

コンピューティショナル・シンキング・スコアに基づく Scratch ユーザの習熟度到達予測

安東 亮汰^{1,a)} 伊原 彰紀^{1,b)}

受付日 2021年8月3日, 採録日 2022年1月11日

概要: 本研究は、プログラミング学習環境 Scratch におけるユーザが作品制作に使用するコンピューティショナル・シンキング (CT) の概念を明らかにし、ユーザが新たに制作する作品が特定の習熟度に至るか否かを予測するモデル (習熟度到達予測モデル) を提案する。Scratch において作品を 20 回以上制作したユーザ 6,323 人を対象に評価実験を行った結果、習熟度到達予測モデルは適合率 0.87, 再現率 0.87 の精度で特定の習熟度 (Developing) への到達を予測することができ、ユーザが特定の習熟度の作品を制作するためには、CT 概念「同期」や「フロー制御」の習熟が重要であることを明らかにした。

キーワード: Scratch, プログラミング教育, コンピューティショナル・シンキング

Predicting Proficiency Level for Scratch Users Based on Computational Thinking

RYOTA ANDO^{1,a)} AKINORI IHARA^{1,b)}

Received: August 3, 2021, Accepted: January 11, 2022

Abstract: This study analyzes a concept of computational thinking (CT) that users use to create a program work in Scratch programming learning environment, and proposes a proficiency level model that predicts whether a new work created by a user reaches a specific proficiency level. In order to evaluate our prediction model, this study targets 6,323 users who have created more than 20 projects in Scratch. Consequently, the proficiency level model achieves performance in precision (0.87) and recall (0.87) for a specific proficiency level (Developing). Furthermore, this study finds “Synchronization” and “Flow control” as an important concepts to create works with master level proficiency.

Keywords: Scratch, programming education, computational thinking

1. はじめに

初等教育からのプログラミング教育が進められており、プログラミング教材の 1 つとしてビジュアルプログラミングの学習環境 Scratch^{*1}がある。Scratch は、プログラムの命令処理をブロックで視覚的に表現することで、ユーザの直感的な作品制作を実現している。Scratch は、テキストによるプログラミング教育と同様に、ユーザがプログラムの実装を通してコンピューティショナル・シンキン

グ (CT) [1] を習得するために広く利用されている。CT は抽象的な分析と効率的な問題解決のための考え方を含み、特に Scratch におけるユーザの CT を評価するためのツールとして、Moreno-León らは Dr. Scratch [2] を開発している。Dr. Scratch は、作品中のプログラムに使用されるブロックを解析し、作品の機能実装に使用される 7 つの CT 概念をそれぞれ 0 点から 3 点で算出し、合計点数 0 点から 21 点までの 22 段階 (CT スコア) で作品を評価する。特に、CT スコアにより習熟度を 3 つに分け、0 点から 7 点を Basic, 8 点から 14 点を Developing, 15 点から 21 点を Master としている [3]。

¹ 和歌山大学
Wakayama University, Wakayama 640–8510, Japan
^{a)} s216009@wakayama-u.ac.jp
^{b)} ihara@wakayama-u.ac.jp

^{*1} Scratch: <https://scratch.mit.edu/>

Scratch は決められた学習順序が存在せず、ユーザ自らが試行錯誤を繰り返し学習する環境を提供している。したがって、高い CT を必要とする作品の制作に向けて、ユーザが習熟すべき CT 概念を把握することは困難である。学習習熟度の把握を目的とした Yang らの研究では、Scratch におけるユーザは多数の作品制作を経て、使用ブロックの種類が増加することを明らかにしている [4]。この結果から、ユーザは複数回の作品制作を通して、高い CT を必要とする作品制作が可能になることが示唆される。しかし、ユーザが高い CT を必要とする作品を制作するまでに、過去に制作した作品で使用した CT 概念は明らかではない。

本研究では、Scratch において特定の習熟度に到達する作品を制作するまでに、ユーザが過去に制作した作品で使用した CT 概念を明らかにする。さらに、ユーザが過去に使用した CT 概念に基づき、新たに制作する作品が特定の習熟度に至るか否かを予測するモデルを構築する。事前分析として、Scratch においてユーザ 6,323 人が制作した作品 126,460 件 (6,323 人 × 20 件) の特徴分析を行い、その後、Dr. Scratch が算出する CT スコアは、多数の作品制作を実施することで高くなるのか否かを明らかにする。

事前分析の結果から、ユーザが過去に獲得した CT 概念の点数に基づき、新たに制作する作品が習熟度 Developing 以上 (CT スコア 8 点以上)、または習熟度 Master (CT スコア 15 点以上) に到達するか否かを予測するモデルを構築する。本研究により、Scratch においてユーザが高い CT を要する作品を制作するまでに習熟した CT 概念を明らかにすることで、ユーザが有する CT の把握、および、Scratch において公開される作品の中からユーザの CT に合わせた作品の推薦に貢献できると考える。

続く 2 章では、Scratch と関連研究について述べ、本研究の立ち位置を説明する。3 章で予測モデルで使用する説明変数の決定に向けた事前分析について述べる。4 章では習熟度到達予測モデルの構築方法を述べ、5 章で Scratch に記録された作品に基づく予測モデルの評価結果を述べる。6 章で結果の考察を述べ、7 章で本研究をまとめる。

2. Scratch を用いたプログラミング学習

2.1 Scratch

Scratch^{*2}は、MIT メディアラボが開発しているビジュアルプログラミング言語の 1 つであり、ブロックのような視覚的なオブジェクトを組み合わせることで直感的なプログラミングを実現している。テキストベースのプログラミング言語を使用した学習と比較して、Scratch は文字入力の間違いによるエラーが発生しないため学習難易度が低く、プログラミング初学者の学習ツールとして利用されることが多い。従来研究では、Scratch のプログラミングの学習

効果を確認し、テキストベースのプログラミングへの移行を容易にすることを実証実験により明らかにしている [5]。Scratch では、ユーザが制作した作品を Scratch サービス上に公開し、さらに、サービス上に公開済みの他の作品を複製して再利用できる機能「リミックス」を提供している。ユーザはリミックスにより他の作品を複製し、機能の追加や削除を行うことで新たな別の作品として制作することが可能である。また、Dasgupta らは、Scratch においてユーザはリミックスを使用して作品を制作することで、新たな CT 概念を使用した作品制作が可能になることを明らかにしている [6]。

