Education and Information Technologies (2023) 28:11221–11268
https://doi.org/10.1007/s10639-023-11611-0

Learning analytics in programming courses: Review and implications

Uzma Omer[1] · Rabia Tehseen[2] · Muhammad Shoaib Farooq[3] · Adnan Abid[3,4]

Received: 24 August 2022 / Accepted: 19 January 2023 / Published online: 16 February 2023

Abstract
Learning analytics (LA) is a significant field of study to examine and identify difficulties the novice programmers face while learning how to program. Despite producing notable research by the community in the specified area, rare work is observed to synthesize these research efforts and discover the dimensions that guide the future research of learning analytics in programming courses (LAPC). This work demonstrates review of the learning analytics research for initial level programming courses by exploring different types and sources of data used for LA, and evaluating some pertinent facets of reporting, prediction, intervention, and refinements exhibited in literature. Based on the reviewed aspects, a taxonomy of LAPC research has been proposed along with the associated benefits. The results reveal that most of the learning analytics studies in programming courses used assessment data, which is generated from conventional assessment processes. However, the analysis based on more granular level data covering the cognitive dimensions and concept specific facets could improve accuracies and reveal the precise aspects of learning. In addition, the coding analysis parameters can broadly be categorized into code quality and coding process. These categories can further be classified to present twenty-five sub-categories of coding parameters for analyzing the behaviors of novice programmers. Moreover, efforts are required for early identification of effective and ineffective behavioral patterns through performance predictions in order to deliver timely interventions. Lastly, this review emphasizes the integration of related processes to optimize the future research efforts of conducting the learning analytics research for programming courses.

Keywords Programming · Prediction · Intervention · Reporting · Review

✉ Uzma Omer
uzma.omer@ue.edu.pk

Extended author information available on the last page of the article

1 Introduction

Learning analytics is the collection, measurement, analysis, and reporting of data about learners and their context to support the related educational environments (SoLAR, 2012). The continuously producing big data from educational institutes triggers the need of its analysis to extract meaningful information that could be utilized for the betterment of existing environments. LA support the learning and teaching processes by providing empirical evidence on the effectiveness of pedagogical enhancements. It can be applied to extract different forms of information from the learner's data enabling to, for example, identify at-risk students (Veerasamy et al., 2022), predict academic performances (Khan et al., 2021), and determine the usefulness of tools or techniques (Zheng et al., 2022). Such information can be useful for teachers and students in many ways. The teachers, for example, can plan the interventions while the students can get the personalized feedback (Grawemeyer et al., 2022) which could help them in identifying the areas to be focused for making improvements in the specific aspects of a course. Although LA is a discipline that relates to all levels of education, yet it has gained immense popularity in higher education (Viberg et al., 2018). Several educational institutes have already opted to adopt various LA techniques for knowledge discoveries as well as for guiding their decision-making policies (Avella et al., 2016). The community at the same time is producing promising research resulting in offering a wide range of LA processes and approaches (Mangaroska & Giannakos, 2018) to educational institutes, which could help in achieving their goals at institution level as well as at individual course level.

The initial level programming courses serve as pre-requisites to learn various higher-level courses in computer science and associated disciplines; however, the students of these courses generally face difficulties while learning how to program (Watson & Li, 2014). The learning difficulties tend to raise if students are not provided timely and appropriate interventions at the beginning of learning to program. Numerous efforts are made by researchers to analyze the learner's data of programming courses in order to find the hidden facets that could be utilized for assisting instructors to design optimized interventions for novice programmers helping them overcome their learning difficulties (Ihantola et al., 2015). These efforts are made to address a diverse range of objectives and to reveal different types of information such as identification of appropriate programming language (Wainer & Xavier, 2018), and the effective and ineffective behaviors of novice programmers (Pereira et al., 2021). Moreover, a number of studies predicted learners' performances by evaluating different parameters and applying various machine-learning techniques (Jamjoom et al., 2021; Omer et al., 2020). Furthermore, the LA studies have shown useful approaches of reporting the findings (Hellings & Haelermans, 2020; Khan et al., 2021), deriving interventions (Azcona et al., 2018; Dorodchi et al., 2017), and presenting enhancements in LA research (Carter et al., 2017; Su et al., 2015). Whilst there are specific objectives associated with particular LA studies, the basic purpose is to identify the aspects that could enhance the teaching and learning processes by evaluating different forms of data and applying various analysis techniques.

Although a number of studies carried out LA for programming courses yet rare or no work is performed to comprehensively synthesize these research efforts and examine the aspects of LA specifically for programming courses to guide and identify the future research dimensions of LAPC. This work presents review of the state-of-the-art LAPC in the light of LA five step model (Pardo, 2014). The specified LA model is chosen to formulate the investigating areas of this study as in our opinion; the particular model exhibits LA through more precise aspects as compared to other LA models or frameworks, which appear to be more anecdotal as present LA through relatively broader facets (Kaliisa et al., 2022; Khalil et al., 2022).

The LA five steps model describes the following steps: 1) *Capture*, which relates to capturing the data as well as the techniques of collecting the data; 2) *Report*, which is based on the approaches that are used to report data to the stakeholders; 3) *Predict*, which focuses on predictions of students' performance; 4) *Act,* which presents interventions to assist learners, and 5) *Refine*, which demonstrates the improvements in LA research. Figure 1 presents an illustration of LA five steps. The LA process starts with the capturing of data based on some pre-defined objectives. It is an essential stage, which an LA study is required to be carried out as a fundamental process step of analysis. The analysis is performed to discover knowledge related to prediction, intervention, refinement, and for reporting the analyzed aspects. It could be based on the defined statistical (España-Boquera et al., 2017; Funabiki et al., 2016), data mining and machine learning techniques (Khan et al., 2021; Pereira et al., 2021) posing evaluation of some proposed approaches (Carter et al., 2017; Omer et al., 2020).

In addition to reviewing the LAPC literature in the light of LA five steps model, this work presents a taxonomy of LA posing various dimensions of each step for programming courses. Moreover, a classification of coding analysis parameters, to evaluate the novice programmer's behaviors, has also been shown in this paper. Based on the meta-analysis of the investigating aspects and findings, some pertinent gaps and concerns in LAPC research are highlighted as well as some opportunities for future research are indicated. This study will be helpful for instructors, practitioners and researchers enabling them to channelize and optimize their future efforts in their respective domains.

Fig. 1 An illustration of LA five-step model

This paper is further organized into the following sections. Section 2 demonstrates the related work by posing the comparison of already conducted research in the field and highlighting the need of conducting this research. Section 3 discusses the methodology for carrying out this study by demonstrating the main steps involved in the process of systematically reviewing the relevant studies. Section 4 presents the results and findings by illustrating the responses to the research questions obtained from synthesizing the shortlisted studies. Section 5 exhibits the discussion and analysis by presenting the principal findings and the implications, which elaborate the concerns and opportunities identified on reviewing the LAPC literature. Section 7 lists the limitations of this work. Lastly, the study has been concluded in Section 8 by highlighting the main findings and outcomes of this work.

2 Related work

Educational Data Mining (EDM) is a discipline that is closely related to LA as it also aims to examine educational data; however, the EDM tends to focus more on technology-assisted analysis whereas LA includes more diverse analytical methods and approaches (Chatti et al., 2012; Hashima et al., 2018). In addition, the LA is more focused on human interpretation of analysis results and techniques as well as the visualization of analyzed and processed data while EDM typically relates to the techniques and methods of analyzing the data exhibiting specific aspects of analysis (Baker & Inventado, 2014).

A number of studies presented the LA review on the basis of the aspects that holistically relate to various domains of courses as well as different levels of education (Mangaroska & Giannakos, 2018; Romero & Ventura, 2020). However, the nature of various domain of studies as well as courses are different, which drives the need to examine the LA studies for specific courses. The literature exhibits rare work to review LA research specifically for programming courses. One of these works examined LA and EDM for programming (Ihantola et al., 2015). The main focus of this work was to evaluate the types of programming data used for analysis and the techniques of collecting and sharing that data. It reviewed students, programming and environmental aspects on learning. This study covered some facets of prediction, data capturing; however, it lacked a comprehensive discussion on others elements covered in this article like combinations of data used, types of predictors, reporting, interventions, and refinements.

Another study presented the future directions of LA in terms of the data that could be generated by integrated development environments (IDEs) (Hundhausen et al., 2017). In this work, the authors proposed taxonomy of programming behaviors that mainly presented parameters classified into the categories of compilation, execution, debugging, and editing. These dimensions relate to the programming behaviors, which emerge during the process of coding while our review additionally highlights some other pertinent dimensions of coding behaviors that relate to the quality of code as well. The analysis of programming behaviors through the parameters depicting the quality of code opens the arena of evaluating some significant programming skills such as optimization and problem solving, which need to be instilled in novice programmers.

Considering the specific nature of programming courses that largely depends upon the use of tools and performing programming activities, the synthesis of the LA research for programming courses could potentially reveal the precise aspects of LA for programming courses enabling to optimize the future dimensions in the specified field. The related work mainly focused on examining the programming data while this study additionally provided review of other types and sources of data like surveys, past academic records, and observations. Moreover, various aspects of reporting, prediction, intervention, and refinements have also been evaluated, which are classified to present the taxonomy of LAPC along with the associated benefits. Furthermore, the identification of various coding analysis parameters lead to pose twenty-five sub-categories of these parameters to examine the behaviors of novice programmers. Finally, this work highlights some major concerns emerged as outcome of the review and some significant areas as opportunities to further explore the field of LAPC. To the best of our knowledge, this is the first effort, which reviewed the LAPC research in view of the LA steps presented in literature and provided the future research dimensions to optimize and guide the research efforts for LAPC.

3 Methodology

This study has been conducted by following the systematic literature review (SLR) guidelines presented in literature (Kitchenham & Charters, 2007) and the techniques observed in an existing SLR (Omer et al., 2021). Major phases of this SLR and the respective activities performed in each phase are demonstrated below:

- **Planning for the Study:** Conception through preliminary analysis, developing the research questions, establishing search strategy, devising inclusion criteria and exclusion criteria for shortlisting the papers, and defining the quality criteria.
- **Searching and shortlisting the papers:** Identifying the papers from databases, and applying inclusion and exclusion criteria for shortlisting.
- **Data extraction and classification:** Extracting the data according to the research questions, **c**lassification, and synthesis of selected studies.
- **Analyzing and Reporting:** Reporting results, findings, and analysis.

3.1 Planning for the study

This phase is initiated through a primary search and analysis of the studies from databases to build the basic know-how of the specified area of study. After this, the research questions are framed according to the motivation of carrying out this work and the associated objectives. Then, a search strategy is devised by identifying the target keywords and the target databases to search the relevant articles. This was followed by establishing the inclusion/exclusion criteria for shortlisting the papers.

3.1.1 Conception through preliminary analysis

This step is performed to build the essential understanding of the field (Omer et al., 2021) by preliminary search and analysis of the relevant research articles. It resulted in acquisition of 25 articles, which were identified through manual searching of relevant articles. This stage helped in establishing the basis to streamline the further steps for conducting this study.

3.1.2 Research questions

The research questions have been formulated to investigate the facets related to different steps of LA model (Pardo, 2014). Table 1 presents the research questions along with the associated research objectives and the respective LA steps.

3.1.3 Search strategy

The searching of the relevant research articles is performed by considering the search strategy followed in a recent study (Omer et al., 2021). An initial searching of papers has been carried out to build the basic understanding of the significant facets of LAPC, which resulted in the acquisition of 22 papers from different sources. It was done through manual searching of articles by using combinations of different keywords presented in Fig. 2. These keywords are categorized as primary, secondary, tertiary, and additional.

