Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

A Review of Learner's Model for Programming in Teaching and Learning

Hafizul Fahri Hanafi[1,*], Abu Zarrin Selamat[2], Miharaini Md Ghani[3], Wan Azani Mustafa[4], Mohd Fauzi Harun[5], Fatin Hana Naning[6], Miftachul Huda[7], Ahmed Alkhayyat[8]

[1] Department of Computing and Digital Technology, Faculty of Computing and Meta-Technology, Universiti Pendidikan Sultan Idris, 35900, Tanjong Malim, Perak, Malaysia
[2] Department of Moral, Civic and character building, Universiti Pendidikan Sultan Idris, 35900, Tanjong Malim, Perak, Malaysia
[3] Modern Language and Communication, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia
[4] Malaysia Advanced Computing Engineering (AdvComp), Centre of Excellence (CoE), Universiti Malaysia Perlis (UniMAP), Pauh Putra Campus, 02600 Arau, Perlis, Malaysia
[5] Department of Creative Multimedia, Faculty of Art, Computing and Creative Industry, Universiti Pendidikan Sultan Idris, 35900, Tanjong Malim, Perak, Malaysia
[6] Science and Technology Department, Faculty of Humanities, Management and Science, Universiti Putra Malaysia Bintulu Sarawak Campus,97008 Bintulu, Sarawak, Malaysia
[7] Department of Islamic Study, Faculty of Human Sciences, Universiti Pendidikan Sultan Idris, Tanjung Malim, Perak, Malaysia
[8] Faculty of Engineering, The Islamic University Najaf, Iraq

ARTICLE INFO

Article history:
Received 29 March 2023
Received in revised form 17 July 2023
Accepted 3 August 2023
Available online 17 November 2023

ABSTRACT

Over recent years, computer science (CS) teachers and instructors have faced several challenges in helping students strengthen their understanding of programming. The existing assessment methods could be more effective in assessing students' programming skills and knowledge, thus requiring a review of issues surrounding the instruction of programming courses. Against this backdrop, the authors systematically reviewed the current literature to identify several socio-cognitive factors that can help develop a learner model for learning programming. Specifically, the Systematic Reviews and Meta-Analyses (PRISMA) technique was utilized to identify and select relevant articles from three primary online databases: Scopus, Web of Science, and Eric. Initially, 401 relevant papers were identified and retrieved, further reduced to only 24 articles based on specific selection criteria. As revealed, several demographic factors (such as gender, age, ethnicity, and socioeconomic status) and socio-cognitive factors (motivation, attitude, and interest) have been shown to impact student learning of programming significantly. The authors' findings from the systematic literature review helped synthesize the essential elements of the learner model that must be carefully considered and utilized. Arguably, the use of such a new learner model can compel instructors to teach programming more effectively by clarifying several students' socio-cognitive backgrounds, which collectively have a significant impact on student learning of programming courses or subjects at the primary, secondary, and tertiary levels of education, especially in the Malaysian educational context.

Keywords:
Learner's; model; programming; teaching; learning

* *Corresponding author.*
E-mail address: hafizul@fskik.upsi.edu.my

https://doi.org/10.37934/araset.33.3.169184

1. Introduction

Recently, academic administrators have had to reassess the design of student assessment procedures as the scope and nature of educational programs continues to grow and change. This understandably brings into question the validity of existing testing and examination practices. For instance, computer science (CS) educators face multiple difficulties in helping students deepen their comprehension of programming. Consequently, more work is being done to enhance computer science schooling overall. Typically, the standard evaluation method in programming courses involves scoring students' projects and summative project-level assignments to gauge their performance. However, this approach alone cannot improve the appetency of programming language education[1]. Furthermore, the need for and interest in learning how to evaluate computational thinking (CT) skills have grown in tandem with the increased focus on CT in education [2].

A crucial component of CS teachers' competence is being able to recognize and correct students' misconceptions [3]. Thus, it is hardly surprising to observe students learning introductory programming courses asking many questions about the subject matter, suggesting they lack programming skills, which may explain the high rates of failure occurring every semester [4]. Admittedly, from the teaching perspective, learning programming entails effective instructions and regular practices. It is, therefore, imperative that instructors play an active role in assisting students to enhance their learning abilities by providing prompt feedback such that students can learn from their mistakes and re-evaluate their knowledge as necessary [5]. The programming skills must be emphasized that it is not simple to teach beginners or fresh students programming courses [6], as the nature of programming, the characteristics of the students, and the conventional teaching strategies applied by the instructors of basic programming courses present many limitation. In order of that, a promising strategy to deal with these issues is using a blended learning [7]. The students will be able to finish a range of programming assignments with varied levels of complexity by doing this. Students require regular, personalized feedback to strengthen their programming skills. Instructors or lecturers can provide individualized feedback using traditional methods, but evaluating the programming solutions crafted by numerous students can represent a substantial workload for them. As a result, they may give students inconsistent feedback, particularly for lengthy or complex programming solutions. Furthermore, the automated assessment tools powered by artificial intelligence may help ease this burden on instructors. These tools can review student code submission test cases, check for errors, and provide customized feedback unique to each student's work. Implementing automated assistants can make programming course feedback more consistent, allowing instructors to focus on higher-level instructional tasks. Overall, a blended elements of human and Artificial intelligence assessment could significantly enrich the programming and the learning experiences [8].