2.2 作品解析ツール：Dr. Scratch

Moreno-León らは、ユーザが制作した Scratch の作品に必要な CT を計測する作品評価ツール Dr. Scratch を開発している [2]^{*3}。Dr. Scratch は、作品内で使用されているブロックやプログラムの構造を解析し、7 つの CT 概念 (抽象化、並列、論理、同期、フロー制御、ユーザ対話性、データ表現) を計測する。表 1 は、各 CT 概念の計測方法を示す。各 CT 概念には、作品中で使用されるブロックの種類、数に基づき、0 点から 3 点の点数が割り当てられ、それら 7 つの CT 概念の点数からユーザが制作した作品を合計点数 0 点から 21 点までの 22 段階 (CT スコア) で評価する。特に、CT スコアにより習熟度を 3 つに分け、0 点から 7 点を Basic、8 点から 14 点を Developing、15 点から 21 点を Master としている [3]。図 1 は、作品 (a) と Dr. Scratch で作品 (a) を評価した結果 (b) を示す。図 1 中の (2) には、作品 (a) を制作するために使用している CT 概念を、抽象化で 1 点、並列で 3 点、論理で 3 点、同期で 3 点、フロー制御で 3 点、ユーザ対話性で 2 点、データ表現で 2 点と算出し、図 1 中の (1) に CT スコア 17 点として習熟度 Master と提示している。個々の作品では、各 CT 概念で 0 点から 3 点のいずれかの点数を獲得するが、Dr. Scratch が定義す

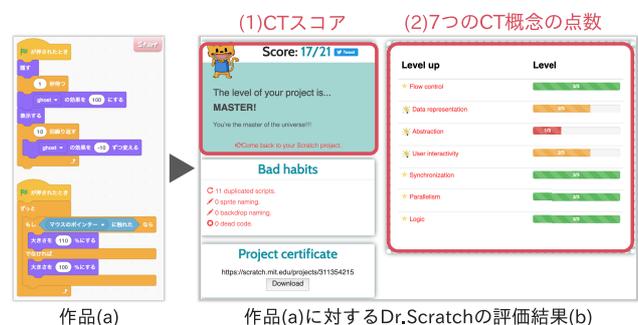


図 1 Dr. Scratch の評価画面 (作品 <https://scratch.mit.edu/projects/311354215> に適用)

Fig. 1 Screenshot of Dr. Scratch analysis result. (e.g., <https://scratch.mit.edu/projects/311354215>)

*2 Scratch: <https://scratch.mit.edu/>

*3 Dr. Scratch: <http://drscratch.org/>

表 1 Dr. Scratch による 7 つの CT 概念の評価方法 [2]
Table 1 Dr. Scratch assessment score for seven CT concepts [2].

CT 概念	0 点	1 点	2 点	3 点
抽象化	-	2 つ以上のスクリプトを使用	定義ブロックを使用	クローンブロックを使用
並列	-	緑の旗ブロックを 2 個以上使用	オブジェクトへのクリック動作により 2 つ以上のスクリプトを同時に 実行する機能を実装	背景が切り替わったとき、 メッセージを受け取ったときなど、 様々なイベント動作により 2 つ以上の スクリプトを同時に実行する機能を実装
論理	-	If ブロックを使用	If else ブロックを使用	論理演算ブロックを使用
同期	-	待機ブロックを使用	メッセージを受信すると プログラムを停止する機能を実装	指定条件を満たすまで プログラムを停止する処理を実装
フロー制御	-	2 個以上の処理ブロックを連結して使用	指定回数/回数無制限の繰返しブロックを使用	指定条件までの繰返しブロックを使用
ユーザ対話性	-	緑の旗ブロックを使用	ユーザによる入力をとまなうブロックを使用	マイクやビデオなどの インタラクティブをとまなうブロックを使用
データ表現	-	オブジェクトの大きさや位置などの プロパティを編集	変数ブロックを使用	リスト変数ブロックを使用

る CT 概念の一部は、同一概念にもかかわらず使用したブロックの種類によって異なる点数を決定している。たとえば、CT 概念の「論理」では、If else ブロックや If ブロックを使用することで 2 点や 1 点を獲得できるが、If ブロック以外 (Not ブロックや And ブロックなど) の論理演算命令でも 3 点を獲得できる。したがって、本研究ではユーザの CT を明確にとらえるため、過去に制作した作品で使用される各 CT 概念の 0 点から 3 点までを区別して調査する。

2.3 従来研究

世界各地で初等教育段階からのプログラミング教育が本格化しており、CT を育むためにはビジュアルプログラミング言語を用いた教育が有効である [7]。特に、Scratch は優れたプログラミング学習環境として世界中の教育機関で利用されており、Scratch の学習効果に関する研究が進められている [4], [8], [9], [10], [11]。

Scratch で制作された作品の特徴を調査するために、Aivaloglou らは Scratch におけるユーザ 109,960 人が制作した公開作品 250,163 件から、各作品に含まれるプログラム、および、各作品を Dr. Scratch によって評価した結果を収集し、公開している [8]。Aivaloglou らは実行可能なプログラムを含む作品 233,491 件を解析した結果、スクリプトの規模は比較的小さく、複雑度も単純であることを明らかにした [12]。

Scratch におけるユーザのプログラミング能力の成長に関する研究として、Yang らは、Scratch で少なくとも 50 回以上の作品制作を行ったユーザ 3,852 人は、繰り返し作品を制作することで、作品に使用するブロックの種類数が増加することを明らかにした [4]。Troiano らは、8th-grade (13 歳から 14 歳) の生徒 19 人を対象に、それぞれの作品制作過程で Dr. Scratch により作品に使用する CT 概念を調査した結果、作品制作を進めるにつれて並列、論理、同期の点数が高くなる生徒は多いが、抽象化、データ表現の点数が高くなる生徒は少ないことを明らかにした [10]。

多くの従来研究は、ユーザが制作した 1 つの作品に使用された CT 概念、または 1 つの作品制作過程において使用される CT 概念の順序を調査し、一部の CT 概念 (抽象化、データ表現など) を使用した作品制作は困難であることを明らかにしている。しかし、ユーザが複数の作品制作を重ねることで使用する CT 概念、および、使用する CT 概念の順序は明らかにされていない。本研究では、ユーザが特定の習熟度に至るまでに作品制作に使用した CT 概念を明らかにし、ユーザが次に制作する作品が特定の習熟度以上の評価を得るか否かを予測するモデルを構築する。続く事前分析では、ユーザが特定の習熟度の評価を得る作品を制作するまでに、過去に制作した作品の特徴を調査する。