It was observed that the LAPC studies used different titles to refer initial level programming courses such as initial level programming, first year programming, fundamentals of programming, and object oriented programming. Therefore, we selected “programming” as a primary keyword to extract programming focused analysis as any specific name for initial level programming courses in the search string could enable the threat of losing studies, which used other titles to refer the same programming courses. In addition, the first and second programming courses in computing disciplines are also named as Computer Science 1 (CS1) and Computer Science 2 (CS2), respectively, which were also used as the primary keywords to extract the LA research exhibiting the analysis of initial level programming courses. The “learning analytics” and “educational data mining” were selected from secondary set of keywords to target the studies focused on learning analytics and those relate to educational data mining. The educational data mining is used as a keyword as the specific domain poses common elements with the field of learning analytics (Qazdar et al., 2019). The “student”, “course”, “learner”, and “novice programmer” were used as tertiary keywords while “perform”, “code”, and “exam” were considered as additional keywords to develop the final search string. The use of Boolean operators is emphasized to relate the different keywords in the search string (Farooq et al., 2021). We have applied the Boolean operators AND, and OR to link the chosen keywords resulted in the formulation of the search string for searching the articles from the digital repositories based on following relation between the keywords. The keywords and their abbreviations that were finalized to build the search string are shown as Listing 1.

Table 1 Research questions with respect to research objectives

LA step	Research question (RQ)	Research objective (RO)
Capture	RQ1: What types and sources of data are used for LAPC?	RO1: To identify the types and sources of data used for LAPC
Report	RQ2: How different aspects of reporting are addressed in LAPC research?	RO2: To examine how reporting of various aspects of learning is posed in programming courses
Predict	RQ3: How learner's performance predictions are performed for programming courses?	RO3: To investigate various facets of performance predictions of novice programmers
Act	RQ4: What forms of interventions are presented to assist novice programmers?	RO4: To explore different forms of interventions reported for programming courses
Refine	RQ5: What refinements are examined in LA research of programming courses?	RO5: To evaluate the enhancements made in LA research of programming courses

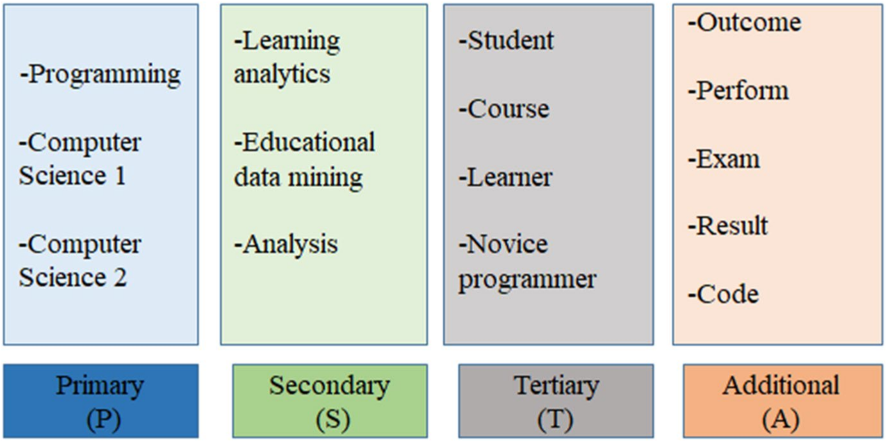

Fig. 2 Keywords for searching the relevant articles

Listing 1
Keywords for search string

Programming (P), Computer Science 1 (CS1), Computer Science 2 (CS2), Learning Analytics (LA),Educational Data Mining (EDM), Student (S), Course (C), Learner (L), Novice Programmer (NP), Perform(PE), Code (CO), Exam (E)

Following equation is formulated to devise the generic search string for identifying the relevant literature from digital repositories:

$$R = \forall \, [(P \vee CS1 \vee CS2) \wedge (LA \vee EDM) \wedge (S \vee C \vee L \vee NP \vee PE \vee CO \vee E)]$$

In the above equation:

R *Results obtain from the search string*

$\forall$ *for all.*

$\vee$ *OR operator.*

$\wedge$ *AND operator.*

The generic search string to find the articles of LAPC from the databases is shown as Listing 2.

Listing 2
Search string

("programming" OR “CS1” OR “CS2”) AND (" learning analytics" OR “ educational data mining”) AND("student" OR "learner" OR "course" OR “novice programmer” OR "perform" OR "code" OR “exam”)

The preliminary search and scrutiny of relevant studies also helped to identify the target databases for searching the related articles. Following databases were identified to find the related literature of LAPC: IEEE Xplore, ACM Digital Library, SAGE, SpringerLink, Wiley, Taylor & Francis, Scopus, ScienceDirect, and ERIC. Table 2 enlists the search strings applied on different databases to find the relevant articles alongwith the filters applied in each database. Since the main objective of this work is to examine the state-of-the-art LAPC, the relevant studies produced between the years 2014 to 2022 are considered for conducting this review. Therefore, the year filtration for the specified range of years has been applied on the digital repositories.

3.1.4 Inclusion and exclusion criteria

The extracted records from databases are shortlisted on the basis of inclusion and exclusion criteria. Since LA focuses on analysis of educational data and related learning environments, the inclusion/exclusion criteria are devised to shortlist the studies, which depicted the analysis of data of programming courses or performed evaluation of the presented solutions. Table 3 enlists the inclusion/exclusion criteria established to include or exclude the studies from the articles identified from different databases.

3.1.5 Quality assessment criteria

The quality assessment of the studies is a significant step to of conducting the SLR. This step is performed to assess the appropriateness of study design and the ability of the research study to present details about the specified investigating aspects in the given domain. A number of SLRs carried out this step for evaluating the strength of the evidence provided by review (Farooq et al., 2022; Omer et al., 2021).

The scoring criteria labeled as (a), (b), (c), and (d) were devised to assess the quality of the shortlisted studies. These criteria have been established by considering the one developed for scoring the studies in a previous work (Ouhbi et al., 2015) (a) Solution poses methodical potential, (b) Outcome of the study is clearly reflected through the conclusions, (c) Methodology is demonstrated clearly (d) Main focus covers the investigating area of this research. The selected studies for the specified criteria are rated on the basis of following scores: 1 for fulfilling the specific criterion, 0 for not fulfilling and, 0.5 for partially fulling the respective criterion. Following is the scoring classification of the studies: high, if the study obtains a cumulative score of greater than or equal to 3; average, if the study acquires cumulative score that falls between 2 and less than 3; and low in case the study obtains a total score of less than 2.

3.2 Searching andshortlisting the papers

The primary searching of related papers from databases resulted in emergence of a huge number of records, which were required to be further scrutinized for shortlisting. Figure 3 presents the stage-wise shortlisting of studies and the number of papers included and excluded at each stage. Firstly, the papers were shortlisted

Table 2 Search strings with respect to databases

Database	Search query	Filter applied
ACM DL	[[All: "programming"] OR [All: "CS1"] OR [All: "CS2"]] AND [[All: " learning analytics"] OR [All: " educational data mining"]] AND [[All: "student"] OR [All: "learner"] OR [All: "course"] OR [All: "novice programmer"] OR [All: "perform"] OR [All: "code"] OR [All: "exam"]]	Proceedings and journals 2014–2022
ERIC	("programming" OR “CS1” OR “CS2”) AND (" learning analytics" OR “ educational data mining”) AND ("student" OR "learner" OR "course" OR “novice programmer” OR "perform" OR "code" OR “exam”)	Peer reviewed articles 2014–2022
IEEE Xplore	(programming OR CS1 OR CS2) AND (learning analytics OR educational data mining) AND (student OR learner OR course OR novice programmer OR perform OR code OR exam	Conferences, journals, and early access articles 2014–2022
SpringerLink	’("programming" OR “CS1” OR “CS2”) AND (" learning analytics" OR “ educational data mining”) AND ("student" OR "learner" OR "course" OR “novice programmer” OR “perform” OR “code” OR “exam”)’	Conference papers and articles 2014–2022
SAGE	for[[All "programming"] OR [All “CS1″] OR [All”CS2]] AND [[All " learning analytics"] OR [All " educational data mining"]] AND [[All "student"] OR [All "learner"] OR [All "course"] OR [All "novice programmer"] OR [All "perform"] OR [All "code"] OR [All "exam"]]	Research articles 2014–2022
Taylor & Francis	[[All: "programming"] OR [All: "CS1"] OR [All: "CS2"]] AND [[All: " learning analytics"] OR [All: " educational data mining"]] AND [[All: "student"] OR [All: "learner"] OR [All: "course"] OR [All: "novice programmer"] OR [All: "perform"] OR [All: "code"] OR [All: "exam"]]	Articles 2014–2022
Wiley	"("programming" OR “CS1” OR “CS2”) AND (" learning analytics" OR “ educational data mining”) AND ("student" OR "learner" OR "course" OR “novice programmer” OR "perform" OR "code" OR “exam”)"	Journals 2014–2022
Scopus	"programming" OR “CS1” OR “CS2” AND " learning analytics" OR “ educational data mining” AND "student" OR "learner" OR "course" OR “novice programmer” OR "perform" OR "code" OR “exam”	Journals 2014–2022
ScienceDirect	("programming" OR “CS1” OR “CS2”) AND (" learning analytics" OR “ educational data mining”) AND ("student" OR "learner" OR "course" OR “novice programmer” OR "perform" OR "code" OR “exam”)	Journal articles 2014–2022

Table 3 Inclusion and exclusion criteria

Inclusion criteria (IC)	Exclusion criteria (EC)
IC1: The investigating areas of this work are clearly discussed in the study	EC1: The paper is published before 2014
IC2: The research is empirically validated	EC2: The paper is duplicated
IC3: The study exhibits the analysis of programming courses' data related to higher education	EC3: The study shows non-validated study such as opinion paper, and reviews; however, those are included in discussion
IC4: The paper is published in peer-review journals or presented in conferences	EC4: The study presents LA research focusing on pre-higher education level programming courses
	EC5: The paper is not in English

by applying the search string and year-wise filtration from the databases, which resulted in obtaining 3,568 records. After this, the papers indicating a clear depiction of learning analytics study were shortlisted by reading the titles, which left 1,107 studies to be moved to the next stage of shortlisting. These studies were then shortlisted by reading the abstracts. This step resulted in inclusion of 240 articles, which were further examined for relevancy by reading full text. The shortlisting process from databases resulted in the acquisition of 61 papers. This was followed by a

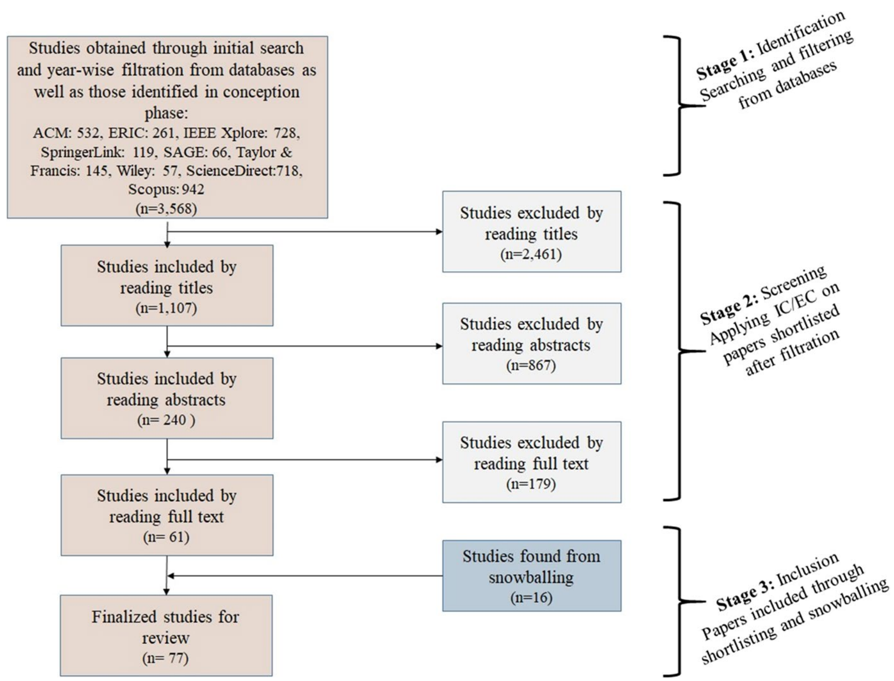

Fig. 3 Stage-wise shortlisting of studies

snowballing process for searching the studies in order to mitigate the risk of losing those articles that are relevant to the underlying domain but could have left from the prior searching and shortlisting processes.

