Predicting Upper Secondary School Students' Programming Self-efficacy in Tobacco Growing Areas of Southwest China Using Decision Tree Analysis

Yanhua Xu
Yuqing Zeng
Zhihua Ai
Chang Wang
Guiyu Wang
Huili Yang

Background：In the field of artificial intelligence, programming self-efficacy plays an indispensable role in the success of programming learning. However, how to predict the level of students' programming self-efficacy has not been addressed. Objective:To predict the level of programming self-efficacy among upper secondary school students in tobacco growing areas of Southwest China, this study used survey data to develop a decision tree model. Methods: First, a total of 512 questionnaires were collected by using the Academic Achievement Test, Creative Style Scale, Programming Learning Attitude Questionnaire, Motivation Scale, Higher-order Thinking Preferences Scale, and Programming Self-efficacy Scale. Secondly, a decision tree model was constructed by SPSS modeler 18.0. Results: The results showed that academic achievement, creativity style, programming learning, motivation, and higher-order thinking propensity were highly predictive of programming self-efficacy. Conclusions: This is the first study in the direction of educational technology and it represents a novel approach to predicting programming self-efficacy among upper secondary school students. The experimental analysis demonstrate that the encouraging results prove the practical feasibility of the approach.

Keywords: Programming Self-efficacy, Tobacco Growing Areas of Southwest China，Predicting, Decision Tree Analysis
Tob Regul Sci.™ 2021;7(5-1):4092-4100
DOI:doi.org/10.18001/TRS.7.5.1.185

In the era of artificial intelligence (AI), programming self-efficacy is crucial and attracts great attention. As an important part of the AI education system, programming education is of great significance to the development of computational thinking. More importantly, computer programming can be used to help develop computational thinking. First, research has found that computational concepts (including variables, loops, and sequencing) can be reinforced through programming. For example, Kazakoff and Bers1 discovered that robot programming had a positive impact on young children's sequencing skills. Second, programming learning could improve students' creativity, and Noh and Lee2 empirically demonstrated that using robots to teach programming can significantly improve students' thinking skills and creativity. In addition, programming learning can effectively promote students' problem-solving skills. Kindergarten students were found to engage in a progressive and iterative approach to solve a series of similar computer programming problems, which facilitates the development of problem-solving and social skills. Thus, computer programming education is an important way to educate students' computational thinking and will be seen as an important driver of sustainable social development in the future.Programming self-efficacy is influenced by many factors.

The effectiveness of programming education and the factors influencing it have increasingly become the target of research in recent years.

Among the influencing factors, programming self-efficacy may be the most important one. Programming self-efficacy was derived from a broader concept of Bandura named self-efficacy3, which means a person's belief in his or her own ability to successfully complete the tasks of computer programming4, also known as ICT self-efficacy or computer self-efficacy. More importantly, self-efficacy directly influences the process of acquiring new things5. Therefore, in the area of programming learning, programming self-efficacy plays a significant role in the success of programming learning6,7. For instance, Tsai found that a student which gets high self-efficacy had better learning outcomes than a student who has low self-efficacy4. Similarly, Baser6 noted that the quality of programming task completion was significantly related to students' attitudes towards programming learning. Furthermore, research has shown that if a student has negative self-efficacy towards programming learning, that student is more likely to fail to complete the programming task8.

In fact, current research on programming self-efficacy has focused on the measurement and investigation of programming self-efficacy8,9, the connection between programming self-efficacy and other variables, and the impact of programming self-efficacy on programming learning effectiveness10. Few scholars have discussed the factors influencing programming self-efficacy. Some experts have found that programming self-efficacy is associated with students' learning experiences, attitudes, learning styles and so on. For example, through a questionnaire survey of 2,421 sixth-grade students, parents, and teachers in 92 elementary schools, Koen collected information on the level of students' ICT self-efficacy and related influences and confirmed these views11. It has also been argued that programming self-efficacy is related to the structure of learning content and programming skills. Askar and Davenport4 for example, explored the factors which may influence programming self-efficacy such as Java among the students studying computer science education and found that programming skills and Java structure had a greater impact on programming self-efficacy. However, these studies are not without flaws. On one hand, Aesaert's sample size is small, which may well explain why there are no significant differences between schools, and it is difficult to draw appropriate conclusions from them11. On the other hand, these studies have a more homogeneous perspective and are less likely to consider students and factors other than programming learning and programming instruction, such as family factors4.