3. 事前分析

本章では、Scratch において制作された作品で実装されたプログラムを収集し、ユーザが特定の習熟度に到達するまでに、過去に制作した作品の特徴を調査する。

3.1 Scratch プログラムの収集

Scratch は、2019 年 1 月 3 日のバージョン 3.0 のリリースにおいて、使用可能なブロックを追加するなど大規模なアップデートを実施している。本研究では、ユーザが共通の開発環境で制作された作品を比較するため、バージョン 3.0 をリリースした 2019 年 1 月 3 日から 2020 年 1 月 3 日までに初めて作品公開を行ったユーザを分析対象とする。まず、Scratch が提供する API^{*4} を用いて、20 件以上の作品を制作したユーザ 7,050 人が制作した作品 141,000 件に関するデータを収集した。その後、収集した作品プログラムの調査、および、Dr. Scratch による評価結果の収集を行った結果、Scratch3.0 には存在しないブロック^{*5}を含む作品 (715 件) や Dr. Scratch による評価に失敗した作品

*4 [https://ja.scratch-wiki.info/wiki/Scratch_API_\(2.0\)#GET_.2Fusers.2F.3Cusername.3E](https://ja.scratch-wiki.info/wiki/Scratch_API_(2.0)#GET_.2Fusers.2F.3Cusername.3E)

*5 https://en.scratch-wiki.info/wiki/Experimental_Blocks

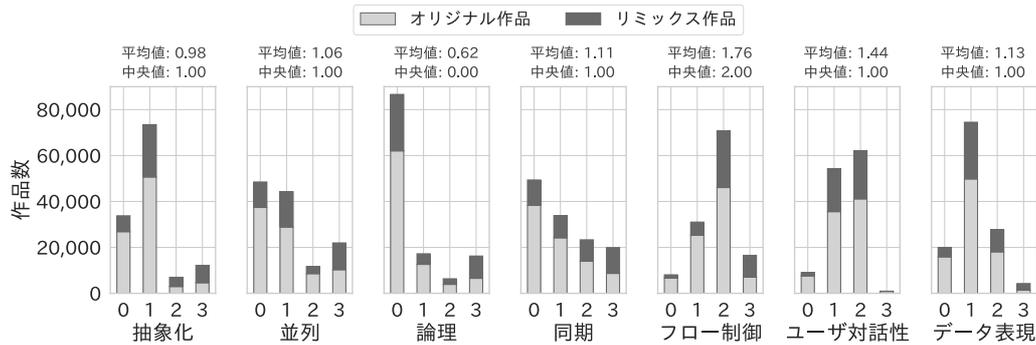


図 2 7つの CT 概念の点数分布

Fig. 2 Distribution of projects for each CT concept assessed by Dr. Scratch.

(322 件) が一部存在したため、それらの作品を制作したユーザ 727 人は分析対象から除外した。最終的に、本研究ではユーザ 6,323 人が 1 番目から 20 番目までに制作した作品 126,460 件を分析対象とする。

3.2 分析対象作品の特徴分析

図 2 は、収集した作品を対象に Dr. Scratch によって計測した 7 つの CT 概念の点数別作品数の分布を示す。各グラフは、横軸に各 CT 概念における点数 [0-3] 点、縦軸に各点数を獲得した作品数を示す。また、グラフ中の灰色で示す箇所はリミックスを使用せずに制作された作品 (オリジナル作品)、黒色で示す箇所はリミックスを使用して制作された作品 (リミックス作品) を示す。さらに、図 3 は、合計点数別作品数の分布を示す。横軸は合計点数、縦軸は各点数を獲得した作品数を示す。灰色、黒色で示す箇所は図 2 と同様である。

図 2 から、フロー制御では高い点数の作品が多い一方で、論理では高い点数の作品が少ないことが分かる。この結果は、Aivaloglou らが公開する作品と同じ結果を示した [8]。

また、図 3 から、CT スコアは 7 点の作品が最頻値であり、8 点の作品以降、CT スコアが高くなるにつれて作品数が減少し、Dr. Scratch が分類する CT の習熟度である Developing (CT スコアが 8 点から 14 点)、Master (CT スコアが 15 点から 21 点) に分類される作品は少ない。各習熟度の作品数は、Basic が 65,976 件、Developing が 47,861 件、Master が 12,623 件である。習熟度による作品規模の違いを図 4 に示す。横軸は 3 種類の習熟度、縦軸はそれぞれの規模を対数軸で示している。それぞれの習熟度間における作品規模は統計的有意な差^{*6}を確認できることから、習熟度 Developing や Master に分類される作品は、ユーザにとって制作困難であることが示唆される。

3.3 ユーザが過去に制作した作品の点数分析

従来研究では、特定の作品 1 件に対する CT の特徴を調査している。本節では、ユーザがオリジナル作品を制作す

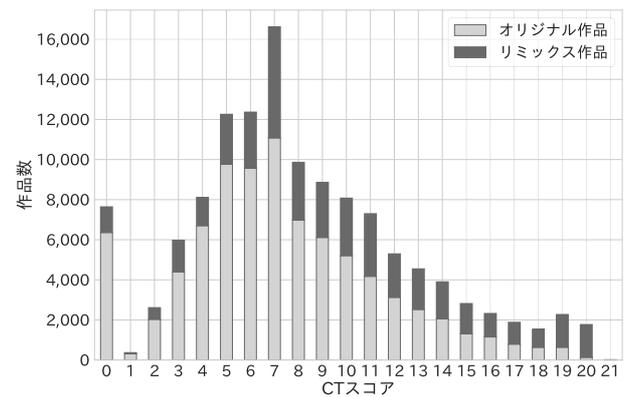


図 3 CT スコアの分布

Fig. 3 Distribution of projects for each CT score.

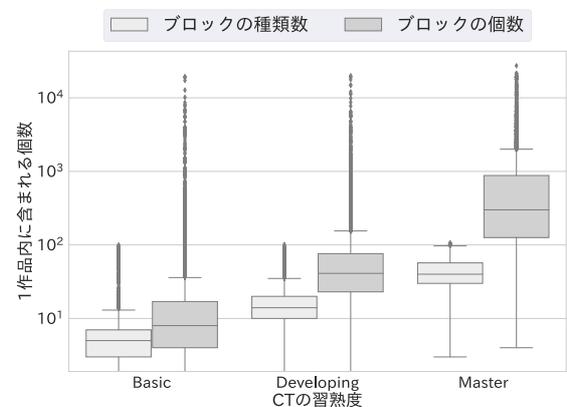


図 4 CT の習熟度別の作品規模の違い

Fig. 4 Number of blocks each proficiency level of CT score.