In addition for developing an effective learner model for programming education demanded a serious and continuous journey to additional demographic and socio-cognitive factors that may influence student learning paradigms and outcomes [9]. Certainly, the further studies for tracking cohorts of students across educational levels would also allow for valuable insights into how learner necessitates developing. The quantitative and qualitative methods should be applied to gather robust data on the multifaceted relationships between student backgrounds, attitudes, motivations, and programming competencies [10]. With a comprehensive interpretation of the learner, instructors can continually refine pedagogical guidelines to optimize programming instruction [11].

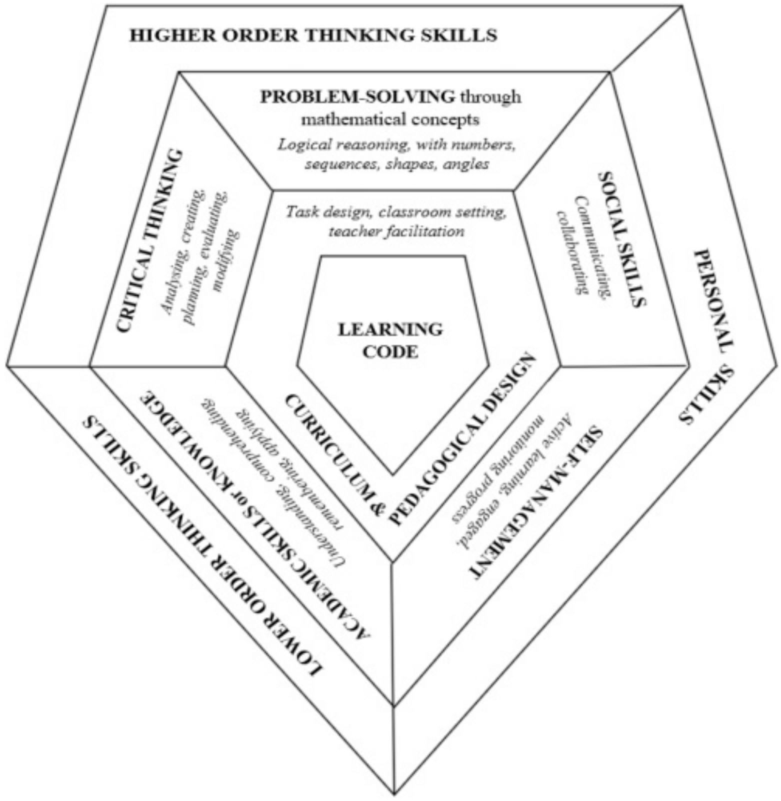

Fig. 1. Model of coding conditioning learning outcomes [9]

In addition, for students to learn programming courses effectively, they must have the necessary communication, teamwork, and self-management capabilities [9]. Particularly, the ability to use computational thinking (CT) is regarded as being crucial for assisting students with a variety of computationally challenging activities. The pedagogical goal of CT in elementary schools is uncertain in this aspect, though, as some countries' present curricula appear to place little emphasis on CT [10]. Indeed, it is accurate to assert that the acquisition of proficient critical thinking (CT) skills requires a prolonged journey of self-guided education, empirical exploration, and refinement via diverse programming assignments. Consequently, educators have to seek out and implement efficacious pedagogical strategies to facilitate students' acquisition of this proficiency [11].

2. Methodology

2.1 Identification

In choosing several appropriate papers for this report, the systematic review process consists of three main phases. The first step is keyword recognition and the quest for linked, similar terms based on the thesaurus, dictionaries, encyclopedia's', and previous studies. Accordingly, search strings on three main databases, Scopus, Web of Science, and Eric (see Table 1), were created after all the relevant keywords were determined. In the first step of the systematic review process, the present research successfully retrieved 401 papers from all databases.

Table 1
The search strings

Scopus **Search Date: December 2022**	TITLE-ABS-KEY (learners AND model* AND programming* AND (teaching* AND learning*)) AND (LIMIT-TO (PUBSTAGE , "final")) AND (LIMIT-TO (PUBYEAR , 2022) OR LIMIT-TO (PUBYEAR , 2021)) AND (LIMIT-TO (DOCTYPE , "ar")) AND (LIMIT-TO (SUBJAREA , "COMP")) AND (LIMIT-TO (LANGUAGE , "English")) AND (LIMIT-TO (SRCTYPE , "j"))
Web of Science	learners AND model* AND programming* AND (teaching* AND learning*) Filters Applied: Web of Science, Journals, 2021-2022 Refined By: Open Access, Publication, Years: 2022 or 2021, Document Types: Article, Languages: English
ERIC	learners AND model* AND programming* AND (teaching* AND learning*) Refined By: Open Access, Publication, Years: 2022 or 2021, Document Types: Article, Languages: English

2.2 Screening

The above-mentioned publications were screened as part of the systematic review process's second stage according to a set of criteria that concentrated on reviews, meta-synthesis, meta-analyses, books, book series, chapters, and conference proceedings, leading to the exclusion of 351 article papers. In other words, the researchers were able to restrict the number of papers initially chosen to 50 papers by using the exclusion and inclusion criteria, including publication language (papers published in English) and timeline (from 2021 to 2022). When the field or discipline criterion was used, in this example computer science or IT, the number was further reduced to 24.