What's more, it is significant to know how to predict programming self-efficacy level based on the above elements. However, few scholars have predicted programming self-efficacy level using decision tree model.

To address the research question, we expect to look to data mining to predict students' programming self-efficacy based on their current status. In educational systems, data mining often uses useful patterns and information to examine and predict student achievement, and the results will support teachers in adapting and providing effective instructional methods12. Commonly used machine learning algorithms such as k-nearest neighbors, decision trees, neural networks, Nave Bayes, and support vector machines13 have been shown that these methods are more suitable for predicting students' academic performance. For example, Chen et al. 14 analyzed students' network usage and academic performance using decision trees, neural networks, and support vector machines and found that network usage can distinguish and predict students' academic performance. Ramaswami et al.15 used the Logistic Regression, Naive Bayes, Random Forest, and k-Nearest Neighbor to fit predictive models of student academic performance.

Considering that decision trees are user-friendly with easy-to-understand rules and a good tolerance for multicollinearity, which is crucial for dealing with complex relationships between predictor variables, we chose decision trees to predict students' programming self-efficacy. It has been noted that the classification decision tree is used when the predictive variable is categorical, and the regression decision tree is suitable for continuous predictive variables 16. For this study, our goal was to identify and predict upper secondary school students' levels of programming self-efficacy in tobacco growing areas of Southwest China, so we used the categorical decision tree algorithm to construct our prediction model.

Yanhua XuCollege of Resource Environment and Tourism Capital Normal University, Beijing, China. Yuqing Zeng ,School of Geography, South China Normal University, Guangzhou, China.Zhihua Ai Shenzhen High School of Science, Shenzhen, China. Chang Wang* Affiliated School of Luohu Education Institute, Shenzhen, China.Guiyu Wang ,No.1 High School of Pidu, Chengdu, China.Huili Yang, Shenzhen High School of Science, Shenzhen, China.Correponding author: Zhihua Ai,Chang Wang. (The two authors share the co-corresponding authorship of this article.)*

METHODS

Participants

This research was completed in an upper secondary school in tobacco growing areas of Southwest China. The school has about 3000 students in their first, second and third year of upper secondary school. 550 students took part in the study and finished questionnaires. Incomplete response cases were removed, reducing the sample size by 515. Of the respondents, 246 (47.77%) were males and 269 (52.23%) were females.

Data Collection and Tools

Invited by researchers from this upper secondary school in the tobacco growing areas of Southwest China, the researchers of this study began data collection. The questionnaire for this study was reviewed by the corresponding author's institution prior to data collection.

This study used a correlation design with a questionnaire as the method of data collection. A QR code consisted of the questionnaire was presented to the students in an online class at the end of the school year, and the students who agreed to participate used their mobile phone to scan the QR code and then answered the questions. In the context of mainland China, QR codes are popularly broadcasted via smartphones, like paying for subscriptions, making payments or opening specific web pages. Although participants completed the questionnaire online, the paper version was also available. Before completing the questionnaire, we obtained permission from these participants and explained the research purpose to them. A brief statement of informed consent and participants' rights was presented prior to completing the questionnaire. Follow-up surveys were initiated only if the participants approved this information. Also, the confidentiality and anonymity of the survey was strictly guaranteed through all research processes.