るまでに公開した複数の作品における CT を調査する。

図 5 は、ユーザが N 番目に CT スコア S (0 ≤ S ≤ 21) 点のオリジナル作品を制作するまでに、過去に制作した作品 (1 番目から N - 1 番目に制作した作品) の CT スコア別作品数の分布をバイオリン図で示す。横軸は N 番目に制作したオリジナル作品の CT スコア、縦軸は 1 番目から N - 1 番目に制作した作品の CT スコアの分布を示す。各バイオリンの膨らみが大きい箇所は、1 番目から N - 1 番目に制作した作品の中で最も獲得した回数が多い合計点数を表す。たとえば、7 点を獲得した作品は、当該作品を制作

*6 マン・ホイットニーの U 検定, 有意水準 5%

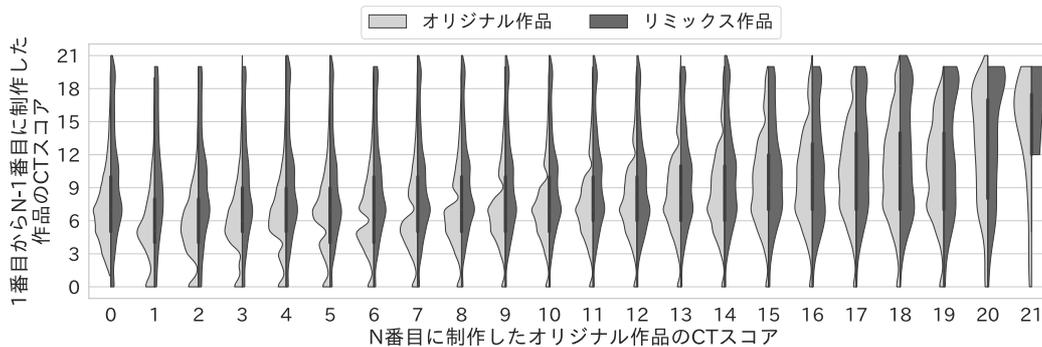


図 5 ユーザが N 番目に CT スコア S ($0 \leq S \leq 21$) 点のオリジナル作品までに制作した作品 (1 番目から $N - 1$ 番目に制作した作品) の CT スコア別作品数の分布

Fig. 5 Number of projects that Scratch users created each CT score ($0 \leq S \leq 21$).

するまでに 5 点の作品を制作した回数が多い。また、左側の灰色の分布は過去に制作したオリジナル作品、右側の黒色の分布は過去に制作したリミックス作品の制作頻度を示す。これらの結果を Dr. Scratch が分類する 3 つの習熟度別に分析結果を述べる。

- **Basic** のオリジナル作品を制作するまでの特徴：CT スコアの中央値が 7 点であり、四分位偏差は 2.0 点である。このことから、ユーザは連続して Basic、または Developing の作品を制作することが多く、Master の作品を制作することは少ない。
- **Developing** のオリジナル作品を制作するまでの特徴：CT スコアの中央値が 7 点であり、四分位偏差は 2.5 点である。このことから、ユーザは Basic のオリジナル作品を制作するまでの特徴と同様に、連続して Basic、Developing の作品を制作することが多く、Master の作品を制作することは少ない。
- **Master** のオリジナル作品を制作するまでの特徴：CT スコアの中央値が 10 点であり、四分位偏差は 3.0 点である。このことから、ユーザは Master の作品を制作するまでに Developing の作品を制作することが多い。図 3 からも Master の作品を制作するユーザは少ないことが分かるように、Master の作品を連続して制作することは少ない。

事前分析の結果から、Scratch を使用した学習において、ユーザは CT スコアが低いほど (特に習熟度 Developing 以下であるほど)、過去に制作した作品の CT スコアは同一点数前後であった。したがって、ユーザは突如点数の高い作品を制作することはなく、類似する CT を使用した作品を連続で制作していることが示唆される。ただし、オリジナル作品とリミックス作品における分布は異なることから、次に制作する作品に影響しているのがオリジナル作品かリミックス作品であるのかは明らかではない。

続く 4 章では、ユーザが過去に制作した作品で使用した CT 概念の特徴量に基づき、ユーザにとって到達が困難な目標習熟度 Developing 以上、または Master の評価を得る

作品を制作するか否かを予測するユーザの習熟度到達予測モデルの構築方法を述べる。また、5 章では予測精度の評価結果を述べる。

4. ユーザの習熟度到達予測モデル

4.1 習熟度到達予測モデルの構築

本研究では、特定の目標習熟度に到達する作品を制作するユーザが、過去に制作した作品に使用する CT 概念を明らかにする。具体的には、ユーザが Scratch において制作する 1 番目から 20 番目までの作品で、目標習熟度に到達するまでに制作した作品の CT 概念を説明変数とし、目標習熟度に到達する作品を制作するユーザか否かを予測する機械学習モデルを構築する。目標習熟度に到達しないユーザは 1 番目から 20 番目までに制作した作品の CT 概念を説明変数とする。3 章の事前分析の結果に基づき、目標習熟度が異なる 2 種類の予測モデルを構築する。

- モデル 1：CT スコアが 8 点以上、すなわち Developing 以上のオリジナル作品を制作するユーザを予測
- モデル 2：CT スコアが 15 点以上、すなわち Master のオリジナル作品を制作するユーザを予測

目的変数は、 N ($2 \leq N \leq 20$) 番目に目標習熟度に到達するオリジナル作品を制作したユーザを正例クラス、それ以外のユーザを負例クラスとする。従来研究 [4] では、ユーザが特定の種類のブロックを 1 回でも使用すれば、そのブロックを学習したととらえていることから、本研究でも同様に、ユーザが目標習熟度を満たす作品を 1 回でも制作すれば到達したと判断する。提案するモデルでは、ユーザが自身で制作したオリジナル作品が目標習熟度に到達することを予測するモデルを構築し、たとえリミックス作品で目標習熟度以上の点数を獲得していても目標習熟度に到達したと判定しない。 $N = 1$ となる正例クラスは、特徴量として使用する作品数が 0 件となるため除外する。

説明変数には、過去に制作した作品における 7 つの CT 概念の 0 点から 3 点の獲得有無を用いる。先行研究では CT 概念の点数獲得回数を用いているが、本研究では説明



図 6 2つのモデルにおける正例クラス、負例クラスの事例

Fig. 6 Example of positive class and negative class on our models.