2.3 Eligibility

As was previously mentioned, the primary goal of the screening process was to eliminate redundant or duplicate content. Overall, 401 publications were assessed for this study using a few criteria, including inclusion and exclusion criteria. Additionally, the researchers chose not to focus on secondary resources, which have little empirical datasets, but solely on publications based on main resources. The third stage of the process, eligibility, helped find 44 publications that were deemed pertinent to the study's goals based on the inclusion criteria applied. Accordingly, 377 publications or articles had to be excluded because their titles and abstracts did not relate to the goal of the study, as determined by the empirical data criterion. After further review, 24 articles were chosen (refer to Table 2).

Table 2
Selection criterion of searching

Criterion	Inclusion	Exclusion
Language	English	Non-English
Timeline	Between 2021 – 2022	< 2021
Literature type	Article	Review, Conference and Note
Subject Area	Learner's, model, programming, teaching, and learning	Besides Learner's, model, programming, teaching, and learning

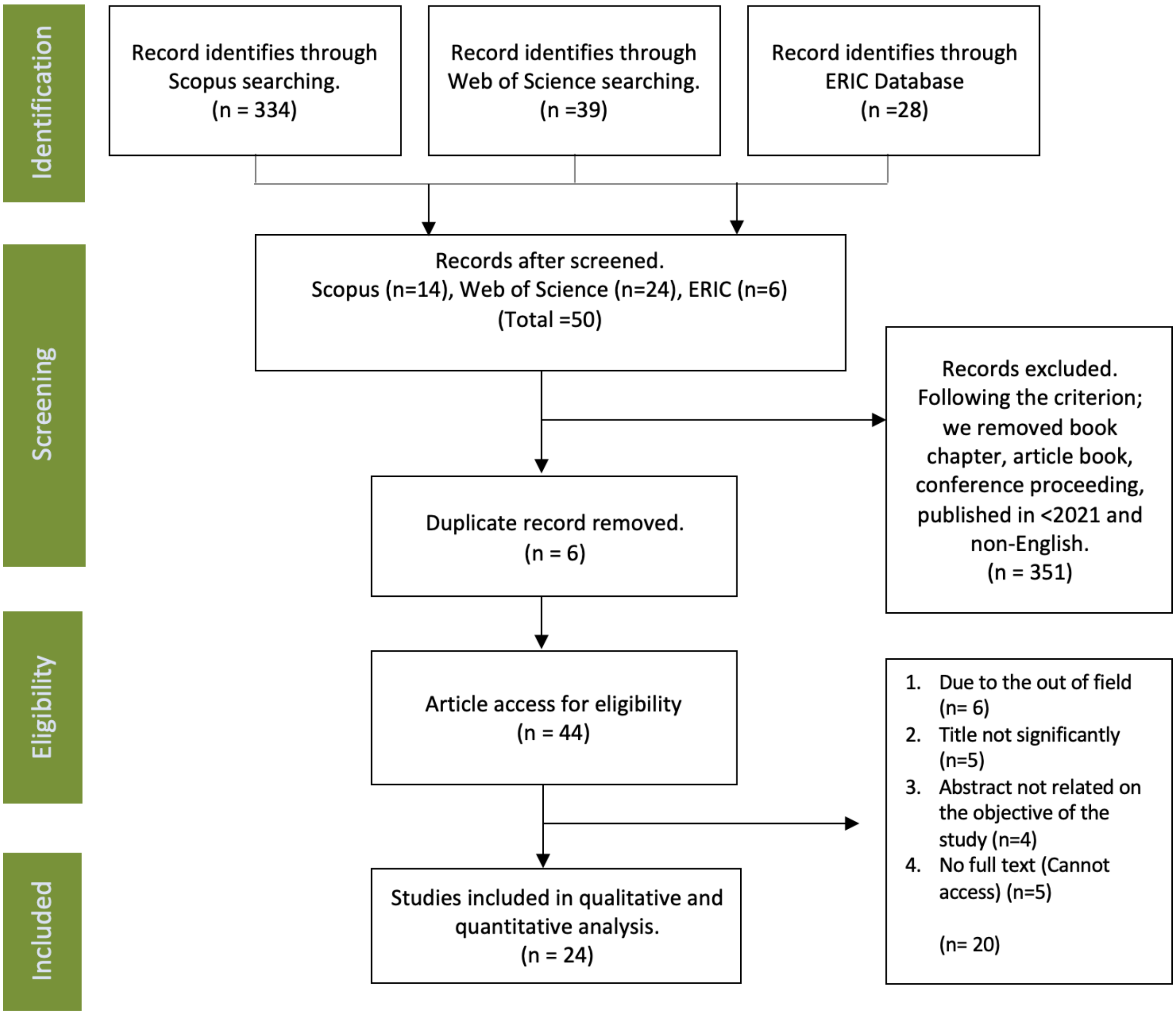

Fig. 1 The flow diagram of the PRISMA review of learner models for the teaching and learning of programming