The questionnaire which was used in this research consisted of six sections: demographic information, geographic academic achievement, Programming Self-efficacy Scale, Creativity Style Scale, Programming Learning Attitude Questionnaire, Motivation for Learning Questionnaire, and Higher-Order Thinking Tendencies Questionnaire. The demographic information section has two parts: gender and geographic achievement (level 1 to level 10, where level 1 is 0 and level 10 is 9). The Programming Self-efficacy Scale, Creativity Style Scale, and Higher-Order Thinking Tendencies Questionnaire were originally developed in English then translated into Chinese for this study. In order to improve the accuracy of the translations, a back-translation method17 was used. First, the instrument was translated from English to Chinese, then the Chinese version was translated into English. The equivalence between the two version was examined. Any inequalities were resolved prior to (solved before) data collection.

Programming self-efficacy scale. Originally developed by Kukul et al.9, the Programming self-efficacy Scale was specifically used to measure the programming self-efficacy of secondary school students. And later, Soykan and Kanbul18 adapted the scale to measure programming self-efficacy of K12 students. It included 31 items, and each item ranges from 1-strongly disagree to 5-strongly agree. In our research, the programming self-efficacy scale's Cronbach's alpha coefficient in this research was 0.983.

The Creativity Style Scale. Developed by Kirton 19, known as the Kirton Adaption-Innovation (KAI) scale, the Creativity Style Scale, was specifically to evaluate whether a person is more inclined to innovate or adapt. It consists of 32 items. It's also a Likert scale from 1 - strongly disagree to 5 - strongly agree. High scorers tend to be innovators and low scorers tend to be adapters. In our research, the creativity style scale's Cronbach's alpha coefficient was 0.943.

Programming Learning Attitude Questionnaire. The Programming Learning Attitude Questionnaire is adapted from the relevant parts of the PISA background questionnaire. It consists of 10 items. It is still a Likert scale from 1 - strongly disagree to 5 - strongly agree. In our research, the programming learning attitude questionnaire's Cronbach's alpha coefficient was 0.846.

Motivation for Learning Questionnaire. The Motivation for Learning Scale was adapted from the Motivation for Learning Scale (MSMT) for secondary school students. In this study, 10 negative items from the original scale were kept and 2 additional items were added to the scale, thus the Motivation to Learn Questionnaire is 12 items. Each question item ranged from "1-strongly disagreeable" to " '5-strongly agree". The total possible scores after addition are from 0 to 12. In our study, the motivation questionnaire's Cronbach's alpha coefficient was 0.676.

Higher-Order Thinking Tendencies Questionnaire. The Higher-Order Thinking Tendencies Questionnaire was developed by Hwang et al. 20 specifically to assess students' higher-order thinking tendencies. The questionnaire was showed using a five-point Likert scale. All of the items raged from "1-strongly disagreeable" to " '5-strongly agree". In our research, the Higher Order Thinking Preference Questionnaire's Cronbach's alpha coefficient was 0.952.

Coding of Key Variables

Based on expert advice, we divided the samples into the groups of high programming self-efficacy and low becoming self-efficacy, using 60% as a point of distinction. Such attribute characteristics will be used as two possible outputs of the predictive model's target tabulations. In terms of predictors, we transformed the variables of creativity style, programming learning attitude, higher order thinking preference, motivation, geographic academic achievement, and gender into binary variables according to the same criteria (Table 1).

Table 1 Variable coding and their descriptive statistics

Variable	Coding	Number	Proportion
Gender	0=female	269	52.23%
	1=male	246	47.77%
Programming self-efficacy	0=low	259	50.29%
	1=high	256	49.71%
Creativity style	0=low	40	7.77%
	1=high	475	92.23%
Programming learning attitude	0=low	320	62.14%
	1=high	195	32.86%
Higher-order thinking tendencies	0=low	55	10.68%
	1=high	460	89.32%
Motivation for Learning	0=low	246	47.77%
	1=high	269	52.23%
Geography academic achievement	0=low	407	79.03%
	1=high	108	20.97%

Data Analysis

The data was analyzed by using SPSS and SPSS Modeler 18.0. SPSS25.0 was used for Pearson's cumulus moment correlation coefficient analysis and SPSS Modeler 18.0 was used for decision tree analysis.