3.2.1 Searching through snowballing

In addition to searching the relevant studies on different research portals by using the devised search string, another search cycle has been carried out through backward snowballing (Wohlin, 2014). The searching of relevant records through this technique was performed by scrutinizing the reference lists of the shortlisted studies obtained through previous searches. It was performed to find studies that may have overlooked from the previous searching and shortlisting process. The shortlisting through snowballing resulted in inclusion of 16 more articles in the pool of papers selected to conduct this review. After searching the papers from snowballing technique 77 papers were finalized to review the investigating aspects of the subject area. The shortlisting process was carried out by the authors. The results of the shortlisting process were reviewed by two independent reviewers. In order to find the interrater reliability, we calculated the Kappa coefficient, the value of which turned out to be 0.92 indicating a strong agreement.

3.3 Data extraction and classification

The bibliometric facts of the selected articles are acquired directly from the publication sources. The data extraction as well as the classification is performed on the basis of the investigating areas demonstrated through RQs.

RQ1 focuses on the types of data used for LA in programming courses; therefore, the types and the combination of those types of data as well as the details about the sources of data are extracted from the selected studies for the respective RQ. The RQ2 is about reporting aspects, so the data related to different facets of reporting such as timing, presentation, audience, content, and channel were extracted. RQ3 examines the prediction related aspects, for which the data from prediction focused studies were retrieved related to the predictors, target variables, data sets used and the related aspects of data sets such as size, number of institutes, instances, and terms were obtained. The target variable refers to the dependent variable on which the impact of prediction is examined. The instances of the course refers to the number of courses while the terms depict the semesters to collect the data. The data pertaining to the technique used for prediction and evaluation of prediction accuracies was also acquired from prediction focused LAPC research. RQ4 investigates the interventions related facets; therefore, the data reflecting different approaches of interventions were acquired. RQ5 evaluates the refinements reported in the different steps of LA for which the particular refinements and the respective stages demonstrating those refinements were examined. Furthermore, any pertinent aspects identified that pose some significant dimensions of LAPC research such as the techniques applied for analysis and the evaluating aspect of the programming courses were also extracted from the selected studies.

4 Results and findings

This section presents the results and findings obtained on synthesizing the shortlisted articles. The details of classification and quality scoring of the shortlisted paper are shown in Table 10 (Appendix).

4.1 Quality assessment

The quality appraisal of selected studies depicted that most of these studies scored above average where 4 being the highest score as per our scoring criteria. The differences in scores were mainly due to the quality criteria 'a' and 'b' while most of the studies addressed the criterion 'c'. However, the criteria 'd' is partially addressed in majority of the selected studies. Figure 4 depicts the analysis of selected studies on the basis of quality scoring classification.

4.2 Publication years and channels

The synthesis of selected studies for publication years and channels has been performed to identify the trend of LAPC research over the years and the major publication sources involved. It has been revealed that a major portion of LAPC research has been conducted in the years 2017, 2018, and 2015, which make about 23% (17), 19% (15), and 16% (11) of these studies, respectively. The year 2014 produced about 12% (10) while each of the years 2020 and 2016 added around 8% (6) articles in the set of selected studies. About 7% (5) of the shortlisted articles are produced in the year 2019, and 4% (3) in 2021 all of which are journal articles and only 5% (4) of the selected studies have been produced in the year 2022. The final search cycle for the data collection was carried out in June 2022.

Figure 5 shows the number of shortlisted articles published through different channels in various years. The shortlisted papers comprised of about 39% (30) journal articles and 61% (47) conference papers. Most of the research published in

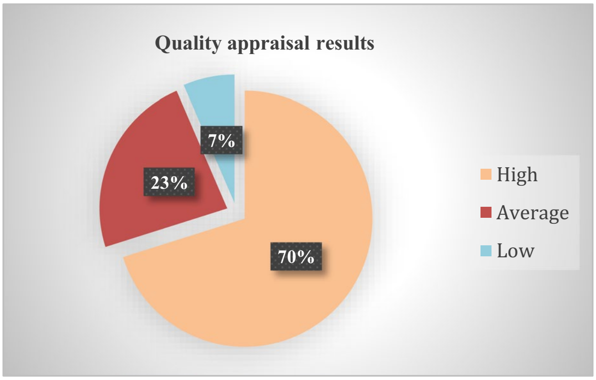

Fig. 4 Distribution of selected studies according to quality scoring classification

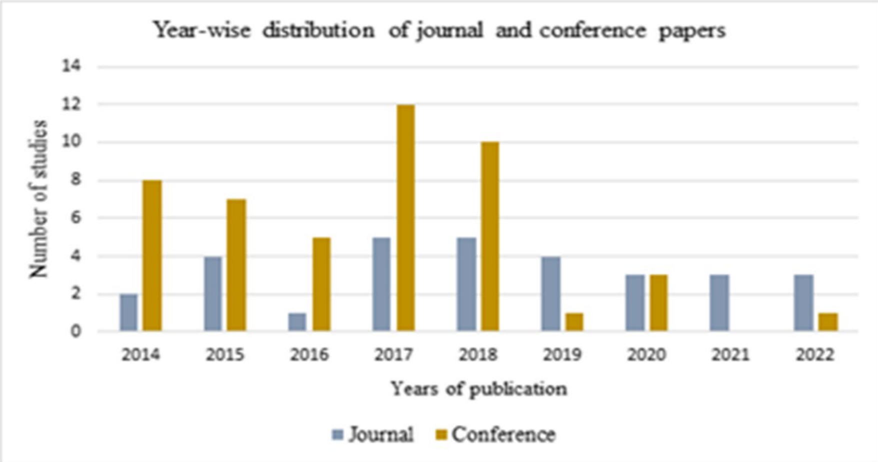

Fig. 5 Distribution of publication years and channels

journals are of Q1 rank, which makes about 77% (23) of journal articles. One of the journal papers is of Q2 rank while about 11% (7) of these articles do not show any specific ranking. A majority of conference papers are presented at core B ranked conferences. This makes a share of 53% (25) of these papers. About 23% (11) of the conference studies are of Core A* and Core A while just one paper was presented in the conference ranked as Core C. The selected studies include about 26% (12) of the conference papers having no specific ranking. Major share of journal articles is of ACM Transactions on Computing Education while IEEE Frontiers in Education Conference has the major share of conference papers. These two sources added about 33% (10) and 43% (20) of the journal and conference papers, respectively. Other major sources that contributed in the pool of shortlisted research include ACM Special Interest Group on Computer Science Education Technical Symposium, ACM International Conference on International Computing Education Research, IEEE Transactions on Education, IEEE International Conference on Computer Software & Applications, and ACM/IEEE International Conference on Software Engineering.

4.3 RQ1: What types and sources of data are used for LAPC?

The selected studies against RQ1 are examined for the types and sources of data that have been used for LAPC.

4.3.1 Types and sources of data

Major types of data used in selected studies are assessment, behavioral, and inquiry responses. As several LAPC studies used more than one type of data, the number of studies in the count for the types and sources of data overlap. The assessment data is used in about 62% (48) of the shortlisted articles while the behavioral data is analyzed in about 53% (41), and the inquiry responses are examined in about 33%(25) of these studies. Other than these types of data, the background and observational data have also been used for LAPC where the background data is analyzed in about 9% (7) of the selected studies while only two studies used the observational data.

Table 4 enlists the parameters acquired from different types of data and the respective acquisition sources. The data is acquired through different sources most of which are based on the activities related to the conduct of course. The course activities are those, which relate to conducting the course and generated by the instructors whereas the learning activities are based on individuals' pace of learning and their learning styles identified through the activities like practicing different exercises and tool interactions. These activities include those that are necessary for conducting the courses. The assessment data is captured through assessment processes posing data of performances in exams, assignments, lab work, and quizzes. Inquiries are generally made through surveys to peruse the experiences or perceptions of the stake holders on certain aspects of the course. The observational data are also gathered while conducting various course activities. This type of data is based on the stakeholders' judgments and observations. Background data is acquired through past academic records as well as through surveys. This type of data poses past academic records and the students' characteristics like gender and age. The behavioral data is obtained through learning or assessment activities. The behavioral data poses parameters of tool interactions, which are obtained through the use of different tools used for conducting courses. In addition, the behavioral data also exhibits the learning or programming behaviors emerge during learning and course conduct activities. The event-based data could be gathered from tools interactions potentially appropriate for behavioral analysis. However, to examine the parameters like accomplishments, and satisfaction, the other types of data such as assessments, surveys, background could be useful for LAPC.

Determining the appropriate type of data and the parameters is one of the main factors to be considered before performing LA to examine the specific aspects of learning. The LAPC research depicted that the analysis of the cognitive learning provides precise findings (Omer et al., 2020). In this context, the Blooms' taxonomy has been used to examine and analyze learning at various levels of cognition through assessment data (Dorodchi et al., 2017; Gomes & Correia, 2018); however, the parameters for cognitive learning analytics using behavioral data is required to be identified. The studies used assessment data are based on the evaluation of graders, which could be inconsistent as reflected in research (Albluwi, 2018). Relying on such type of data for LA could give inappropriate results and misleading findings. Hence, the measures to evaluate the data authenticity and integrity needs to be taken to ensure the reliability on the outcomes of analysis.

The assessment and behavioral data have been acquired through the support as well as programming tools. The use of tool is obligatory in programming course for practicing and lab work. In addition, the support tools are also useful for assisting instructors and students during the teaching and learning processes, which include assessments, feedback deliverance, peer support etc. Some of the examples of such tools used to generate the specific data for performing LA in programming courses

Table 4 Types of data, sources, and aspects acquired

Types of data	Sources	Aspects acquired	References Example
Assessment	Course activities	Performances in exams, assignments, lab work, and quizzes	Omer et al., 2020; Gomes et al., 2016; Allinjawi et al., 2014; Effenberger et al., 2020
Inquiry	Surveys	Experiences and perceptions	Hilton & Rague, 2015; Santana et al., 2018; Janke et al., 2015; Yeomans et al., 2019
Observational	Course activities and learning activities	Judgments and observations	Rosiene & Rosiene, 2015; Wood & Keen, 2015
Background	Students' records and surveys	Past academic performances and individual's information	Hellings & Haelermans, 2020; Azcona et al., 2018; Watson et al., 2014; Ninrutsirikun et al., 2020
Behavioral	Tools, learning activities, and course activities	Social interactions, tool interactions behaviors, programming behaviors, and learning styles	Echeverría et al., 2017; Fu et al., 2017; España-Boquera et al., 2017; Lagus et al., 2018

include TAsystem, to establish social learning activities and analyzing comments (Echeverría et al., 2017). Another tool, Quizit is used to produce MCQs and serve as a social learning platform (Chung & Hsiao, 2020). The CPPCheck and Clang are used for static analysis of codes to generate reports on students' submissions (Delev & Gjorgjevikj, 2017). The studies also used author-defined tools such as WEbTA, which is designed to examine the students' programs to deliver automated and dynamic feedback (Ureel & Wallace, 2015, 2019). Similarly, the use of a tool named WPGA is examined for assistance in grading and delivering feedback based on the data of paper-based assessments (Hsiao et al., 2017). Moreover, LAPLE is used to provide feedback on programming exercises and recommend personalized learning materials (Fu et al., 2017). Most of the author defined tools are used to generate data related to the responses on the feedback deliverance and auto-grading which posed the major focuses of LAPC researched based on learners' data of system interactions (Ala-Mutka, 2005).