3. Theme

The research findings of this study indicate two main themes: Learner models for programming learning and programming learner's learning experience, the detailed discussions of which are highlighted as follows:

3.1 Learner's Model in Programming

Learning computer programming courses online can be difficult and stressful for many students, which has been made worse by the COVID-19 pandemic. There are certain socio-cognitive challenges associated with learning programming online that may be distinguishable from those related to conventional learning [12]. To ensure academic success, it is essential to have learning materials that are systematically organized and easily accessible for learners to self-assess their progress [13]. Furthermore, the reason for the learning material that is especially true considering the widespread adoption of online platforms in education. However, it could be challenging for learners to lead themselves in appropriately evaluating their learning outcomes without the assistance of a competent mentor. This article also highlights how students could benefit programming skills using Quizzes, an adaptive platform, to self-evaluate their programming course progress [14].

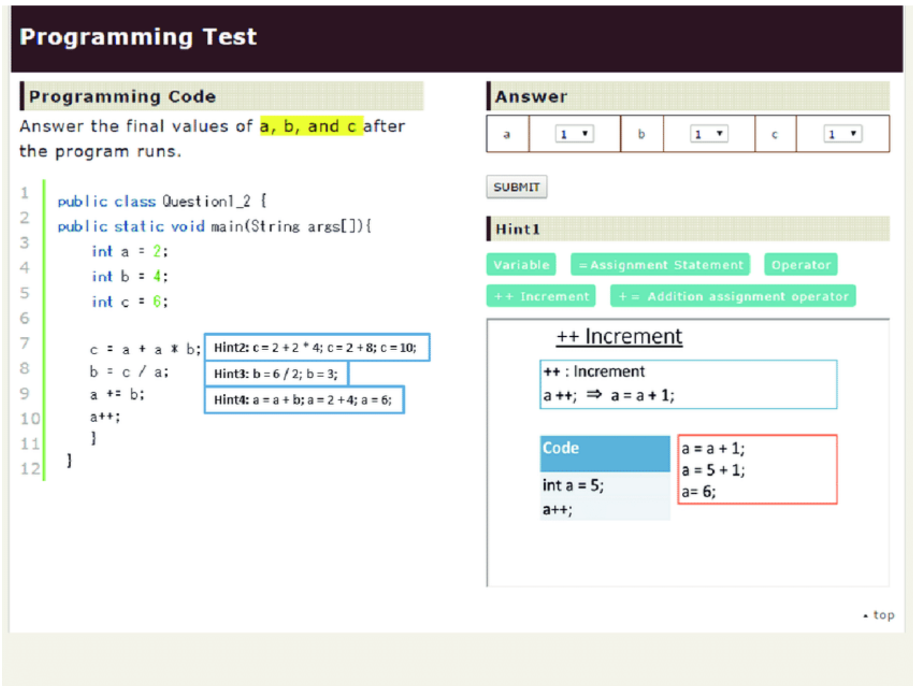

Fig. 2. Dynamic assessment system for programming [15]

Effectively, a dynamic assessment system for programming learning (see Fig. 2) can be applied as a guide for learners based on contextual information [15] and could assist optimize the learning environment allowing students to iterate and improve their work rapidly [16]. Moreover, students can acquire knowledge and proficient problem-solving skills by interacting in a programming learning environment. Therefore, assimilated learning in programming must be based on several critical criteria, such as motivation and ability [17]. In programming, modelling is a generalized explanatory system that uses computer science concepts to organise key characteristics of an object or event of interest. In addition, modelling is a significant factor in helping learners gain a sound comprehension of programming. As such, more efforts must be put in place to help improve the approach to learning programming by using a collaborative process using activities and learning modules [18]. To date, several researchers have proposed some programming learning guidelines based on learners' data and patterns to predict the future direction of programming education, which must also take into consideration their current level of understanding [19].