In this study, decision tree which was widely used to predict students' academic performance and learning behavior was adopted14,21 to predict students' performance in programming self-efficacy. The decision tree can be used to create classification rules from training samples and use these rules to classify new samples22. As a top-down chart, it is easy to understand and interpret. Usually, the components of a decision tree model are root nodes, internal nodes, and leaf nodes. The root and internal nodes represent the corresponding test conditions, while the leaf nodes represent the final output. We can infer rules based on the tree structure formed by each node.

We use SPSS modeler 18.0 to perform data mining analysis of students' creativity style, programming learning attitude, higher-order thinking tendencies, motivation for learning, and geography academic achievement. The C5.0 algorithm was chosen for decision tree analysis, which is an improved ID3 algorithm after the C4.5 algorithm proposed by Quinlan23,24 because it is suitable for big data, runs faster, and predicts better25.

In the specific operation, 70% of the sample data (n=361) was used as training data and the remaining 30% (n=154) was used as test data. The applicability of the model constructed from the training data to the new data will be reflected by the test data. Accuracy, precision and recall are important metrics for evaluating the model. Accuracy is the proportion of all samples that are correctly classified. Accuracy is for predictions and it indicates how many samples predicted to be positive are truly positive. Recall is for the actual sample and it indicates how many correctly predicted positive instances are in the sample.

RESULTS

Descriptive and Correlational Analysis

The descriptive statistics for programming self-efficacy, creativity style, programming learning attitude, higher-order thinking tendencies, motivation for learning, and geography academic achievement are presented in Table 2.

The relationships among the six variables were assessed by calculating Pearson product correlation coefficients. The results showed that programming self-efficacy was positively and significantly correlated with creative style, programming learning attitude, higher-order thinking tendencies, motivation for learning, and geography academic achievement. The results are shown in Table 3.

Table 2The Descriptive Statistics of the These Variables

Variable	Full Score	Mean Value	Standard Deviation	60% of the full score
Programming self-efficacy	5	2.96	1.05	210
Creativity style	5	3.57	0.54	475
Programming learning attitude	5	2.80	0.82	195
Higher-order thinking tendencies	5	3.69	0.66	460
Motivation for learning	12	7.49	2.46	269
Geography academic achievement	9	4.10	1.78	108

Table 3Pearson's r of the These Variables

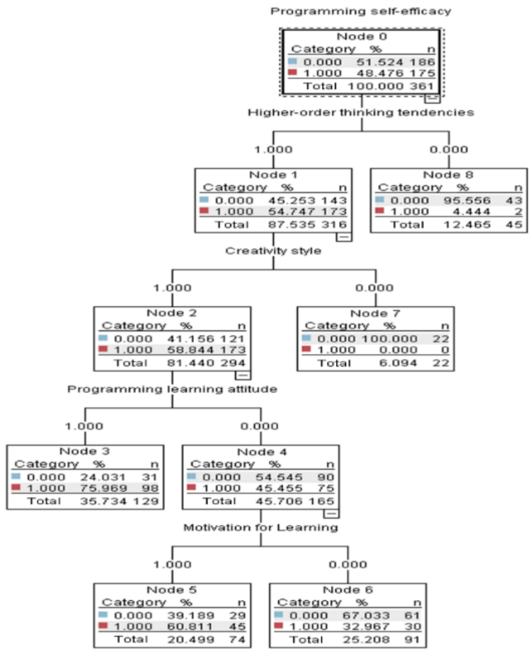
	1	2	3	4	5	6
1 Programming self-efficacy	1					
2 Creativity style	0.321***	1				
3 Programming learning attitude	0.872***	0.375***	1			
4 Higher-order thinking tendencies	0.600***	0.267***	0.752***	1		
5 Motivation for learning	0.369***	0.207***	0.394***	0.261***	1	
6 Geography academicachievement	0.388***	0.218***	0.500***	0.534***	0.134**	1