The tool interactions determine the learners' behavior as well as their learning styles. This could be useful in determining the engagements and participation levels of novice programmers. Such data can also be analyzed to identify different behavioral patterns, which can support the instructor to devise appropriate assistance to the students by determining the effective and ineffective behaviors. An increasing trend of using games in computer science education (Ahmad et al., 2020) specifically in programming courses has been observed (Rojas-López et al., 2019). Generally, the surveys are conducted and evaluated to examine the experiences to learn programming using games; however, no specific data type is reported in the selected LAPC studies through the learners' interactions with gaming interfaces. Apart from the identified types of data from LAPC literature, another potential type that can be used for LA is the physiological data (Hundhausen et al., 2017), which can be collected through gamification fostering the behavioral analysis through the parameters like mood, and stuck in the flow.

4.4 RQ2: How different aspects of reporting are addressed in LAPC research?

The shortlisted articles for this RQ are examined to investigate the content, channel, presentation, timing, and the audience for reporting, which are identified as significant dimensions of reporting of learners' performances (Hundhausen et al., 2017).

4.4.1 Channel

The dashboard is identified as one of the useful channels to provide insight into different aspects of students' performance (Fu et al., 2017; Seanosky et al., 2017; Hellings & Haelermans, 2020). The guidelines to students have been provided through learning management system (LMS) by analyzing the critique report on students' programs submitted and tested on IDE (Ureel & Wallace, 2019). This work indicates towards integration of tools which could serve for identifying and reporting the related aspects of students learning. In addition, the reporting has also been made through emails where the audiences were given choice to opt-in or out from receiving the emails (Azcona et al., 2018). Since, the use of tool is an obligatory aspect of conducting programming courses, integrating the tools would support the analysis significantly resulting in providing better assistance to the stakeholders. It can also help in providing the personalized learning materials to students on their respective accounts by auto-analyzing the students' programs, and identifying the most suitable set of parameters for performance predictions.

4.4.2 Content

The performance parameters are presented through precise comments on the intermediate code (Ureel & Wallace, 2019), feedback on the proficiency of particular concepts through the frequency of applying a concept while solving programming problems (Seanosky et al., 2017), and predictive performance in comparison to the peers (Hellings & Haelermans, 2020). Moreover, the performance predictions were transmitted by examining academic backgrounds and behavioral logs (Azcona et al., 2018). Furthermore, the difficult areas course is reported for instructors; e.g., the troublesome concepts which are examined by perusing the concepts presented in the most visited pages of the system. The personalized learning materials were referred according to the types of errors students encountered (Fu et al., 2017) and the performances in the quizzes (Lin et al., 2018).

The contents of reporting depict the course aspects that are applicable to all students such as the common weak areas of course that can be determined, for example, by perusing the frequently occurring errors while applying different programming constructs. Moreover, the contents could present the individual's performances and learning gaps by evaluating the individual's activities. Furthermore, the reporting can be either comparative, which is presented by posing the comparison of a learner's performance with peers or it can be made without having information about the peers' learning states. The former way of reporting could inculcate a sense of competition with the peers, which could motivate the learners to work more and perform better. However, the comparative reporting could leave positive or negative impact on learner's motivation and behavior, which needs to be monitored by the instructors in order to ensure the positive impact of reporting on students' performances.

4.4.3 Presentation

The visualization of analysis through graphs has been reported in literature where the students are presented two graphs; one graph highlighted per student and other showed cumulative competency (Seanosky et al., 2017). Another study presented performance prediction of individual student in comparison to the average prediction of peers using two gages representing each of the specified aspect (Hellings & Haelermans, 2020). Furthermore, video feedback is provided to students on their submitted programs (Hilton & Rague, 2015). The self-regularization habits were observed on acquiring the feedback through the specified technique.

In addition to observing the typical approaches of presentation such as charts and graphs, use of sophisticated techniques of visualization for reporting the learners' performances have been followed such as stop light metaphor (Khan et al., 2021; Ureel & Wallace, 2015) as indicator of different levels of performance and the performance meter for demonstrating the predictions. Furthermore, the classification tree as presentation technique to pose different performing classes is also used for visualization of reporting aspects (Veerasamy et al., 2022). More sophisticated visualization techniques could make the presented information simplified to comprehend. It would emphasize intrinsic motivation for audience and help them perform their respective roles effectively.

4.4.4 Timing

Mainly two different timings of reporting are observed: regular and random. The regular reporting takes place after specific time intervals (Azcona et al., 2018) while the random reporting provides random alerts to audience based on identifying certain states of learning (Ureel & Wallace, 2015). The latter could be used as messages to aware the audience with specific levels of learning. The reporting could be based on varied time periods with the intervals unknown to the students. It seems that different timings of reporting could leave different psychological effect on the learning and performance. This would be a call for investigating the psychological effect on students' performance of different timings of report. Furthermore, the timings of reporting could be linked with different sections of the course indicating the state of learning in specified stages of the course. Immediate reporting would be useful to provide support for students as they do practice of programming problems. A programming assistance would be helpful in providing timely help to the novice programmers. The regular reporting would be useful on different stages of the course, which could be based on conducting assessments after specific time intervals or reflect the programming behaviors exhibiting programming practices as per some specified programming standards.

4.4.5 Audience

The LAPC research reported the findings for students (Ureel & Wallace, 2019) as well as for instructors (Hellings & Haelermans, 2020); however, the reporting could be required for the course evaluators who are responsible for assuring and enhancing the quality of the courses (Omer et al., 2020). They may seek diverse range of information such as quality of conducting the course, comparative analysis of performance in the successive terms taught by different instructors, cognitive competencies acquired by learners, and the extent of achieving the learning and programing objectives. It has been observed that the LAPC research is generally confined to course level where the audiences are instructors and students but LA has a huge potential to support educational institutes by guiding their decision-making polices (Avella et al., 2016). It is significant to conduct LA for academic institutes to sustain the programing course in computer science and associated disciplines and to ensure quality of these courses based on the predefined quality parameters and objectives.

4.5 RQ3: How learners' performance predictions are performed for programming courses?

The performance predictions are examined on the basis of the predictors used, the target variables, the approaches of predictions as well as the aspects related to the data sets used for predictions. Table 5 illustrates how the aforementioned facets related to predictions are addressed in LAPC research.

4.5.1 Predictors

The predictors are the parameters that can be used to predict learners' performance. Three out of the five of the identified types of data, which include assessment, behavioral and background, were used as predictors. The predictors based on the behavioral data in most of the cases relate to programming behaviors. Background data is typically used to predict the programming potential of students to get an insight into students' capabilities before indulging into the programming. The observational data and inquiry responses are not identified as data types for predictions. This is perhaps due to the nature of these types of data which could be more vulnerable towards inaccuracies due to individual's perspectives and judgments.

The predictors are either acquired directly from the sources or processed after acquisition. The directly acquired predictors include the parameters like number of posts (Carter et al., 2017) and number of questions asked (Ashenafi et al., 2015). The processed predictors can be further classified into two sub-categories. One of these categories includes those in which simple mathematical calculations is performed; e.g., the severity of programming errors through the frequency of occurrence and time to fix errors where the latter two were obtained directly from the sources (McCall & Kölling, 2019). The other category of predictors poses more sophisticated transformational techniques before analyzing the specific aspects, e.g., a study designed metrics of aggregate cognitive score and aggregate level-wise cognitive score to calculate the cumulative and individual level cognitive performance on sets of concepts (Omer et al., 2020). The metrics were based on a transformational set up of the concepts and assessment data acquired using the technique of concept mapping in augmentation with Bloom's taxonomy.

4.5.2 Target variables

The studies generally predicted the final exams and the course grades (Al-Rifaie et al., 2017; Lagus et al., 2018). Some studies carried out performance predictions for weekly assignments or lab work (Azcona et al., 2018; Carter et al., 2017). Other studies predicted the behavioral pattern (Pereira et al., 2021), performances in mid exams (Carter et al., 2017), and knowledge and skills (Ninrutsirikun et al., 2020). A shortfall in case of predicting the performances in exams is identified when measures taken to address the authenticity of rubrics and grading to evaluate the quality of data are generally not discussed. Only two studies used the approach of Cronbach's alpha test to examine the reliability of grading (Ninrutsirikun et al., 2020;

Table 5 Prediction details in related studies

Ref. no	Target variables	Data sets			Predictors	Prediction approach	
		Sizes	Sources: instances/institutes/terms	Types	Parameters	Technique	Evaluation
Al-Rifaie et al., 2017	Final grade	993	single /-/-	Behavioral	Clicking events	SVM	-
Estey et al., 2017	Mid and Final exam score	514	four/single/four	Behavioral	Compilation attempts and hints used while practicing	-	-
Carter et al., 2017	Assignment performance, Final grade	95 + 108 + 110	three/single/multiple	Behavioral	Number of posts, transition between states of program	Linear regression	Variance
Ashenafi et al., 2015	Final exam score	206	two/single/-	Behavioral	Questions asked, peer comments or answer, peer rates the comments	Multiple linear regression	RMSE
Liao et al., 2019	Final exam score	334 + 747 + 345	/three/two/multiple	Behavioral	Clicker's responses on instructor's questions	SVM	Sensitivity, specificity, AUC
Omer et al., 2020	Quiz performance	40 + 37	two/two/single	Assessment	Cognitive performance on particular concepts	Linear regression	MAE
Azcona et al., 2018	Programming lab exam performance	134 + 140	two/single/multiple	Background, Behavioral	Previous skills, students' characteristics, Programming submissions	KNN, Decision tree	F-measure

Table 5 (continued)

Ref. no	Target variables	Data sets			Predictors	Prediction approach	
		Sizes	Sources: instances/ institutes/terms	Types	Parameters	Technique	Evaluation
Khan et al., 2021	Final grade	50	single/single/two	Assessment, Background, Behavioral	Test marks, learning behaviors (e.g., attendance), course attributes (e.g., class duration), background (e.g., gender), assessments (e.g., CGPA)	Decision tree	Sensitivity, f-measure
Pereira et al., 2021	Behavioral patterns	2056	multiple/single/ multiple	Behavioral	Programming behavioral parameters (e.g., amount of change, ide use, copy paste, Watwin score, error quotient etc.)	Random forest, deep learning	F-measure
Veerasamy et al., 2022	Final exam score	391	single/single/four	Assessment	Formative assessment performance	Classification tree	Sensitivity, specificity
Koong et al., 2018	Final grades	34	single/single/single	Assessment, Behavioral	Mid exams results, number of answers and modification time	Random forest, SVM, Neural network, Naive Bayes	-

Table 5 (continued)

Ref. no	Target variables	Data sets			Predictors	Prediction approach	
		Sizes	Sources: instances/institutes/terms	Types	Parameters	Technique	Evaluation
Lagus et al., 2018	Final exam score	247 + 101	two/single/-	Background, Behavioral	Background skills, students' characteristic, compilation states, events, completion	SVM, random Forest	F-measure
Castro-Wunsch et al., 2017	Final grade	897 + 519	/two/single/multiple	Behavioral	Number of steps to solve problem, correctness	Neural network, Naive Bayes, decision tree, random forest	-
Ahadi et al., 2015	Final grade, exams questions scores	86 + 210	/wo/single/two	Behavioral	Watwin score, Error quotient	Naive Bayes, Decision tree	F-measure, ROC
Carter et al., 2015	Assignment performance, Final grade	95	single/single/single	Behavioral	Compilation and execution states transition	Linear regression	Variance
Jamjoom et al., 2021	Pass/fail classification	244	single/single/four	Assessment, Background	Cumulative marks of past academic record, scores in mid, quizzes, and practical	Decision tree, Naive Bayes, KNN, SVM	F- measure
Ninrutsirikun et al., 2020	Knowledge and skills	115	single/single/single	Assessment, Background	Test marks, and academic background	Multiple linear regression	MSE

Data sets sizes refer to the sizes of data used for LA, institutes means number of educational institutes involved for collecting the data, terms refer to the number of academic terms such as semesters or academic years, and the instances mean number of course instances involved for collecting data.