変数に獲得有無を用いる。特定の習熟度に到達する正例クラスの作品数は到達までに制作した作品数，すなわちユーザ1人あたり1作品から19作品である一方で，負例クラスのユーザは到達できず，ユーザ1人あたり20作品と多くの作品数から説明変数を計測するため，本研究では作品数の偏りを解決するため獲得有無を用いる。獲得点数の計測は，リミックス作品の制作による学習効果が示されているため [6]，オリジナル作品の制作において獲得した点数と，リミックス作品の制作において獲得した点数と区別する。なお，説明変数の計測にリミックス作品を加えることの妥当性については6.3節で考察する。したがって，56次元（7（7つのCT概念）×4（0点から3点）×2（オリジナル作品またはリミックス作品））の説明変数を計測する。

図 6 は，構築する2つのモデルにおける正例クラスと負例クラスの事例を示し，3人のユーザ（a, b, c）が1番目から20番目までに制作した作品事例から，説明変数の計測方法を述べる。

- ユーザ (a) : 1度も Developing 以上のオリジナル作品を制作していないため，どちらのモデルにおいても負例クラスとなり，説明変数の計測には1番目から20番目までのすべての作品を使用する。
- ユーザ (b) : 9番目の作品で Developing のオリジナル作品を制作しているため，モデル1では正例クラスとなり，学習データには1番目から8番目の作品を使用する。一方で，Master のオリジナル作品は1度も制作していないため，モデル2では負例クラスとなり，説明変数の計測には1番目から20番目までのすべての作品を使用する。
- ユーザ (c) : 5番目の作品で Developing のオリジナル作品を制作しているため，モデル1では正例クラスとなり，説明変数の計測には1番目から4番目の作品を使用する。加えて，11番目の作品で Master のオリジナル作品を制作しているため，モデル2においても正例クラスとなり，説明変数の計測には1番目から10番目の作品を使用する。

予測モデルの構築には，3章で使用したデータセットを使用し，表 2 は，2つのモデルにおける正例クラス，負例クラスに分類されたユーザの内訳を示す。

本研究では，習熟度到達予測モデルの構築にランダムフォレスト法 [13] を用いる。ランダムフォレスト法は，与

表 2 正例クラス，負例クラスに該当するユーザの内訳
Table 2 Number of target users on positive/negative class in our models.

	正例クラス	負例クラス
モデル 1	3,436	1,108
モデル 2	1,345	4,800

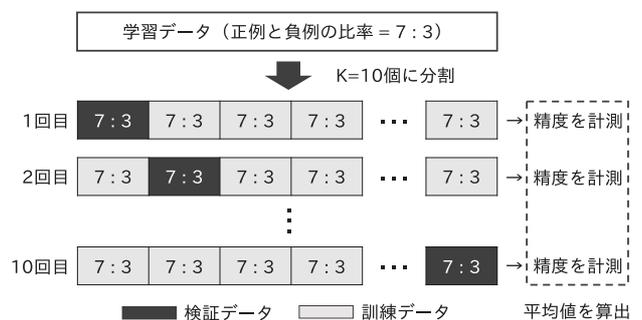


図 7 層化 K-分割交差検証法

Fig. 7 Stratified K-fold cross validation.

えられた訓練データから複数の決定木を作成することでモデルを生成するアンサンブル学習法であり，予測結果から分類精度に強く影響した特徴量を特定できる。ランダムフォレストの実装には Python の scikit-learn を用いる。また，ランダムフォレストのパラメータは Breiman らが推奨する値をもとに，決定木の個数は 200，各決定木の生成に使用する特徴量の個数は説明変数 56 次元の平方根とする。また，予測モデルの検証には層化 K-分割交差検証法を用いる。交差検証法は，データを K 個に分割し，K-1 個を学習データ，残りを検証データとして使用する手法である。図 7 は，K = 10，学習データ内の正例クラスの割合が 70%，負例クラスの割合が 30% のときの層化 K-分割交差検証法の事例を示す。交差検証法には，本研究で用いる層化 K-分割交差検証法以外にも，K-分割交差検証法が存在する。ただし，K-分割交差検証法は，学習データ内のクラス数が不均衡である場合，偏りのある検証データや訓練データを作成し，その結果，学習不足や過学習などの問題を引き起こすことがある。層化 K-分割交差検証法は，図 7 に示すとおり，検証データ，訓練データ作成時の正例クラス，負例クラスの比率を学習データと同じにすることで，K-分割交差検証法が抱える問題を解決している。本研究では，K = 10 として交差検証を行う。また，予測モデルの構築時に使用する正例クラスと負例クラスに属するデータ

表 4 2つのモデルにおける重要度の高い説明変数の上位 10 件

Table 4 Top 10 important explanatory variables on our prediction models.

グループ	モデル 1		モデル 2	
	重要度	説明変数	重要度	説明変数
1	0.05	{オリジナル/フロー制御/0 点}	0.04	{オリジナル/データ表現/0 点}
2	0.04	{リミックス/抽象化/0 点}	0.04	{オリジナル/抽象化/0 点}
3	0.04	{オリジナル/データ表現/0 点}	0.03	{オリジナル/同期/2 点}
4	0.04	{オリジナル/ユーザ対話性/0 点}	0.03	{オリジナル/フロー制御/3 点}
5	0.04	{リミックス/並列/0 点}	0.03	{リミックス/同期/0 点}
6	0.04	{リミックス/データ表現/0 点}	0.03	{オリジナル/論理/0 点} {オリジナル/フロー制御/1 点} {リミックス/論理/0 点}
7	0.04	{リミックス/抽象化/2 点}	0.03	{オリジナル/並列/3 点}
8	0.03	{リミックス/同期/0 点}	0.03	{オリジナル/同期/3 点}
9	0.03	{リミックス/フロー制御/1 点}	0.03	{オリジナル/並列/2 点} {オリジナル/データ表現/2 点}
10	0.03	{リミックス/論理/3 点}	0.03	{オリジナル/論理/2 点}

の割合は、表 2 に示すように偏りが存在するため、モデル構築時に各クラスの出現する割合の逆数に基づいた重みを割り当てる。予測モデルの評価指標は適合率、再現率、F 値の 3 つを用い、交差検証により 10 回の構築を行うことで予測モデルから得た各指標の平均値を算出する。