Note. * p < .05, ** p < .01, *** p < .001

Predicted Results

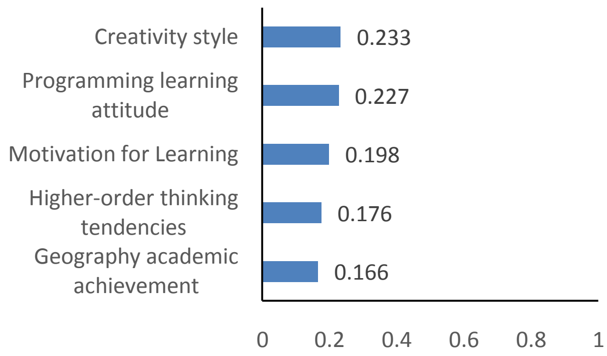
Using the C5.0 algorithm with programmingself-efficacy as the target variable and other variables as input variables, a decision tree model was obtained, as presented in Figure 1.

Figure 1. Prediction model of programming self-efficacy



As shown in the figure 2 below among all the predictors, creativity style was the most important predictor. Programming learning attitude ranked the second most important among the predictors, followed by motivation for learning. Higher-order thinking tendencies and Geography academicachievement also played an important role in the predictive model, ranking fourth and fifth, respectively.

Figure 2. The importance of predictors



TheEvaluationofthePredictionModel

We obtained confusion matrix demonstrated in table 4 and classification accuracy shown in table 5. The model accuracy of testing data set is 74.68%. According to the definition of precision and recall, we calculated the model precision of testing data and the model recall of testing data shown in table6. These three evaluation indicators show that the proposed prediction model work well.

Table 4Confusion matrix

		Predicted class	
		Class=low	Class=high
Actual class of training data	Class=low	126	60
	Class=high	32	143
Actual class of testing data	Class=low	45	28
	Class=high	11	70

Table 5Classificationaccuracy

		Number	Proportion
Training data	Correct	269	76.52%
	Wrong	92	25.48%
	Total	361	
Testing data	Correct	115	74.68%
	Wrong	39	25.32%
	Total	154	

Table 6Recall and Precision of the prediction model

	Recall Rate 1	Precision Rate2
Low programming self-efficacy	66.02%	79.91%
High programming self-efficacy	83.20%	70.76%

1 Recall is TP (true positive) divided by TP (true positive) plus FN (false negative).
2Accuracy is TP (true positive) divided by TP (true positive) plus FP (false positive).

DISCUSSION

DiscussionoftheResults

All the conclusions of this study are derived from the previous literature basis and relevant data analysis results.

First, the fact that programming self-efficacy, creativity style, programming learning attitude, higher-order thinking tendencies, motivation for learning, and geography academic achievement showing significant correlations predict that these elements are, to some extent, mutually influential1–12,18,26–31. Thus, it can be inferred that these factors are predictive of programming self-efficacy.

Second, this study utilizes the decision tree C5.0 algorithm to construct a model which consisted of five-factor to predict the programming self-efficacy and to investigate the contributions of these factors. The evaluation results of the prediction model showed that the model could effectively predict the level of programming self-efficacy with an accuracy of 74.68%, and its recall and accuracy were relatively good.

Third, these results are consistent with the previous research on the factors influencing programming self-efficacy1–12,18,26–31; Previous research on the present study has identified many factors that influence programming self-efficacy, including social factors and individual participant characteristics. Among the individual characteristics, academic achievement, creative style, motivation, attitudes toward programming learning, and higher-order thinking preferences were identified as very important factors. These results are consistent with Pamuk & Peker's4 findings that a positive correlation between programming learning attitudes, motivation, etc. and programming self-efficacy was actually true. This is because the possible sources of self-efficacy include emotional elements such as attitude anxiety. In this context, negative emotions towards programming learning activities inhibit the performance of programming tasks. In other words, low execution of programming learning tasks is perceived as negative, which can lead to an under-performance of programming self-efficacy. Likewise, a positive attitude towards programming learning can also improve self-efficacy.