Table 3
The twelve papers appropriate with the Learner's Model in Programming

No	Authors	Year	Scopus	Web of Science	ERIC	Approach for Modelling
1	Kim *et al*. [12]	2022	√			Scaffolds during block-based programming
2	Wu and Chen[13]	2022		√		Computational thinking (CT) and programming languages
3	Woo and Kim[14]	2022		√		Influential characteristics for learning model
4	Krishnan *et al*., [15]	2022		√		Learning Analytical students' using digital footprints
5	Hsueh *et al*., [16]	2022		√		Learners' behavioural engagement
6	Lyu [17]	2022			√	Extraction module and scoring module
7	Li *et al*., [18]	2022			√	Potential characteristics of learners' emotional states
8	Soclo *et al*., [19]	2022			√	A bilingual Electronic Dictionary (ED)
9	Zuvanov *et al*., [20]	2021	√			Learning from learners' feedback
10	Kim and Mun [12]	2021	√			A self-diagnostic on their learning progress
11	Shobana and Kumar [21]	2021		√		The reliability and validity of the assessment component
12	Liu *et al*., [22]	2021		√		CAD geometric modelling

Table 3 shows the lists of 12 papers published from 2021-2022 that are relevant to the modelling of the learner in programming education. The papers are being indexed in Scopus, Web of Science, and ERIC databases.

3.2 Programming Learner's Experience

The learner's experience plays a crucial role in enhancing the learning process inside the classroom. Hence, it is important to supervise the learning phase meticulously to guarantee that the degree of learning aligns with the learning process. However, the educator must apply and implemented more deliberations in facilitating the learning process of programming to maximise the effectiveness of the acquired knowledge [23].

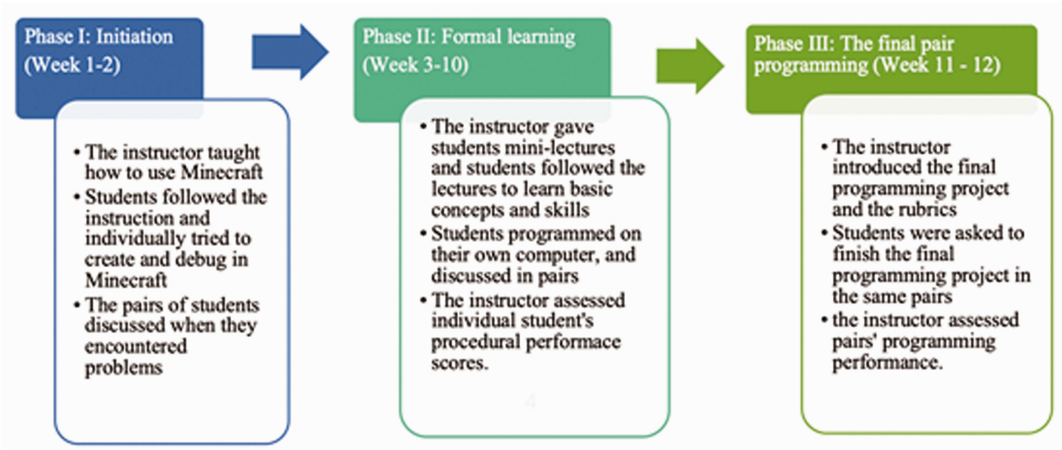

Fig. 3. The students' pair programming learning process[24]

In addition, lecturers can implement the student-paired programming learning process, as depicted in the Fig. 3, to assist students learn programming more efficaciously. Also, an online flip classroom with teacher-led adaptive learning can be employed to improve the quality of online courses [24]. Such an online classroom may embrace community as well as knowledge, learner, and assessment centres. Furthermore, the authors propose a remote learning mode that is guided by relevant socio-cognitive principles, which may include micro lectures, class interactions, after-class exercises, and programming training [22]. Admittedly, the conventional learning approaches commonly used instructors have little impacts on students' motivation and interest in learning programming [25]. In addition, the learning approaches has been a major problem among instructors in teaching big data analytics courses among undergraduates majoring in IT and computer science. In order to encourage students to study programming with greater rigor and enthusiasm, effective learning modules must be created and implemented [26]. Furthermore, the usage of scaffolds can definitely assist struggling students in learning programming with more confidence, as has been seen in various studies that suggest a variety of emotional and psychological aspects have a significant impact on student learning [27-29].

Table 4 shows the lists of 12 papers from 2021-2022 that relate to modelling the learner's experience in education. The papers were extracted from Scopus, Web of Science, and ERIC databases.

Table 4
The twelve papers appropriate with the Programming learner's experience

	Authors	Year	Scopus	Web of Science	ERIC	Approach for Leaner's Experience
1	Devi *et al.*, [30]	2021	√			Using significant automata theory concepts
2	Lim and Moon [26]	2021		√		Using a big data analytics course
3	Kim *et al.*, [31]	2022		√		Debug block-based code with and without scaffolding.
4	Wang [32]	2022	√			Build an accurate and complete learner model
5	Yang and Tan [33]	2022	√			Using a deep learning in the recommendation systems
6	Zhang *et al.*, [34]	2022	√			A personalized learning recommendation method based on learners' trust
7	Kang and Kang [35]	2022	√			Aptitude for reflecting students' learning state
8	Ninrutsirikun *et al.*, [36]	2021	√			To probe the moderation effect of the different learning preferences
9	Liang and Zhang [37]	2022		√		Using a discrete dynamic model big data game
10	Feng [38]	2021		√		Using an evaluation and decision-making
11	Zheng and Dai [39]	2022			√	Explores the application of artificial intelligence technology
12	Wang *et al.*, [40]	2022			√	Using an information extraction and information utilization

4. Discussion

Of late, computer science (CS) teachers and instructors have faced several challenges in assisting students in strengthening their understanding of programming. Additionally, existing assessment methods could be more effective in assessing students' programming skills and knowledge, thus entailing a review of issues surrounding the instructions of programming courses. Against this backdrop, the authors reviewed the current literature to identify several socio-cognitive factors that can help develop a learner model for learning programming. Specifically, the systematic review process consisting of three phases, namely identification, screening, and eligibility, was conducted on three main online databases: Scopus, Web of Science, and Eric. Initially, 401 relevant papers were identified and retrieved from these databases.