Omer et al., 2020). Furthermore, some studies presented at-risk students as the target group for prediction; however, it seems that the other group could implicitly be identified while investigating a particular group. Moreover, the techniques of data cleansing and their impact on the prediction results are rarely observed.

4.5.3 Data sets

The data sets used for predictions of learners' performance relate to the students who are studying the programming courses in different institutes where research studies used the data sets of students of single or multiple institutes, terms, and instances. The sizes of data observed in the selected studies range from 50 to 2056, which poses the number of students involved in prediction-based studies. However, depending upon the parameters used for predictions, the actual size of the data can be calculated through the number of parameters generated against each students. For example, the involvement of 77 students in the data set where each student's data of 192 assessment item's responses is evaluated, makes the data size of 14,784 responses (Omer et al., 2020).

4.5.4 Prediction approach

Various techniques are observed to predict learners' performances. The support vector machine (SVM) is the most commonly used classifier to predict performances in programming courses. Other techniques for prediction include random forest, K-nearest neighbor (KNN), Naive Bayes, decision tree, random forest, neural network and deep learning.

The prediction results are evaluated through number of ways such as sensitivity, specificity, F- measure, variance, root mean square error (RMSE), mean absolute error (MAE), receiver operating characteristic curve (ROC), and area under the curve (AUC). The results indicates that the F-measure is the most commonly used technique to evaluate the prediction accuracy in LAPC studies.

4.6 RQ4: What forms of interventions are presented to assist novice programmers?

The interventions present actions performed on identifying the particular states of learning. An intervention can be described as an event in which a combination of information, guidance and feedback is provided (Hundhausen et al., 2017). The forms of interventions depend on modes of education and hence could be auto or manual.

The manual interventions are designed for traditional learning environments. Online or blended learning modes mostly observe auto-intervention techniques involving various tools. The types of interventions include the feedback based on which different forms of feedback are delivered (Azcona et al., 2018). Other interventions include the material recommendation identified by evaluating the performance gaps and other forms such as devising some learning activity planned by identifying the common weak areas of a cohort of students (Koong et al., 2018), planning and executing a specific tasks that to support the learning process (Dorodchi et al., 2017), and developing and presenting the learning objects or tools to assist the specific learning needs of students on particular concept (Lin et al., 2018).

The studies posed different actions as interventions such as providing program improvement suggestions to novice programmers, which are presented to them by examining the prediction results (Azcona et al., 2018). Some of the students corrected the programs by incorporating the given suggestions and presented better outcomes as compared to the students who did not correct the programs as per the given suggestions. This indicates the effectiveness of useful interventions to help students overcome their learning difficulties. Another study presented the performance prediction to the instructors through LA dashboard (Hellings & Haelermans, 2020). A research is performed to present the personalized learning material to students (Koong et al., 2018). Moreover, a work suggested the adaption of multimedia tool to mitigate the gap of secondary digital on performance (Hsu & Mimura, 2017). The determination of actions could be incorporated in the support tools by integrating the prediction features. Smart recommender system would thus help to optimize learning processes and can also reduce the teachers planning time by assisting them in deciding and delivering the appropriate interventions. Other studies presented interventions through reviewing the student's developed concept maps (Turner et al., 2018), incorporating the use of new tools (Edwards et al., 2014a; Santana et al., 2018) that support learning in programming courses through detailed feedback (Edwards et al., 2014b), by devising class activities (Dorodchi et al., 2017), and by providing customized learning material according to the specific needs of the novice programmers (Lin et al., 2018; Malliarakis et al., 2016). Applying the learning theories such as cognitive load theory or social cognitive theory can help to design optimized interventions. The factors highlighted by the learning theories need to be considered to design appropriate interventions for novice programmers.

4.7 RQ5: What refinements are examined in LA research of programming courses?

The refinements in LAPC research are examined on the basis of improvements shown in the existing work with respect to different LA steps. Need of refinements is emphasized as replication or re-analysis of the same problem with different settings could give inconsistent findings (Ihantola et al., 2015).The studies posing methodical techniques pose potential for replication and re-analysis. Table 6 enlists the refinements in LAPC research and the associated steps of LA model.

Most of the refinements are related to predictions, which include those in which prediction accuracies were improved by including or excluding the predictors e.g., the social learning behavioral data when augmented with the programming data resulted in improved prediction accuracies (Carter et al., 2017). Moreover, the improvements are examined through the comparisons of potential predictors to find the most optimized set of predictors; e.g., a study examined different types of tests and exams questions to find the appropriate questions that correlates with the final

Table 6 Refinements in LAPC research

Study	Refinements	Related stage
Baker & Inventado, 2014	Improved reporting for instructors through auto identification of parameters and visualization of students' weekly prediction results	Report
Carter et al., 2017	Adding social learning behavior data to programming behaviors for predictions	Predict
Azcona et al., 2018	Auto-sending continuous and adaptive feedback based on assessment results	Act
Lagus et al., 2018	Precisely transferring the heterogeneous instances	Predict
Castro-Wunsch et al., 2017	Reproduced previous prediction model and employing on different context	Predict
Gomes et al., 2016	Using additional set of data to predict performances	Predict
Su et al., 2015	Detailed feedback by adding the parameters that tells about program structure in addition to running the tests	Report
Hijon-Neira et al., 2014	Enhanced existing features by integrating a module that guides and motivates through different messages	Report

exams (Gomes et al., 2016). Furthermore, improvement in instance transfer method of traditional machine algorithms is observed through transferring the data set according to their specified values in the system thus addressing the concern of heterogeneity of the data (Lagus et al., 2018).

The reporting related improvements are generally confined to the visualization and automation of presenting findings to the stakeholders. The refinements are also examined to improve feedback by making the feedback process more adaptive and frequent, which could be enforced through the automated capturing of data for feedback deliverance (Azcona et al., 2018).

The refinement step emphasizes the iterative nature of LA in which the conducted studies need to be revisited for improvements; however, we observed little effort in this direction, which calls for carrying out the work to attempt improvements by reproducing the existing research (Ihantola et al., 2015).

5 Discussion and analysis

The LAPC research is classified to present the taxonomy depicting an illustration of the precise dimensions related to different LA steps along with the benefits of integrating LA in programming courses, as shown in Fig. 6.

The types and sources of data capturing show five further classifications identified on reviewing the LAPC literature. Similarly, the report step is presented by specifying the content, channel, presentation, timing, and audience of reporting where each of these sub-dimensions show further categories. The step of predict is demonstrated through the predictors, target variables of prediction, techniques used for predictions, and the evaluation approaches for finding prediction accuracies where each of these dimensions exhibit different categories. Similarly, the step of act is presented through the forms of interventions reported, which include auto, and manual while the refine step is demonstrated through the LA steps that posed refinements. These steps include act, report, and predict as per the investigating areas of this work. The LA is performed for knowledge discovery that pave the way of guided decision-making. It could be useful to support learners through optimized interventions by addressing their specific learning gaps. Furthermore, the quality enhancements could also be made by quality evaluation and assurance of learning and teaching approaches.

As described in Section 1, a minimalistic form of LA study relates to one or two steps of LA model. Example of such work is the research that relates to the steps of “capture” and “report” by demonstrating the capturing of learners’ data and reporting to the stakeholders through automated solutions (Hsiao et al., 2017). The data capturing is an integral part of LA process, which has been carried out in all of the selected studies. The prediction studies include 33% (25) of the selected studies while reporting related aspects are presented in 15% (11) of these studies.

The interventions identified by perusing “act” are shown in about 11% (8) of the selected studies and the refinements are exhibited in only 14% (10) of these papers.

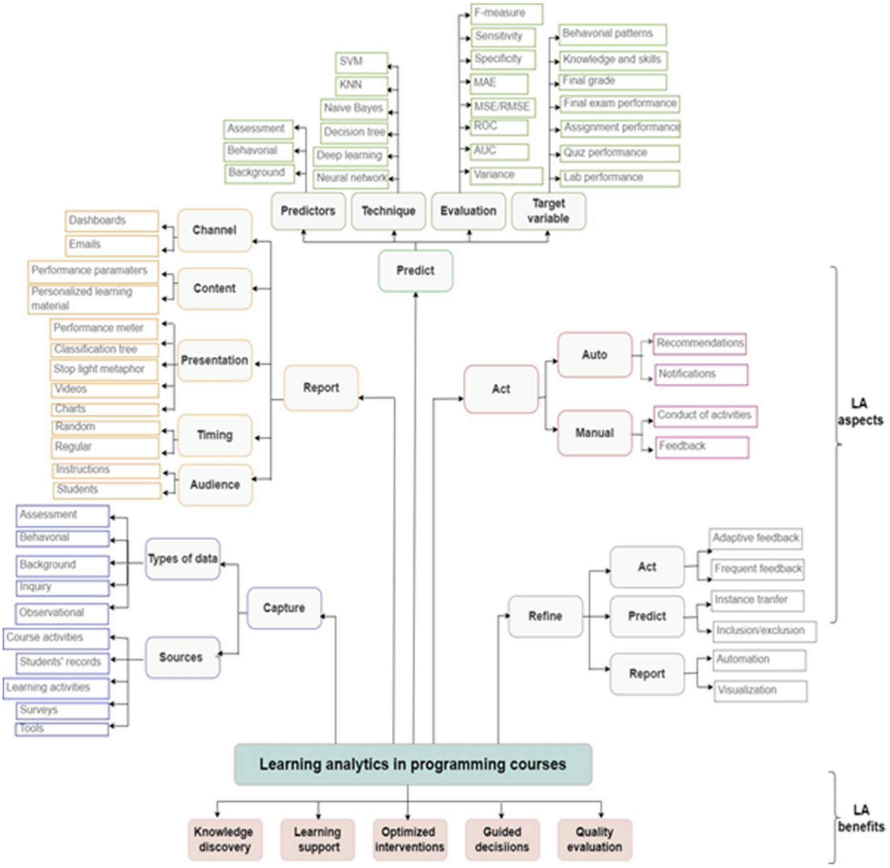

Fig. 6 A *taxonomy of LAPC and associated benefits*

Analyzing the LA steps coverage indicates the "act" and "refine" as neglected areas, which need to be investigated from different dimensions. Moreover, the review of LAPC identified some studies that presented the analysis of aspects in addition to those illustrated as information acquisition and presentation in Fig. 1. These studies examined applicability of different techniques and evaluation of different aspects of learning. In this context, about 22% (16) studies examined the applicability of technique and around 23% (17) of the selected studies investigating some learning aspect. Table 7 enlists the analyzing aspects and analysis approaches of the studies that examined the applicability of some technique and its effect on learning while Table 8 illustrates the analyzing aspects and analysis approach of the studies that evaluated some learning dimension in programming courses.