Implications

Theoretically unique in linking programming self-efficacy with geography academic achievement, creativity style, motivation, programming learning attitudes, and higher-order thinking tendencies, this study deepens the understanding of the factors influencing programming self-efficacy. Furthermore, the predictive effects of creativity style, attitudes toward programming, motivation, higher-order thinking preferences, and geography achievement found in this study suggest that upper secondary school students' changing creativity style, attitudes toward programming, motivation, and geography achievement all contribute to improving their programming self-efficacy. In a practical sense, the relationship between the five pairs of variables presented in this study into self-efficacy may help practitioners better understand the mechanisms of programming self-efficacy and thus be better prepared to help upper secondary school students improve their programming and adapt to the age of artificial intelligence.

Limitations and Future Directions

There are several limitations to this research. First, a cross-sectional design was uesed. Second, all participants were from one upper secondary school in tobacco growing areas of Southwest China, which may have weakened the generalizability of the findings. A longitudinal survey design should be used for the future researchers. In addition, the sample should be more diverse. Moreover, the dimensions of programming self-efficacy which are being influenced by which of the above factors, etc. should be taken into account. Nonetheless, the present study provides a new pathway for the prediction of programming self-efficacy. Future work is necessary to investigate how the mechanisms of programming self-efficacy can be formed to benefit uppersecondary school students' learning, academic efficiency, and the age of artificial intelligence. Whether the predictions are good in both tobacco-growing and non-tobacco-growing areas is also something to think about subsequently. In addition, factors such as the level of civilization in tobacco growing areas need to be further considered.

Conclusion

This study used a decision tree algorithm to predict upper secondary school students' programming self-efficacy in tobacco growing areas of Southwest China. The results showed that creativity style, attitudes toward programming learning, motivation, higher-order thinking preferences, and geographic academic achievement were relatively accurate in predicting programming self-efficacy in upper secondary school students. In addition, the recall and accuracy of the predictive effects of these factors on programming self-efficacy were also well. It is important to note that although these factors were somewhat predictive of upper secondary school students' programming self-efficacy, they were only a small part of the mechanisms influencing programming self-efficacy.

DECLARATION OF CONFLICTING INTERESTS

The authors declare no conflicts of interest.

FUNDING

This research was funded by the National Educational Information Technology Research Project (Practical Research on the Microlecture Teaching Strategy of High School Geography), grant number 143641937.

REFERENCES

1. Kazakoff E, Bers M. Programming in a Robotics Context in the Kindergarten Classroom: The Impact on Sequencing Skills. Journal of Educational Multimedia & Hypermedia. 2012;21(4):371-391.
2. Noh J, Lee J. Effects of robotics programming on the computational thinking and creativity of elementary school students. Educational Technology Research and Development. 2019;68(1).
3. Tsai MJ, Tsai CC. Information searching strategies in web-based science learning: the role of internet self-efficacy. Innovations in Education & Teaching International. 2003;40(1):43-50.
4. Askar P, Davenport D. AN INVESTIGATION OF FACTORS RELATED TO SELF-EFFICACY FOR JAVA PROGRAMMING AMONG ENGINEERING STUDENTS. Online Submission. 2009;8(1):26-32.
5. Erman, Yukselturk, Serhat, Altiok. An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. British Journal of Educational Technology. 2017.
6. Baser M. Attitude , Gender and Achievement in Computer Programming. Middle East Journal of Scientific Research. 2013;14(2):248-255.
7. Marakas GM, Yi MY, Johnson RD. The Multilevel and Multifaceted Character of Computer Self-Efficacy: Toward Clarification of the Construct and an Integrative Framework for Research. Information Systems Research. 1998;9(2):126-163. doi:10.1287/isre.9.2.126
8. Ramalingam V, Wiedenbeck S. Development and Validation of Scores on a Computer Programming Self-Efficacy Scale and Group Analyses of Novice Programmer Self-Efficacy. Journal of Educational Computing Research. 1998;19(4):367-381.
9. Kukul V, Gkearslan A, Günbatar MS. Computer

programming self-efficacy scale (CPSES) for secondary school students: Development, validation and reliability. Eğitim Teknolojisi Kuram ve Uygulama. 2017;7(1):158-158.