Further screening assisted and reduced the number of relevant papers to 50. Finally, 24 articles were retained based on specific selection criteria. As revealed in previous studies, some demographic and socio-cognitive factors significantly impact student programming learning.

This systematic review revealed several factors that require to be carefully considered for the design and development of a learner model for learning programming for students—for instance, using Scaffolds during block-based programming. For example, it is essential to ensure that the teaching of the subject matter must focus on developing students' computational thinking (CT) skills. Such a skill is vital as it can assist students in understanding and using the appropriate ways of discerning programming problems logically and developing proper solutions. From the behavioural perspective, emphasis on motivation should be factored into the design of such a learner model to ensure students can be fixated on exploring programming concepts and principles, with which they can help develop a strong sense of interest in pursuing programming courses. Nonetheless, with strong motivation, students will be willing to invest more time and effort in learning programming courses, leading to a greater understanding of the subject matter.

Unquestionably, learning programming is complex and challenging for many students. As such, cognitive scaffoldings should be made easily accessible to students to help guide them to learn the proper techniques and strategies for solving programming problems. Besides, the instructors must inform the students and reflect on their understanding of the subject matter during and after the learning process. Indeed, this entails an effective means, such as a self-diagnostic assessment system, of providing prompt feedback on problem-solving tasks. To be informed of their learning progress, students can be mentally prepared to take a proper course of action to remedy any mistakes, misconceptions, and misunderstandings they might have.

5. Conclusion

From the learning perspective, the development of the learner model should also focus on deep and personalized learning, given the diversity of student populations in terms of demographics, such as gender, age, ethnicity, and social and economic status (SES). Ideally, learning can be tailored to meet students' specific learning needs and requirements, entailing personalized learning. Moreover, to strengthen students' programming skills, deep understanding should be encouraged by using appropriate techniques that utilize innovative learning objects and environments, enabling them to explore and experiment with programming precepts in greater depth.

Overall, this study has aided the authors in identifying contributing factors critical for adequate instruction in teaching and learning programming courses. The authors' findings of the systematic literature review helped synthesize the learner model's essential elements that must be carefully considered and used. Arguably, the adaptation of such a new learner model could assist and guide instructors to teach programming more efficaciously by concentrating on several students' socio-cognitive backgrounds, which collectively have a significant impact on student learning of programming courses or subjects at the primary, secondary, and tertiary levels of education, expressly in the Malaysian educational context.

Acknowledgement
This research was funded by a grant from Ministry of Higher Education of Malaysia (FRGS Grant Reference Code = FRGS/1/2022/ICT01/UPSI/02/1, Project Code = 2022-0093-109-02). The authors would also gratitude University Pendidikan Sultan Idris (UPSI) for funding this study.