The applicability of different techniques of teaching, learning and analysis has been examined from various perspectives. The evaluation of learning is performed, e.g., by comparing the evaluating facet of different groups of leaners and by scrutinizing different parameters most of which relate to programming

Table 7 Application of techniques

Study	Technique	Approach for analyzing the effectiveness of technique
Echeverría et al., 2017	Collaborative learning technique	Comparison of performances in exams with and without deploying the specific learning technique
Ullah et al., 2019	Rule-based technique to classify performances on different cognitive levels	Comparison of the classification results obtained by applying the rule-based technique with the results obtained through manually classification
Bhatia et al., 2018	Technique to auto-correct programs	Percentage of appropriate corrections made in programs
Gomes & Correia, 2018	Technique to teach basic programming loop	Comparison of average, standard deviation and median values of the performances of learners on different questions depicting various levels of Bloom's taxonomy
King, 2018	Peer-assessment technique	Comparison of experimental and control groups' performances as well as survey-based analysis on the applicability of respective analyzing aspects
Marcolino et al., 2018	Use of catalogue of gesture in M-learning	Survey-based analysis of learning experiences of learners
Iqbal Malik & Coldwell-Neilson, 2017	Application of teaching and learning approach based on the sequence of approach, development, result and improvement	Comparison of results of programming questions from treatment and control groups
Funabiki et al., 2016	Application of fill-in-the-blanks of Java programs as assessment approach	Correlation of students' demonstrated performance through specified assessment with course grades
Janke et al., 2015	Outside in teaching approach by directly teaching object oriented concepts	Comparison of percentages of correct answers of experimental group with control group
Rosiene & Rosiene, 2015	Flip-classroom approach	Students' learning experiences and instructors' observation on students learning behaviors
Scott et al., 2015	Robot Olympics to exercise programming and enhance practicing	Analysis of survey results related to students' programming behaviors and final coursework performance comparison of experimental and control groups
Wood & Keen, 2015	Technique to bridge the imperative and Object orientation concepts	Comparison of students' performance of experiment and control groups
Doshi et al., 2014	Technique of taking cue-based practical exam	Analysis of the questionnaire inquiring students' understanding
Rubio et al., 2014	Teaching through physical components	Analysis of survey inquiring students learning experiences
Effenberger et al., 2020	Additive factor model for refining knowledge components	Comparison with baseline models

Table 8 Evaluation of learning aspects

Study	Learning aspect	Approach for evaluating the learning aspect
Chung & Hsiao, 2020	Persistency of learning	Applying Poisson mixture model to cluster response-stream data
Delev & Gjorgjevikj, 2017	Programming behaviors	Identifying the most commonly occurring errors through frequency of error occurrence
España-Boquera et al., 2017	Programming environment	Correlation of the analytical data obtained from the programming environment (accesses/compilations/executions) with marks obtained
McCall & Kölling, 2019	Severity of programming errors	Evaluating the number of occurrences of programming errors
Yeomans et al., 2019	Difficulties of concepts	Comparing the learners' responses on questionnaire about difficult to understand programming concepts
Albluwi, 2018	Grading differences	Identification of similarities and differences in grading of same solutions from different graders
Chaweewan et al., 2018	Programming aptitude and skills	Evaluating the amount of time consumption in solving programming problems and submission frequencies of programs
Esteero et al., 2018	Difficulties of concepts	Comparing the performances of groups applying different programming concepts while solving programing problem
Wainer & Xavier, 2018	Appropriateness of programming language	Comparing the performances of learners using different programming languages
Seeling & Eickholt, 2017	Learning behaviors	Comparison of performance of learners in different iterations of course
Berges et al., 2016	Programming competencies	Identification of the number and frequencies of errors' occurrences
Simkins & Decker, 2016	Learning difficulties	Percentages of responses on survey about difficulties of concepts and reasons of facing the difficulties perceived by learners
Xinogalos, 2015	Difficulties in the concepts of Objects and Classes	Percentages of evaluating students' competencies based on detailed understanding of specific concepts
Allinjawi et al., 2014	Concept-effect propagation	Analysis of involvement of different concepts while solving problems through normalized marks and assigning weights to the concepts involved through test item concepts relationships
Heinonen et al., 2014	Differences of errors made by novice programmers	Segregating performances according to the types and number of errors
Zur & Vilner, 2014	Difficulties of concepts	Averaging the performances of students in individual concepts
Pereira et al., 2020	Identifying behavioral classes	Comparing semantic and syntactic differences of computer programs

behaviors. Moreover, the analysis of specific aspect is performed by examining the responses to the survey conducted to investigate the perception or learning experiences of students. The evaluation and application based studies typically used simple statistical technique to examine the analyzing aspect for LA like average, median, standard deviation, significance tests, correlation, frequencies, and percentages. The correlation analysis is mostly performed to evaluate the relationship of specific parameters or metrics with exams' performances.

Use of more than one sources and types of data for LA has been emphasized (Tempelaar, 2021) as can improve precisions and accuracies (Carter et al., 2017). This triggers the need to examine the combinations of data types used for LA in programming courses. Figure 7 depicts the various types of data as well as the different combinations of these types used in the selected studies for LA.

It has been revealed that 39% (18) of studies, which used assessment data, relied only on assessment data and 61% (29) of those studies used assessment data in combination with the other types of data. The behavioral data is another major type of data that has been analyzed in about 53% (41) of the selected studies. Around 37% (15) of these studies relied on the behavioral data as a single source for analysis and about 63% (26) considered analyzing behavioral data along with other types. The third major type of data is based on inquiry responses which is analyzed in about 33%(25) of the selected studies where about 33% (8) of these studies used it as a sole source of data and around 67% (17) in combination with other data types. The background data is analyzed in about 9% (7) of the selected studies while none of these studies used it as a single type of data for

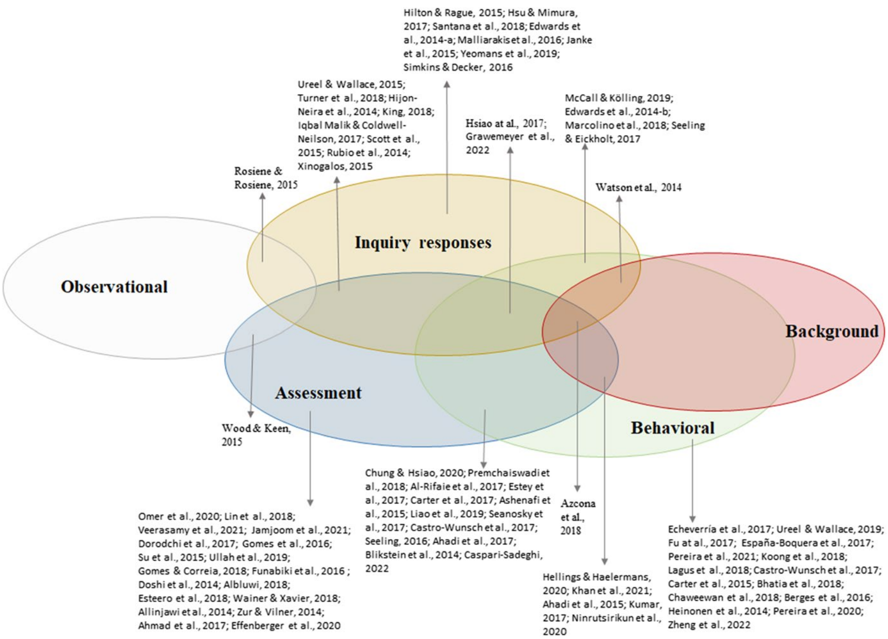

Fig. 7 Types and combinations of types of data

analysis. The least relied type of data is observational, which is used in only two studies and that too with the combination of other data types.

The behavioral data is one of the major types of data used for LA in programming courses, which reflected four sub-categorizes of learners' behaviors described as:

i) Tool interactions behaviors, which present the actions performed while using tools and interacting with the applications. This type of learners' behaviors has been examined through the parameters such as include number of actions performed (Premchaiswadi et al., 2018), and accesses to the developed projects (España-Boquera et al., 2017).
ii) Social learning behaviors, which exhibit the learners' activities while collaborating with each other or reviewing each other's work (Al-Rifaie et al., 2017; Estey et al., 2017; Turner et al., 2018). Online social activities involve the data of participation demonstrating parameters like number of posts (Carter et al., 2017; Echeverría et al., 2017), and rating of peers' responses (Ashenafi et al., 2015).
iii) Learning activities behaviors, which demonstrate the actions, performed during different course related activities (Hsiao et al., 2017).
iv) Programming behaviors, which depict the learners' behavior while solving programming problems. Such behaviors represent parameters like time to fix errors (Liao et al., 2019), and frequencies of errors (Estey et al., 2017).

The analysis of programming behaviors exhibit the aspects reflecting the code quality, and coding process. The code quality can be analyzed by examining the parameters related to optimization, presence of dead code, understandability, and the outcomes of code-testing processes while the coding process is evaluated through the facets that can be sub-categorized as attempts of different activities, code state, coding styles, and error occurrences. Figure 8 poses a classification for coding parameters to analyze the programming behaviors of novice programmers while the Table 9 enlists the research, which used the specified coding parameters for LA.

Learning to program requires building the skills like problem solving and optimization. However, LAPC studies generally lacked in presenting the information about how the analyzed parameters relate to the specific programming skills. For example, certain number of compilation attempts while solving a programming problem represent what skills of programming and how these can be linked with certain levels of performances. Moreover, some of the coding analysis parameters relate to the testing of programs while there are concerns regarding the abilities of novice programmers to verify or validate their programs through formal processes as the concept of testing is not generally included in programming course content (Omer et al., 2020). Consequently, the results of the studies, which used parameters such as test cases passed, correctness or completeness, could have been influenced by this potential shortcoming in the curriculum of programming courses. Such prior deliberations need to be made before considering the types and parameters of data used for LA.

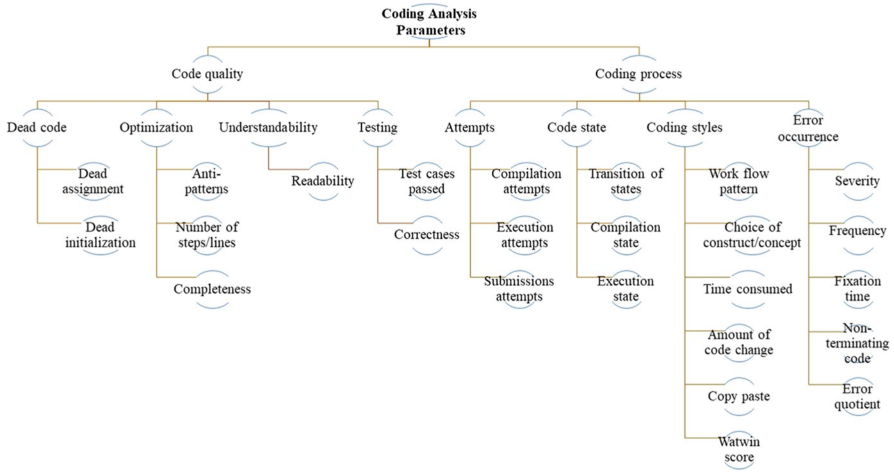

Fig. 8 A classification of coding analysis parameters

6 Implications

The implications are presented by highlighting the principal findings, concerns to present the dimensions that exhibits gaps and issues, and opportunities to demonstrate the areas that could further be explored to examine more diverse facets of LA for programming courses.

