKAMI Yukasa、TSUNODA Masateru	4. 巻 E105.D
2. 論文標題 Replicated Study of Effectiveness Evaluation of Cutting-Edge Software Engineering	5. 発行年 2022年
3. 雑誌名 IEICE Transactions on Information and Systems	6. 最初と最後の頁 21 ~ 25
掲載論文のDOI (デジタルオブジェクト識別子) 10.1587/transinf.2021MPL0002	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

〔学会発表〕 計12件 (うち招待講演 0件 / うち国際学会 12件)

1. 発表者名 Sumika Jinnouchi, and Masateru Tsunoda
2. 発表標題 Visualizing Program Behavior with a Ball and Pipes for Computer Science Unplugged
3. 学会等名 Asia-Pacific Software Engineering Conference (APSEC 2023) (国際学会)
4. 発表年 2023年

1. 発表者名 Masateru Tsunoda, Hidetsugu Suto and Takeshi Yamada
2. 発表標題 Toward Enhancing Software Developers' Persuasion and Engagement in Gamification
3. 学会等名 Asia-Pacific Software Engineering Conference (APSEC 2023) (国際学会)
4. 発表年 2023年

1. 発表者名 Yuta Yamasaki, Nikolay Fedorov, Masateru Tsunoda, Akito Monden, Amjed Tahir, Kwabena Ebo Bennin, Koji Toda, and Keitaro Nakasai
2. 発表標題 Software Defect Prediction by Online Learning Considering Defect Overlooking
3. 学会等名 International Symposium on Software Reliability Engineering Workshops (ISSREW) (国際学会)
4. 発表年 2023年

1. 発表者名 Hodaka Shinbori, and Masateru Tsunoda
2. 発表標題 How Does Grit Affect the Performance of Software Developers?
3. 学会等名 Asia-Pacific Software Engineering Conference (APSEC 2022) (国際学会)
4. 発表年 2022年

1. 発表者名 Hiroshi Demanou, Akito Monden and Masateru Tsunoda
2. 発表標題 A Dynamic Model Selection Approach to Mitigate the Change of Balance Problem in Cross-Version Bug Prediction
3. 学会等名 International Workshop on Quantitative Approaches to Software Quality (QuASoQ 2022) (国際学会)
4. 発表年 2022年

1. 発表者名 Ikuto Yamagata, Masateru Tsunoda, and Keitaro Nakasai
2. 発表標題 How Does Future Perspective Affect Job Satisfaction and Turnover Intention of Software Engineers?
3. 学会等名 Asia-Pacific Software Engineering Conference (APSEC 2022) (国際学会)
4. 発表年 2022年

1. 発表者名 Yuriko Takatsuka, and Masateru Tsunoda
2. 発表標題 Preliminary Analysis of the Influence of the Stereotype Threat on Computer Programming
3. 学会等名 Asia-Pacific Software Engineering Conference (APSEC 2022) (国際学会)
4. 発表年 2022年

1. 発表者名 Takuto Kudo, and Masateru Tsunoda, Amjed Tahir, Kwabena Ebo Bennin, Koji Toda, Keitaro Nakasai, Akito Monden, and Kenichi Matsumoto
2. 発表標題 Preliminary Analysis of Review Method Selection Based on Bandit Algorithms, ''
3. 学会等名 Asia-Pacific Software Engineering Conference (APSEC 2022) (国際学会)
4. 発表年 2022年

1. 発表者名 Takeru Tabo, Takuto Kudo, Masateru Tsunoda, Amjed Tahir, Kwabena Ebo Bennin, Koji Toda, Keitaro Nakasai, and Akito Monden
2. 発表標題 Clone Detection Method Selection Based on Bandit Algorithm: a Preliminary Analysis
3. 学会等名 International Workshop on Software Clones (IWSC 2022) (国際学会)
4. 発表年 2022年

1. 発表者名 Takuya Asano, Masateru Tsunoda, Koji Toda, Amjed Tahir, Kwabena Ebo Bennin, Keitaro Nakasai, Akito Monden, Kenichi Matsumoto
2. 発表標題 Using Bandit Algorithms for Project Selection in Cross-Project Defect Prediction
3. 学会等名 International Conference on Software Maintenance and Evolution (ICSME 2021) (国際学会)
4. 発表年 2021年

1. 発表者名 Hirotto Mizutani, Masateru Tsunoda, and Keitaro Nakasai
2. 発表標題 How to Enlighten Novice Users on Behavior of Machine Learning Models?
3. 学会等名 International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2021) (国際学会)
4. 発表年 2021年

1. 発表者名 Masateru Tsunoda, Akito Monden, Koji Toda, Amjed Tahir, Kwabena Ebo Bennin, Keitaro Nakasai, Masataka Nagura, and Kenichi Matsumoto
2. 発表標題 Using Bandit Algorithms for Selecting Feature Reduction Techniques in Software Defect Prediction
3. 学会等名 Mining Software Repositories Conference (MSR 2022) (国際学会)
4. 発表年 2022年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

-

6. 研究組織

氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8 . 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関			
ニュージーランド	Massey University			
オランダ	Wageningen University & Research			