References
[1] Wang, Yanqing, Hang Li, Yuqiang Feng, Yu Jiang, and Ying Liu. "Assessment of programming language learning based on peer code review model: Implementation and experience report." *Computers & Education* 59, no. 2 (2012): 412-422. https://doi.org/10.1016/j.compedu.2012.01.007
[2] Tang, Xiaodan, Yue Yin, Qiao Lin, Roxana Hadad, and Xiaoming Zhai. "Assessing computational thinking: A systematic review of empirical studies." *Computers & Education* 148 (2020): 103798. https://doi.org/10.1016/j.compedu.2019.103798
[3] Qian, Yizhou, and James Lehman. "Students' misconceptions and other difficulties in introductory programming: A literature review." *ACM Transactions on Computing Education (TOCE)*18, no. 1 (2017): 1-24. https://doi.org/10.1145/3077618
[4] Troussas, Christos, Akrivi Krouska, and Cleo Sgouropoulou. "A novel teaching strategy through adaptive learning activities for computer programming." *IEEE Transactions on Education* 64, no. 2 (2020): 103-109. https://doi.org/10.1109/TE.2020.3012744
[5] Ullah, Zahid, Adidah Lajis, Mona Jamjoom, Abdulrahman Altalhi, Abdullah Al-Ghamdi, and Farrukh Saleem. "The effect of automatic assessment on novice programming: Strengths and limitations of existing systems." *Computer Applications in Engineering Education* 26, no. 6 (2018): 2328-2341. https://doi.org/10.1002/cae.21974
[6] Haspel, Richard L., Asma M. Ali, Grace C. Huang, Matt H. Smith, James B. Atkinson, Devon S. Chabot-Richards, Robin M. Elliott et al. "Teaching genomic pathology: translating team-based learning to a virtual environment using computer-based simulation." *Archives of pathology & laboratory medicine* 143, no. 4 (2019): 513-517. https://doi.org/10.5858/arpa.2018-0153-OA
[7] Alammary, Ali. "Blended learning models for introductory programming courses: A systematic review." *PloS one* 14, no. 9 (2019): e0221765. https://doi.org/10.1371/journal.pone.0221765
[8] Buyrukoglu, Selim, Firat Batmaz, and Russell Lock. "Improving marking efficiency for longer programming solutions based on a semi-automated assessment approach." *Computer Applications in Engineering Education* 27, no. 3 (2019): 733-743. https://doi.org/10.1002/cae.22094
[9] Tong, Ke, Yuan Ni Chan, Xiaoqin Cheng, Bobby Cheon, Michelle Ellefson, Restria Fauziana, Shengchuang Feng et al. "Study protocol: How does cognitive flexibility relate to other executive functions and learning in healthy young adults?." *PloS one* 18, no. 7 (2023): e0286208. https://doi.org/10.1371/journal.pone.0286208
[10] Bergmark, Ulrika. "Teachers' professional learning when building a research-based education: context-specific, collaborative and teacher-driven professional development." *Professional Development in Education* 49, no. 2 (2023): 210-224. https://doi.org/10.1080/19415257.2020.1827011
[11] McDonald, Jason K. "The everydayness of instructional design and the pursuit of quality in online courses." *Online Learning*(2023). https://doi.org/10.24059/olj.v27i2.3470
[12] Kim, Semin, and Hyung-Jin Mun. "Design and development of a self-diagnostic mobile application for learning progress in non-face-to-face practice learning." *Applied Sciences* 11, no. 22 (2021): 10816. https://doi.org/10.3390/app112210816
[13] Wu, Ting-Ting, and Jian-Ming Chen. "Combining Webduino programming with situated learning to promote computational thinking, motivation, and satisfaction among high school students." *Journal of Educational Computing Research* 60, no. 3 (2022): 631-660. https://doi.org/10.1177/07356331211039961
[14] Woo, HoSung, and Ja-Mee Kim. "Impacts of Learning Orientation on the Modeling of Programming Using Feature Selection and XGBOOST: A Gender-Focused Analysis." *Applied Sciences* 12, no. 10 (2022): 4922. https://doi.org/10.3390/app12104922

[15] Krishnan, Reshmy, Sarachandran Nair, Baby Sam Saamuel, Sheeba Justin, Celestine Iwendi, Cresantus Biamba, and Ebuka Ibeke. "Smart analysis of learners performance using learning analytics for improving academic progression: a case study model." *Sustainability* 14, no. 6 (2022): 3378. https://doi.org/10.3390/su14063378

[16] Hsueh, Nien-Lin, Bilegjargal Daramsenge, and Lien-Chi Lai. "Exploring the Influence of Students' Modes of Behavioral Engagement in an Online Programming Course Using the Partial Least Squares Structural Equation Modeling Approach." *Journal of Information Technology Education: Research* 21 (2022): 403-423. https://doi.org/10.28945/5010

[17] Lyu, Jianlan. "Writing assistant scoring system for English second language learners based on machine learning." *Journal of Intelligent Systems* 31, no. 1 (2022): 271-288. https://doi.org/10.1515/jisys-2022-0009

[18] Li, Mingyong, Mingyuan Ge, Honggang Zhao, and Ziye An. "Modeling and analysis of learners' emotions and behaviors based on online forum texts." *Computational intelligence and neuroscience* 2022 (2022). https://doi.org/10.1155/2022/9696422

[19] Soclo, Melvin Inertia, Olufemi Deborah Ninan, and Karen Cowan Olufokunbi. "Mobile game-based learning system for a local language." *International Journal of Mobile Learning and Organisation* 16, no. 1 (2022): 52-73. https://doi.org/10.1504/IJMLO.2022.119954

[20] Zuvanov, Luíza, Ana Letycia Basso Garcia, Fernando Henrique Correr, Rodolfo Bizarria Jr, Ailton Pereira da Costa Filho, Alisson Hayasi Da Costa, Andréa T. Thomaz et al. "The experience of teaching introductory programming skills to bioscientists in Brazil." *PLoS computational biology* 17, no. 11 (2021): e1009534. https://doi.org/10.1371/journal.pcbi.1009534

[21] Shobana, B. T., and G. A. Kumar. "I-Quiz: An Intelligent Assessment Tool for Non-Verbal Behaviour Detection." *Computer Systems Science & Engineering* 40, no. 3 (2022). https://doi.org/10.32604/csse.2022.019523

[22] Liu, Hao, Shuhao Xu, and Shenglan Liu. "An online course mode based on microlecture videos: Using CAD geometric modeling course as an example." *Computer Applications in Engineering Education* 29, no. 5 (2021): 1300-1311. https://doi.org/10.1002/cae.22386