4.2 モデルに影響する特徴量の重要度

ユーザがオリジナル作品の制作において目標習熟度に到達するまでに、過去に制作した作品の CT 概念の特徴を明らかにするために、モデルへの精度向上に寄与する各説明変数の重要度を算出する。予測モデルの評価では層化 10 分割交差検証を用いて 10 個の予測モデルを構築するため、モデルの構築ごとに算出される各説明変数の重要度が異なる。本研究では、10 個のモデルにおける説明変数の重要度ランキングを明らかにするために、順位傾向のクラスター分析手法 Scott-Knott Effect Size Difference (ESD) 検定 [14] を用いる。Scott-Knott ESD は統計的に複数のスコアをクラスタリングする手法であり、予測モデルに影響を及ぼす説明変数の特定にも使用されている。本研究では、10 個の各予測モデルから得られる説明変数の順位を分類し、いずれのモデルにも強く寄与する説明変数を特定する。Scott-Knott ESD の実装には、R パッケージの ScottKnottESD を用いる。

5. 実験結果

本章は、2つの習熟度到達予測モデル（モデル 1：CT スキルの合計点数が 8 点以上のオリジナル作品を制作するユーザを予測、モデル 2：CT スキルの合計点数が 15 点以上のオリジナル作品を制作するユーザを予測）の精度評価、および、分類精度に寄与する説明変数を明らかにする。

表 3 2つのモデルの分類精度（平均値）

Table 3 Prediction results.

	適合率	再現率	F 値
モデル 1	0.87	0.87	0.87
モデル 2	0.73	0.49	0.58

5.1 習熟度到達予測モデルの分類精度

表 3 は、2つの予測モデルそれぞれの分類精度（適合率、再現率、F 値）を示す。モデル 1 の F 値は 0.87 であり、モデル 2 の F 値は 0.58 であった。モデル 2 の分類精度はモデル 1 に比べて低いため、公開した 20 件の作品の中で 1 度も Master に到達しなかったユーザの一部は、Master に到達したユーザとの間で、過去に使用した CT 概念に違いがないことが示唆される。続いて、高い精度で分類できた 2 つのモデルに寄与する特徴量の重要度を分析し、さらに 6 章でモデルの妥当性を考察する。

5.2 特徴量の重要度

表 4 は、2つの予測モデルそれぞれにおいて、分類に寄与する説明変数の重要度の順位を示す。表中には、分類に強く寄与する説明変数を {作品の種類/CT 概念/点数} のように示す。モデル 2 のグループ 6、グループ 9 に示すように、Scott-Knott ESD 検定の結果、統計的に有意な差がない説明変数は、同一順位として示している。モデル 1 では、説明変数 {オリジナル/フロー制御/0 点}*7 が分類精度に最も寄与し、続いて {リミックス/抽象化/0 点}、{オリジナル/データ表現/0 点} が寄与していることが分かった。また、モデル 2 では、{オリジナル/データ表現/0 点}、{オリジナル/抽象化/0 点}、{オリジナル/同期/2 点} が分類精度に寄与していることが分かった。それぞれのモデル間に

*7 オリジナル作品の制作による CT 概念「フロー制御」での 0 点の獲得有無

における重要度の高い説明変数の違いとして、モデル1では0点のCT概念が70%（10件中、7件）を占めていることに対して、モデル2では1点以上のCT概念が約62%（13件中、8件）を占めていることが分かる。それぞれのモデルに使用したCT概念が特に寄与するクラスについては、考察で議論する。

6. 考察

6.1 モデルの精度に寄与する要因

本研究では、ユーザが過去に使用したCT概念に基づき、新たに制作する作品が特定の習熟度に到達するか否かを予測する機械学習モデルを構築した。モデルの評価実験の結果、Developingに関する予測を行うモデル1の精度はF値が0.87、Masterに関する予測を行うモデル2の精度はF値が0.58であった。特に、モデル1では適合率、再現率ともに0.87となり、高い精度で分類することができた。その一方で、モデル2では適合率0.73、再現率0.49となり、適合率は高かったものの、再現率は適合率に比べて低かった。その理由は2つ考えられる。

- 正例クラスと負例クラスのユーザが使用したCT概念の違い。

表4より、2つのモデルそれぞれにおける重要度の高い説明変数として、モデル1では0点のCT概念が多く、モデル2では2点以上となるCT概念が多かった。まずモデル1において、最も重要度の高かった{オリジナル/フロー制御/0点}を獲得したユーザは、正例クラスに約18%（3,436人中、614人）存在し、負例クラスでは約48%（1,108人中、536人）存在していた。同様に、重要度順位が2位の{リミックス/抽象化/0点}を獲得したユーザは、正例クラスでは約17%（3,436人中、591人）存在し、負例クラスでは57%（1,108人中、633人）存在していた。したがって、モデル1では、Developing以上に到達しないユーザの多くが0点を獲得していることが予測精度に影響を及ぼしているため、習熟度Developingに到達するためには、フロー制御などのCT概念を習熟することが期待される。

次にモデル2において、モデル1と同様に重要度の高い上位2件のCT概念は{オリジナル/データ表現/0点}、{オリジナル/抽象化/0点}であり、どちらも0点であった。一方で、モデル1では出現しなかったオリジナル作品で2点以上を獲得しているCT概念が、モデル2において重要度順位が3位の{オリジナル/同期/2点}と4位の{オリジナル/フロー制御/3点}に出現していた。重要度順位が3位の{オリジナル/同期/2点}を獲得したユーザは、正例クラスでは約69%（1,345人中、927人）、負例クラスでは約46%（4,800人中、2,199人）存在していた。重要度順位が4位の{オリジナル/フロー制御/3点}を獲得したユーザは、正例ク

ラスでは約38%（1,345人中、508人）、負例クラスでは約18%（4,800人中、855人）存在していた。したがって、モデル2では、正例クラスの特徴がモデルの予測精度に影響を及ぼしているため、ユーザは習熟度Masterに到達するためには、2点以上の同期やフロー制御のCT概念を習熟することが期待される。