6.1 Principal findings

The review of LA in programming courses resulted in identification of four major types of studies that are categorized on the basis of the scope of analysis. Examples of these types include the research, which presented sophisticated processes resulting in discovering new facets for analyzing learning (Carter et al., 2017; Omer et al., 2020) and transformed the conventional scenarios into the automated solutions (Hilton & Rague, 2015; Ureel & Wallace, 2019). Such studies also investigated the potential predictors for devising prediction models for future research (Funabiki et al., 2016; Premchaiswadi et al., 2018; Seanosky et al., 2017), and demonstrated performance predictions by applying the specific prediction techniques (Ninrutsirikun et al., 2020; Veerasamy et al., 2022). The latter two types either performed predictions or supported prediction work by evaluating different parameters that could potentially be used as predictors while the first two types are confined to analysis. The prediction

Table 9 Classification of coding analysis parameters

Main classification	Sub-classification	Parameter	Reference
Code Quality	Dead code	Dead assignment	Delev & Gjorgjevikj, 2017
		Dead initialization	
	Optimization	Anti-patterns	Turner et al., 2018
		Number of steps/lines	Castro-Wunsch et al., 2017; Ahadi et al., 2015; Pereira et al., 2020
		Completeness	Lagus et al., 2018
	Understandability	Readability	Chaweewan et al., 2018
	Testing	Test cases passed	Chaweewan et al., 2018
		Correction	Lagus et al., 2018; Bhatia et al., 2018; Effenberger et al., 2020
Coding Process	Attempts	Compilation attempts	Carter et al., 2015; Lagus et al., 2018; Carter et al., 2017
		Execution attempts	Carter et al., 2015; Carter et al., 2017
		Submission attempts	Chaweewan et al., 2018
	Code state	Transition of states	Carter et al., 2015; Carter et al., 2017
		Compilation states	Carter et al., 2015; Lagus et al., 2018; Carter et al., 2017
		Execution states	Carter et al., 2015; Carter et al., 2017
	Coding style	Workflow patterns	Estey et al., 2017; Heinonen et al., 2014
		Choice of constructs/concepts	Esteero et al., 2018
		Time consumed	Chaweewan et al., 2018
		Amount of code change	Blikstein et al., 2014; Pereira et al., 2020
		Copy paste	Pereira et al., 2020
		Watwin score	Pereira et al., 2020
	Error occurrence	Severity	McCall & Kölling, 2019
		Frequency	McCall & Kölling, 2019; Estey et al., 2017; Watson et al., 2014
		Fixation time	McCall & Kölling, 2019
		Non-terminating code	Edwards et al., 2014b
		Error quotient	Pereira et al., 2020

support studies mostly investigated the relationship between different parameters of the data through correlation and regression analysis.

The behavioral and assessment data are the two major types used for LAPC. The shortlisted studies mostly examined the impact of analysis on final outcomes of the course, which generally depicts the course grades or final exam and less work is observed to evaluate the LA impact on the aspects related to the in-between stages of the terms. Reporting the analysis findings on different stages of course would help learners to timely identify their learning gaps and support instructors to devise appropriate interventions before teaching more complex programming concepts or planning to teach the forthcoming modules of the programming courses. The presentation of reporting is significant to perceive the states of leaning. The need of dashboard to present different aspects of learning has been emphasized in literature to reveal different dimensions of learning and enabling learners and instructors to make guided decisions. Intervention focused studies are very few in number in the pool of shortlisted studies, which depicted a neglected area in the specified field of study. As long as the impact of analysis on strategies used to support learning is not discussed, the analysis would be less significant for the stakeholders. Similarly, few studies examined the refinements and less replication, re-analysis or reproductive studies are observed. Such researches are significant to be conducted in order to evaluate the robustness of proposed solutions. A number of techniques other than LA steps presented in Fig. 1. are found. This indicates the need of exploring a comprehensive model of LA covering different dimensions of LAPC research.

6.2 Concerns and opportunities

The analysis of the results revealed some pertinent concerns of LAPC research. These concerns reflect the gaps and issues that indicate to explore some significant dimensions presented as future research opportunities for researchers. Figure 9 shows the opportunities for future research of LAPC with respect to the identified concerns. The future research opportunities along with the respective concerns as listed as follows.

- **Data granularity:** The granularity of data reveals the level of details it exhibits. Using more granular level data for LA could unveil the detailed facets of learning and improve precisions in results. The assessment data is used in 65% of the selected studies while only 15% of these studies provided information about the specific concepts and the level of cognition of learners the assessment items examined. Considering not all concepts lie on same difficulty level and the cognitive levels of assessment items could also vary, ignoring these detail could affect precision in results. It was discovered that only one study used the cognitive performance of learners on specific concepts to predict learners' performance and reported improvements in accuracies (Omer et al., 2020). In addition to improving the prediction accuracies, the analysis based on more granular data could lead to more customized reporting and optimized interventions.

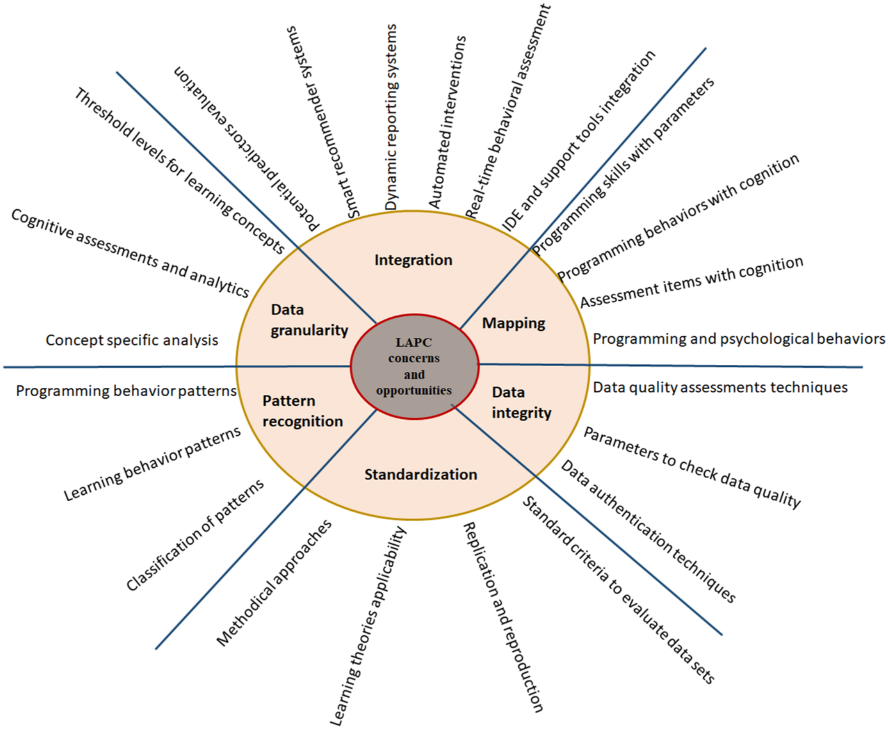

Fig. 9 LAPC concerns and future research opportunities

Analyzing learning by incorporating the cognitive aspects and by considering the concept specific analysis could lead to more precise findings. Since, learning a relatively higher level programming concept depends upon learning one or more related concepts, establishing the relation between different levels of learning among various concepts could help to understand the threshold levels of learning required for related concepts in programming courses.

- **Integration:** A number of studies performed analysis to identify the most appropriate parameters that could be used to examine students' performances as discussed in Sect. 6.1. It was also revealed that incorporating multiple features to predict learners' performance could improve results (Carter et al., 2017). Conducting the programming courses on various aspects rely on the use of tools, which generally are operated in isolation. The integration of different tools would support evaluating the relationships among various types of predictors. Moreover, the integration of tools could help in real-time analysis of learner's state resulting in improving the reporting of learners performances.

  The issues related to integration provides opportunity to build sophisticated tools for reporting and enforce the dynamicity and mobility aspects. The tools integration could improve the reporting systems by remotely providing to-date learning status and feedback. Such tools may include the features like auto-messaging, notifications, auto-alerts, and feedback on personalized accounts of learners. It further emphasizes the remote messaging of personalized learning materials by evaluating the knowledge gaps of individual learners.

- **Mapping:** The LAPC research demonstrate the use of various coding parameters for examining different aspects of programming behaviors (as presented in Table 9); however, these parameters are generally not mapped with any programming skills or aspect such as optimization, and problem solving. Consequently, it remained unclear that how exhibiting certain programming behaviors could help in achieving a certain programming skill. It also raises the concerns about mapping the numbers obtained by analyzing the programming behaviors with different levels of learning (Guzmán-Valenzuela et al., 2021). Furthermore, establishing relation between the analyzing aspect with cognitive levels as well as with the psychological traits of learners are rarely observed. Similarly, the assessment items used to assess learners also need to be mapped with the different cognitive levels as it enforces data granularity. This has also been observed as a neglected aspect while producing the data for LAPC despite most of the studies used assessment data where the information on learners' cognition is generally lacked.

  LA performed on the social learning behaviors poses promising dimensions to examine learning; however, mapping such behaviors with psychological aspects would aid the learning support by assisting the psychological needs of novice programmers. Identifying the relationship of coding analysis parameters with the specific traits of programmers could help in devising optimized interventions. In recent past, an increasing trend of learning through gamification has been observed in computer science education (Ahmad et al., 2020), which could be a potential source to understand the psychological aspects of learning.
- **Data integrity:** The target variables in most of the prediction focused studies are the performances in exams or the final grades (as shown in Table 5), which seem to be based on the implicit assumption about the authenticity of the assessment process in terms of ranking the learner's knowledge. The assessment data of students' attempts passes through a grading procedure, which involves judgments of the graders. The differences in grading have been found when more than one evaluator graded the same solutions of the given programming problem (Albluwi, 2018). LA performed on assessment data raises concerns of the data integrity where the measures taken to address the authenticity of rubrics are generally not discussed.

  The integrity of data can be examined by investigating techniques to evaluate the quality of data. The parameters to evaluate the quality of data need to be identified for quality assurances. Moreover, the techniques like Cronbach's alpha test, correlation analysis need to be applied to evaluate the data quality for examining the reliability and authenticity of data for performing analysis.

- **Standardization:** A gap is identified when a standardized technique working appropriately in specific study areas does not present comparable results in the context of programming courses (Effenberger et al., 2020). It triggers the need to explore the applicability of learning theories in programming scenarios and the need analysis of specialized learning theories by perusing the specific aspects of programming courses.

Moreover, the LAPC studies need to highlight how these studies are supported by the standardized learning theories.

In addition, the selected studies used a diverse range of different datasets related to different institutes, which would have been operated on different environments. This calls for establishing and applying the criteria for prior evaluation of data on common parameters to understand the robustness of programming courses research (Omer et al., 2021). Furthermore, the methodical potential of proposed solutions of the underlying problems could help in replication and reproduction analysis of the studies. This would be useful in identifying the applicability of solutions on different cohorts as standardized solution of the given or similar problems. Furthermore, some of the data related issues also indicate the need of building standard measures including techniques and parameters to evaluate the quality of data.

- **Pattern recognition:** The pattern recognition has been identified as one of the predominant area of study in LA research (Du et al., 2021). However, in case of programming focused LA studies, this seems to be a deficient area of research (as depicted through Table 5). Early detection of inappropriate learning patterns could help to timely solve the learning difficulties. Pattern recognition can play a vital role in identifying the appropriate/inappropriate behaviors before advancing to learn relatively higher level programming concepts. Efforts are required to identify the various learning and behavioral patterns in order to help instructors devise and deliver appropriate and timely interventions.

The concerns related to pattern recognition reflect the need to analyze the identification of patterns exhibiting learning behaviors of specific groups of learners. It would be useful to find the learning or behavioral patterns that would lead to undesirable outcomes. In addition, the classification of learning patterns and their relationships with learning outcomes of the course could help in segregating the effective or ineffective learning patterns of novice programmers.

7 Limitations

Some limitations related to this review are listed below:

- The search string has been developed to extract the most relevant research articles. A number of keywords has been chosen to search the papers from the digital repositories. However, there may be alternate keywords or synonyms that might alter the results. In addition to searching the papers from digital repositories, the backward snowballing has been performed to mitigate the risk of missing any relevant study.

- Papers from various sources are examined and selected by authors who used university's subscription of the research portals to search articles. Therefore, some papers might have overlooked due to the limited access to the sources.
- The shortlisting and classification was carried out by authors of this paper and reviewed by two independent reviewers. In case of disagreements in the classification results, detailed discussions were made until reaching the consensus. The value of Kappa coefficient was 0.92, which shows high agreement.