10. Brown J. Investigating Collaborative Self-Modeling and its Impact on Introductory Programming Self-Efficacy. In: IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2009, Corvallis, OR, USA, 20-24 September 2009, Proceedings. ; 2009.
11. Aesaert K, Braak JV. Exploring factors related to primary school pupils' ICT self-efficacy: A multilevel approach. Computers in Human Behavior. 2014;41(dec.):327-341.
12. Austin HS. Predictors of Pascal programming achievement for community college students. In: Sigcse Technical Symposium on Computer Science Education. ; 1987.
13. Shahiri AM, Husain W, Rashid NA. A Review on Predicting Student's Performance Using Data Mining Techniques. Procedia Computer Science. 2015;72:414-422.
14. Chen J, Feng J, Sun X, Wu N, Chen S. MOOC Dropout Prediction Using a Hybrid Algorithm Based on Decision Tree and Extreme Learning Machine. Mathematical Problems in Engineering. 2019;2019(1):1-11.
15. Ramaswami G, Susnjak T, Mathrani A, Lim J, Garcia P. Using educational data mining techniques to increase the prediction accuracy of student academic performance. Information and Learning Sciences. 2019;ahead-of-print(ahead-of-print).
16. Vlm A, Af B, Pjvg B, As B. Early segmentation of students according to their academic performance: A predictive modelling approach. Decision Support Systems. 2018;115:36-51.
17. Brislin RW. Back-Translation for Cross-Cultural Research. Journal of Cross-Cultural Psychology. 1970;1(3):185-216. doi:10.1177/135910457000100301
18. Soykan F, Kanbul S. Analysing k12 students' self-efficacy regarding coding education. 2018.
19. Kirton, Michael. Adaptors and Innovators: A Description and Measure. Journal of Applied Psychology. 1976;61(5):622-629.
20. Hwang GJ, Lai CL, Liang JC, Chu HC, Tsai CC. A long-term experiment to investigate the relationships between high school students' perceptions of mobile learning and peer interaction and higher-order thinking tendencies. Educational Technology Research and Development. 2017.
21. Asif R, Merceron A, Ali SA, Haider NG. Analyzing undergraduate students' performance using educational data mining. Computers & Education. 2017;113(oct.):177-194.
22. Mucherino A, Papajorgji PJ, Pardalos PM. Introduction to Data Mining. In: Data Mining in Agriculture. Vol 34. Springer Optimization and Its Applications. New York, NY: Springer New York; 2009:1-21. doi:10.1007/978-0-387-88615-2_1
23. Quinlan JR. Induction of decision trees. Mach Learn. 1986;1(1):81-106. doi:10.1007/BF00116251
24. Quinlan JR. C4.5: Programs for Machine Learning. C4.5: Programs for Machine Learning; 1993.
25. Xiong, Ping. Data Mining Algorithms and Clementine Practice. Beijing: Tsinghua University Press; 2011.
26. Akcaoglu M. Learning problem-solving through making games at the game design and learning summer program. Educational Technology Research & Development. 2014;62(5):583-600.
27. Bandura A. Self-efficacy: Toward a unifying theory of behavioral change. Psychological Review. 1977;84(2):191-215. doi:10.1037/0033-295X.84.2.191
28. Jacobson CM. Cognitive Styles of Creativity: Relations of Scores on the Kirton Adaption-Innovation Inventory and the Myers-Briggs Type Indicator among Managers in USA: Psychological Reports. 2016;72(3_suppl):9.
29. Pajares F. Motivational role of self-efficacy beliefs in self-regulated learning. 2008.
30. Pamuk S, Peker D. Turkish pre-service science and mathematics teachers' computer related self-efficacies, attitudes, and the relationship between these variables. Computers & Education. 2009;53(2):454-461.
31. Weinberg AE. Computational thinking: An investigation of the existing scholarship and research. Dissertations & Theses - Gradworks. 2013.