- 正例クラスと負例クラスの学習に使用した作品数の違い。

モデルの学習に使用する作品数は、正例クラスではN番目に特定の習熟度に到達するまでに制作した作品N-1件であることに対して、負例クラスではユーザが制作した全作品20件である。具体的には、モデル1の正例クラスでは、学習に使用した作品数の中央値は5件、四分位偏差は3.5であり、モデル2の正例クラスにおいて学習に使用した作品数の中央値は11件、四分位偏差は5.5であった。図8、図9は、それぞれモデル1とモデル2の正例クラスにおいて、N番目に特定の習熟度に到達したユーザ数とその予測結果の分布を示す。横軸は、特定の習熟度に到達した作品番号、縦軸は到達したユーザ数を示す。グラフ中の黒色で示す箇所は、正例クラスのユーザのうち正しく予測できた人数、灰色で示す箇所は予測できなかった人数を示す。図8、図9は、それぞれモデル1とモデル2の正例クラスにおいて、N番目に特定の習熟度に到達したユーザ数とその予測結果の分布を示す。横軸は、特定の習熟度に到達した作品番号、縦軸は到達したユーザ数を示す。グラフ中の黒色で示す箇所は、正例クラスのユーザのうち正しく予測できた人数、灰色で示す箇所は予測できなかった人数を示す。図8、図9から、本提案モデルは作品数が少なくして特定の習熟度に到達するユーザほど正しく予測できた。一方で、作品制作数が増加するにつれ、誤った予測結果が増えたのは、本研究の事前分析から、ユーザは突如点数の高い作品を制作することがないため、特定の習熟度に到達する時点を正しく予測できなかったことが原因と考える。今後は特定の習熟度に到達する時点を見積もる手法を検討する。

6.2 プログラミング学習時におけるモデルの活用

本研究において構築した習熟度到達予測モデルは、プログラミングユーザが有するCTの把握、および、Scratchにおいて公開される作品の中から、ユーザのCTに合わせた作品の推薦に貢献できると考える。

Scratchにおいて、ユーザはリミックス機能などを用いて、他の作品を参考にした作品制作を行うことも多い。ユーザがリミックス対象となる作品を選択するまでの手順は、Scratchのサービス上で自然言語による作品の検索を行い、検索単語に紐づいた作品の抽出があげられる。ただし、Scratchの検索結果は、閲覧数の多い作品やリミック

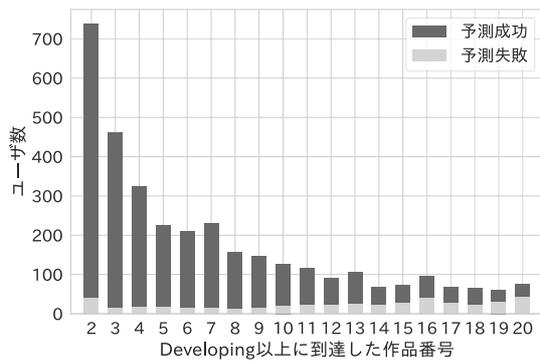


図 8 N 番目に Developing 以上に到達したユーザ数とモデル 1 による予測結果

Fig. 8 Number of users who created project with more than specific proficiency level (Developing) and prediction result in model 1.

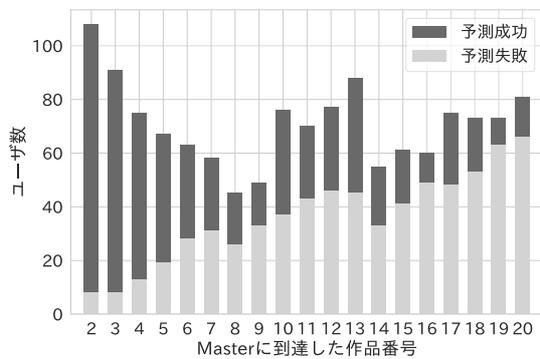


図 9 N 番目に Master に到達したユーザ数とモデル 2 による予測結果

Fig. 9 Number of users who created project with specific proficiency level (Master) and prediction result in model 2.

ス回数の多い作品など、人気度の高い順に表示されるため、ユーザの CT に合った作品が必ずしも上位に表示されるには限らない。この課題の解決策の 1 つとして、Dr. Scratch の評価結果に基づく作品提示方法が考えられる。たとえば、ユーザが習熟度 Basic の作品を制作した場合には、習熟度 Developing の作品を検索結果として提示することで、ユーザは自身の CT に合った作品を見つけることが容易になる。ただし、本研究において、特定の習熟度への到達可否は、ユーザが過去に使用した CT 概念の影響を受けることが明らかになったため、単に習熟度が 1 段階上の作品を提示するだけでは、ユーザの CT に合わせた作品の提示は困難であると考えられる。本研究の結果から、Developing 以上に到達していないユーザの多くは、CT 概念の「抽象化」や「データ表現」などにおいて 0 点を獲得することが示された。このことから、0 点を獲得し続けるユーザに対しては、当該 CT 概念での点数を満たす作品を提示することで、Developing 以上の作品を制作するための支援ができることを考える。一方で、Master に到達するユーザの多くは、CT 概念の「同期」や「フロー制御」で 2 点以上を獲得しているため、これらの概念を使用するユーザはさらに高い習熟

表 5 説明変数からリミックス作品を除去した分類精度 (平均値)

Table 5 Prediction results without using remix-projects in explanatory variables.

	適合率	再現率	F 値
モデル 1	0.88 (*0.87)	0.68 (*0.87)	0.76 (*0.87)
モデル 2	0.55 (*0.73)	0.44 (*0.49)	0.48 (*0.58)

* リミックス作品を除去しない場合の分類精度

度を必要とする作品を制作できることが示唆される。したがって、当該 CT 概念で高い点数を獲得するユーザに対して、高い習熟度の作品を推薦することは、ユーザのさらなる CT 向上につながると思われる。今後は、ユーザが過去に使用した CT 概念に基づく、ユーザの CT に合わせた作品提示を検討する。

6.3 リミックス作品がモデルの精度に及ぼす影響

本研究で構築した習熟度到達予測モデルにおける説明変数には、リミックス作品の制作による学習効果が示されていることから [6]、オリジナル作品で獲得した点数と、リミックス作品で獲得した点数と区別して計測した。しかし特定の習熟度に到達するか否かを判定するためにリミックス作品の制作が有用であるか否かは明らかでないため、リミックス作品を分析対象外とした場合の予測モデルも構築し、精度結果を比較する。

表 5 は説明変数からリミックス作品を除去した場合の分類精度を示す。表 5 から、リミックス作品を除去した場合、除去前と比較して、モデル 1 では F 値が 0.11、モデル 2 では F 値が 0.10 低下することが分かった。したがって、ユーザが Developing 以上の習熟度に到達するか否かを予測するために、リミックス作品を説明変数の特徴量として使用することは予測精度向上に有効であることが示唆される。今後は、特定の習熟度への到達に寄与するリミックス作品の特徴、および、ユーザがリミックスによって習熟する CT 概念を明らかにする。