[23] Garcia, Manuel B. "Facilitating Group Learning Using an Apprenticeship Model: Which Master is More Effective in Programming Instruction?." *Journal of Educational Computing Research* (2023): 07356331231170382. https://doi.org/10.1177/07356331231170382

[24] Sun, Dan, Fan Ouyang, Yan Li, and Hongyu Chen. "Three contrasting pairs' collaborative programming processes in China's secondary education." *Journal of Educational Computing Research* 59, no. 4 (2021): 740-762. https://doi.org/10.1177/0735633120973430

[25] Wenge, Ma. "Artificial Intelligence-Based Real-Time Communication and Ai-Multimedia Services in Higher Education." *Journal of Multiple-Valued Logic & Soft Computing*36 (2021).

[26] Permana, Asep Yudi, Riskha Mardiana, Nitih Indra Komala Dewi, Rd Vena Ventiany Sumanta, Fanny Muthia Ezzaty, and Puspita Alfira Nareswari. "Evaluation of Classroom Performance in The Post-Covid-19 New Normal Era at The Building Program Vocational High School." *Journal of Southwest Jiaotong University* 57, no. 2 (2022). https://doi.org/10.35741/issn.0258-2724.57.2.11

[27] Chen, Liyan, Xiaoxia Yang, Beizhan Wang, Yang Shu, and Hao He. "Research on augmented reality system for childhood education reading." In *2018 12th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, pp. 236-239. IEEE, 2018. https://doi.org/10.1109/ICASID.2018.8693131

[28] Lu, Yang, Savvas Papagiannidis, and Eleftherios Alamanos. "Exploring the emotional antecedents and outcomes of technology acceptance." *Computers in Human Behavior* 90 (2019): 153-169. https://doi.org/10.1016/j.chb.2018.08.056

[29] Roslan, Nur Widad, Normaliza Abd Rahim, Nur Maisarah Roslan, and Siti Nur Aliaa Roslan. "Students' presupposition towards incooperating AI (Artifical Intelligence) technology in virtual and face-to-face classes." *International Journal of Advanced Research in Future Ready Learning and Education*27, no. 1 (2022): 16-19.

[30] Devi, B. Padmini, S. K. Aruna, and K. Sindhanaiselvan. "Performance analysis of deterministic finite automata and Turing machine using JFLAP tool." *Journal of Circuits, Systems and Computers* 30, no. 06 (2021): 2150105. https://doi.org/10.1142/S021812662150105X

[31] Kim, ChanMin, Lucas Vasconcelos, Brian R. Belland, Duygu Umutlu, and Cory Gleasman. "Debugging behaviors of early childhood teacher candidates with or without scaffolding." *International Journal of Educational Technology in Higher Education* 19, no. 1 (2022): 1-26. https://doi.org/10.1186/s41239-022-00319-9

[32] Wang, Yamin. "Research on online learner modeling and course recommendation based on emotional factors." *Scientific Programming* 2022 (2022). https://doi.org/10.1155/2022/5164186

[33] Yang, Xia, and Leping Tan. "The construction of accurate recommendation model of learning resources of knowledge graph under deep learning." *Scientific Programming* 2022 (2022). https://doi.org/10.1155/2022/1010122
[34] Zhang, Xuefeng, Mengfan Li, Dewen Seng, Xiuli Chen, and Xiyuan Chen. "A novel precise personalized learning recommendation model regularized with trust and influence." *Scientific Programming* 2022 (2022): 1-15. https://doi.org/10.1155/2022/8479423
[35] Kang, Bochun, and Sicheng Kang. "Construction of Chinese language teaching system model based on deep learning under the background of artificial intelligence." *Scientific Programming* 2022 (2022). https://doi.org/10.1155/2022/3960023
[36] Ninrutsirikun, Unhawa, Debajyoti Pal, Chonlameth Arpnikanondt, and Bunthit Watanapa. "Unified Model for Learning Style Recommendation." *Journal of Web Engineering*20, no. 5 (2021): 1487-1526. https://doi.org/10.13052/jwe1540-9589.2058
[37] Liang, Yan, and Wei Zhang. "Adaptive model of discrete real-time linear dynamic logic for the optimization of big data game education mode." *Scientific Programming* 2022 (2022). https://doi.org/10.1155/2022/6185837
[38] Feng, Liu. "Research on higher education evaluation and decision-making based on data mining." *Scientific Programming* 2021 (2021): 1-9. https://doi.org/10.1155/2021/6195067
[39] Zheng, Hanwen, and Dandan Dai. "Construction and optimization of artificial intelligence-assisted interactive college music performance teaching system." *Scientific Programming*2022 (2022). https://doi.org/10.1155/2022/3199860
[40] Wang, Xinzheng, Bing Guo, and Yan Shen. "Predicting the at-risk online students based on the click data distribution characteristics." *Scientific Programming* 2022 (2022). https://doi.org/10.1155/2022/9938260