6.4 妥当性への脅威

- 内的妥当性：本研究では、習熟度到達予測モデルの構築にランダムフォレスト法を用いた。ランダムフォレストのパラメータには、Breiman らが推奨する値をもとに、決定木の個数は 200、各決定木の生成に使用する特徴量の個数は説明変数 56 次元の平方根と定めた。精度向上のために説明変数の選択、対象ユーザの選択、と同時にモデルのパラメータの調整も今後は検討する。
- 外的妥当性：本研究では、ユーザが自身で制作するオリジナル作品が目標習熟度に到達することを予測するモデルの構築を行ったが、オリジナル作品の中には、公開されている他の作品を模倣して制作した作品が存在することも考えられる。また、Scratch のサービス

上に共有されている作品しか調査できていないため、サービス上に非公開の作品を制作していることも考えられる。本研究で対象とするユーザが制作した作品の中にも、これらの事例が含まれる場合、予測モデルの精度低下につながる事が考えられる。

7. おわりに

本研究では、事前分析として、ユーザが制作した作品のCTスコアを調査した結果、ユーザは突如点数の高い作品を制作することはなく、類似のCTを使用した作品を連続で制作していることが示唆された。その後、ユーザが制作する作品を3つの習熟度であるBasic, Developing, Masterに分類し、作品制作に使用するCT概念を分析した。具体的には、ユーザが過去に使用したCT概念に基づき、新たに制作する作品がDeveloping以上、またはMasterの習熟度に到達するか否かを予測する2つの習熟度到達予測モデルを構築した。その結果、Developing以上に関する予測はF値が0.87、Masterに関する予測はF値が0.58の精度で分類することができた。その要因として、目標習熟度に到達するユーザと到達しないユーザの間では使用するCT概念(「同期」や「フロー制御」など)に違いがあることが示唆される。しかし、提案する予測モデルは、初めて目標習熟度に到達するまでに多数の作品制作するユーザを予測することが困難であった。今後は、ユーザが特定の習熟度に到達する時点を見積もる手法を検討し、作品制作数の多いユーザについても考慮した習熟度到達予測モデルを提案する。

参考文献

- [1] Wing, J.M.: Computational thinking, *Commun. ACM*, Vol.49, No.3, pp.33-35 (2006).
- [2] Moreno-León, J., Robles, G. and Román-González, M.: Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking, *RED-Revista de Educación a Distancia*, Vol.15, No.46, pp.1-23 (2015).
- [3] Moreno-León, J. and Robles, G.: Analyze your Scratch projects with Dr. Scratch and assess your computational thinking skills, *Scratch Conference*, pp.12-15 (2015).
- [4] Yang, S., Domeniconi, C., Reville, M., Sweeney, M., Gelman, B.U., Beckley, C. and Johri, A.: Uncovering Trajectories of Informal Learning in Large Online Communities of Creators, *Proc. 2nd Conference on Learning @ Scale (L@S'15)*, pp.131-140 (2015).
- [5] Weintrop, D. and Wilensky, U.: Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms, *ACM Trans. Computing Education*, Vol.18, No.1, pp.1-25 (2017).
- [6] Dasgupta, S., Hale, W., Monroy-Hernández, A. and Hill, B.M.: Remixing As a Pathway to Computational Thinking, *Proc. 19th Conference on Computer-Supported Cooperative Work & Social Computing (CSCW'16)*, pp.1438-1449 (2016).
- [7] 杉浦 学, 松澤芳昭, 岡田 健, 大岩 元: アルゴリズム構築能力育成の導入教育: 実作業による概念理解に基づ

くアルゴリズム構築体験とその効果, 情報処理学会論文誌, Vol.49, No.10, pp.3409-3427 (2008).

- [8] Aivaloglou, E., Hermans, F., Moreno-León, J. and Robles, G.: A Dataset of Scratch Programs: Scraped, Shaped and Scored, *Proc. 14th International Conference on Mining Software Repositories (MSR'17)*, pp.511-514 (2017).
- [9] Robles, G., Moreno-León, J., Aivaloglou, E. and Hermans, F.: Software clones in scratch projects: On the presence of copy-and-paste in computational thinking learning, *Proc. 11th International Workshop on Software Clones (IWSC'17)*, pp.1-7 (2017).
- [10] Troiano, G., Snodgrass, S., Argimak, E., Robles, G., Smith, G., Cassidy, M., Tucker-Raymond, E., Puttick, G. and Hartevelde, C.: Is My Game OK Dr. Scratch?: Exploring Programming and Computational Thinking Development via Metrics in Student-Designed Serious Games for STEM, *Proc. 18th International Conference on Interaction Design and Children (IDC'19)*, pp.208-219 (2019).
- [11] Troiano, G., Chen, Q., Vargas-Alba, Á., Robles, G., Smith, G., Cassidy, M., Tucker-Raymond, E., Puttick, G. and Hartevelde, C.: Exploring How Game Genre in Student-Designed Games Influences Computational Thinking Development, *Proc. Conference on Human Factors in Computing Systems (CHI'20)*, pp.1-17 (2020).
- [12] Aivaloglou, E. and Hermans, F.: How Kids Code and How We Know: An Exploratory Study on the Scratch Repository, *Proc. Conference on International Computing Education Research (ICER'16)*, pp.53-61 (2016).
- [13] Breiman, L.: Random forests, *Machine Learning*, Vol.45, No.1, pp.5-32 (2001).
- [14] Tantithamthavorn, C., McIntosh, S., Hassan, A.E. and Matsumoto, K.: The Impact of Automated Parameter Optimization on Defect Prediction Models, *IEEE Trans. Softw. Eng.*, Vol.45, No.7, pp.683-711 (2019).



安東 亮汰 (学生会員)

2020年和歌山大学システム工学部卒業。現在、同大学大学院博士前期課程に在学中。ソフトウェア工学、特にプログラミング学習支援の研究に従事。



伊原 彰紀 (正会員)

2007年龍谷大学理工学部卒業。2009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。2012年同大学博士後期課程修了。2012年同大学情報科学研究科助教。2018年和歌山大学システム工学部講師。博士(工学)。ソフトウェア工学、特にオープンソースソフトウェア開発・利用支援の研究に従事。電子情報通信学会、ソフトウェア科学会、IEEE各会員。