8 Conclusions

This study presented the review of the state-of-the-art LA in programming courses by carefully searching, and synthesizing the LAPC research articles published in well-reputed research venues. The searching of the research articles was performed through a number of prominent research portals where the main source of research articles for LAPC were IEEE Xplore and ACM digital library. After going through a multi-stage and systematic shortlisting process, 77 articles were finalized for review. The investigating areas of for this study were established by considering the five-steps LA model, which include capture, report, predict, act, and refine. These steps are explored through a number of significant dimensions including types, combinations and sources of data, techniques of reporting the data to the audience, interventions reported on results, and the refinements attempted in the LA steps. The results of the review reveal that the assessment and behavioral are the most used types of data for LAPC where the cognitive aspects of data for analysis of learners' performances can improve precisions in results as presents aspects that are more granular and specific. Moreover, the dynamic and real time feedback would help learners to cover their learning gaps before learning more complex programming concepts. Furthermore, the mapping of programming behaviors with different skills of programmers could help in detailed analysis of learning. Furthermore, efforts are required to determine frequent occurring patterns and classify those according to their effectiveness in meeting the learning objectives of programming courses. These aspects would help in achieving better accuracies in predictions and would support in reporting as well as devising interventions. The results also indicate that the LAPC research investigated other dimensions in addition to those illustrated by the LA model to examine the applicability of different techniques and to evaluate the specific aspects of learning. This work will be helpful for the researchers to optimize the future research dimensions as per the highlighted gaps and concerns of the LAPC research.

Appendix 1

Table 10 Classification and quality scoring

Ref.no	Sources		Analysis facets	Quality Scoring					
Year	Publication channel	LA steps involved	Type of data	(a)	(b)	(c)	(d)	Total	
Caspari-Sadeghi, 2022	2022	Conference	Capture	ASD, BHD	0	0.5	1	0.5	2
Zheng et al., 2022	2022	Journal	Capture	BHD	1	1	1	0.5	3.5
Grawemeyer et al., 2022	2022	Journal	Capture	ASD, BHD, INR	0.5	1	1	0.5	3
Veerasamy et al., 2022	2022	Journal	Capture, Predict	ASD	0.5	0.5	1	1	3
Khan et al., 2021	2021	Journal	Capture, Predict	ASD, BKD, BHD	0.5	1	1	1	3.5
Pereira et al., 2021	2021	Journal	Capture, Predict	BHD	1	0.5	1	1	3.5
Jamjoom et al., 2021	2021	Journal	Capture, Predict	ASD	1	1	0.5	1	3.5
Hellings & Haelermans, 2020	2020	Journal	Capture, Report, Predict	ASD, BKD, BHD	1	1	1	1	4
Pereira et al., 2020	2020	Journal	Capture, Predict	BHD	0.5	1	1	1	3.5
Omer et al., 2020	2020	Journal	Capture, Predict	ASD	1	1	1	1	4
Chung & Hsiao, 2020	2020	Conference	Capture	ASD, BHD	0.5	0.5	1	0.5	2.5
Effenberger et al., 2020	2020	Conference	Capture, Predict	ASD	0.5	0.5	1	1	3
Ninrutsirikun et al., 2020	2020	Conference	Capture, Predict	ASD, BHD, BKD	1	1	1	1	4
Liao et al., 2019	2019	Journal	Capture, Predict	ASD, BHD	1	1	1	1	4
McCall & Kölling, 2019	2019	Journal	Capture	BHD, INR	0.5	0.5	1	0.5	2.5
Ullah et al., 2019	2019	Journal	Capture	ASD	0.5	0.5	0.5	0.5	2
Ureel & Wallace, 2019	2019	Conference	Capture, Report	BHD	1	1	1	1	4
Yeomans et al., 2019	2019	Journal	Capture	INR	1	1	1	0.5	3.5
Albluwi, 2018	2018	Journal	Capture	ASD	0.5	1	0.5	1	3
Azcona et al., 2018	2018	Conference	Capture, Report, Predict, Act, Refine	ASD, BHD, BKD, INR	1	1	1	1	4
Bhatia et al., 2018	2018	Conference	Capture	BHD	1	1	1	0.5	3.5
Chaweewan et al., 2018	2018	Conference	Capture	BHD	0.5	0.5	0.5	0.5	2
Esteero et al., 2018	2018	Conference	Capture	ASD	0	1	0.5	0.5	2
Gomes & Correia, 2018	2018	Conference	Capture	ASD	0	0.5	0.5	0.5	1.5
King, 2018	2018	Conference	Capture	ASD, INR	0.5	1	1	0.5	3

Table 10 (continued)

Ref.no	Sources		Analysis facets	Quality Scoring					
Year	Publication channel	LA steps involved	Type of data	(a)	(b)	(c)	(d)	Total	
Koong et al., 2018	2018	Conference	Capture, Predict, Act	BHD	1	1	1	1	4
Lagus et al., 2018	2018	Journal	Capture, Predict, Refine	BHD, BKD	1	0.5	1	1	3.5
Lin et al., 2018	2018	Journal	Capture	ASD	0.5	0.5	0.5	0.5	2
Marcolino et al., 2018	2018	Conference	Capture	BHD, INR	0	0.5	0.5	0.5	1.5
Premchaiswadi et al., 2018	2018	Conference	Capture, Predict	ASD, BHD	1	1	1	1	4
Santana et al., 2018	2018	Conference	Capture	INR	0	0.5	0.5	0.5	1.5
Turner et al., 2018	2018	Journal	Capture	ASD, INR	0.5	1	1	0.5	3
Wainer & Xavier, 2018	2018	Journal	Capture	ASD	0.5	1	1	0.5	3
Ahadi et al., 2017	2017	Journal	Capture, Predict	ASD, BHD	0.5	1	1	1	3.5
Ahmad et al., 2017	2017	Conference	Capture, Predict	ASD	0.5	1	0.5	1	3
Al-Rifaie et al., 2017	2017	Conference	Capture, Predict	ASD, BHD	0.5	1	1	1	3.5
Carter et al., 2017	2017	Journal	Capture, Predict, Refine	ASD, BHD	1	1	1	1	4
Castro-Wunsch et al., 2017	2017	Conference	Capture, Predict, Refine	BHD	1	1	1	1	4
Delev & Gjorgjevikj, 2017	2017	Conference	Capture	ASD, BHD	0.5	0.5	0.5	0.5	2
Dorodchi et al., 2017	2017	Conference	Capture	ASD	0.5	0.5	0.5	0.5	2
Echeverría et al., 2017	2017	Journal	Capture	BHD	0.5	1	1	0.5	3
España-Boquera et al., 2017	2017	Conference	Capture	BHD	0.5	1	1	0.5	3
Estey et al., 2017	2017	Conference	Capture	ASD, BHD	0.5	1	1	0.5	3
Fu et al., 2017	2017	Conference	Capture, Report	BHD	1	1	1	1	4
Hsiao et al., 2017	2017	Journal	Capture, Report	ASD, BHD, INR	0.5	1	1	1	3.5
Hsu & Mimura, 2017	2017	Conference	Capture, Predict, Act	INR	0.5	0.5	1	1	3
Iqbal Malik & Coldwell-Neilson, 2017	2017	Journal	Capture	ASD, INR	1	1	1	0.5	3.5
Kumar, 2017	2017	Conference	Capture, Predict	ASD, BHD, BKD	0.5	1	1	1	3.5
Seanosky et al., 2017	2017	Conference	Capture, Report	ASD, BHD	0.5	1	1	1	3.5
Seeling & Eickholt, 2017	2017	Conference	Capture	BHD, INR	0.5	1	1	0.5	3
Berges et al., 2016	2016	Conference	Capture	BHD	0.5	1	1	0.5	3

Table 10 (continued)

Ref.no	Sources		Analysis facets	Quality Scoring					
Year	Publication channel	LA steps involved	Type of data	(a)	(b)	(c)	(d)	Total	
Funabiki et al., 2016	2016	Conference	Capture	ASD	0.5	1	1	0.5	3
Gomes et al., 2016	2016	Conference	Capture, Predict	ASD	0.5	1	0.5	1	3
Malliarakis et al., 2016	2016	Journal	Capture	INR	0.5	1	1	0.5	3
Seeling, 2016	2016	Conference	Capture	ASD, BHD	0.5	1	1	0.5	3
Simkins & Decker, 2016	2016	Conference	Capture	INR	0	1	0.5	0.5	2
Ahadi et al., 2015	2015	Journal	Capture	ASD, BHD, BKD	0.5	0.5	1	0.5	2.5
Ashenafi et al., 2015	2015	Conference	Capture, Predict	ASD, BHD	1	1	1	1	4
Carter et al., 2015	2015	Journal	Capture, Predict	BHD	1	1	1	1	4
Hilton & Rague, 2015	2015	Conference	Capture, Report	INR	0	0.5	1	1	2.5
Janke et al., 2015	2015	Conference	Capture	INR	0	0.5	0.5	0.5	1.5
Rosiene & Rosiene, 2015	2015	Conference	Capture	INR, OBD	0.5	0.5	1	0.5	2.5
Scott et al., 2015	2015	Journal	Capture	ASD, INR	0.5	0.5	0.5	0.5	2
Su et al., 2015	2015	Conference	Capture, Refine	ASD	1	1	1	1	4
Ureel & Wallace, 2015	2015	Conference	Capture, Report	ASD, INR	1	1	1	1	4
Wood & Keen, 2015	2015	Conference	Capture	ASD, OBD	0.5	1	0.5	0.5	2.5
Xinogalos, 2015	2015	Journal	Capture	ASD, INR	0.5	1	1	0.5	3
Allinjawi et al., 2014	2014	Journal	Capture	ASD	0.5	0.5	1	0.5	2.5
Doshi et al., 2014	2014	Conference	Capture	ASD	0.5	1	1	0.5	3
Edwards et al., 2014a	2014	Conference	Capture	INR	0.5	1	1	0.5	3
Edwards et al., 2014b	2014	Conference	Capture	BHD, INR	0.5	1	1	0.5	3
Heinonen et al., 2014	2014	Conference	Capture	BHD	0.5	0.5	1	0.5	2.5
Hijon-Neira et al., 2014	2014	Conference	Capture, Refine	ASD, INR	1	1	1	1	4
Rubio et al., 2014	2014	Conference	Capture	ASD, INR	0.5	1	1	0.5	3
Watson et al., 2014	2014	Conference	Capture	BHD, BKD, INR	0	0.5	1	0.5	2
Zur & Vilner, 2014	2014	Conference	Capture	ASD	0	0.5	0.5	0.5	1.5
Blikstein et al., 2014	2014	Journal	Capture, Predict	ASD, BHD	1	1	1	1	4

ASD is abbreviated for assessment data, *BHD* for behavioral data, *INR* for inquiry responses, *BKD* for background data, and *OBD* is abbreviated for observational data

Data availability This work presents a meta-analysis and review of the research articles available on the respective digital repositories.

Declarations

Competing interest There is no competing interest associated to this research.

References

Ahadi, A., Hellas, A., & Lister, R. (2017). A contingency table derived method for analyzing course data. *ACM Transactions on Computing Education (TOCE), 17*(3), 13.

Ahadi, A., Lister, R., Haapala, H., & Vihavainen, A. (2015). Exploring machine learning methods to automatically identify students in need of assistance. In: *Proceedings of the eleventh annual International Conference on International Computing Education Research* (pp. 121–130). ACM.

Ahmad, R., Sarlan, A., Hashim, A. S., & Hassan, M. F. (2017). Relationship between hands-on and written coursework assessments with critical thinking skills in structured programming course. In: *2017 7th World Engineering Education Forum (WEEF)* (pp. 231–235). IEEE.

Ahmad, A., Zeshan, F., Khan, M. S., Marriam, R., Ali, A., & Samreen, A. (2020). The impact of gamification on learning outcomes of computer science majors. *ACM Transactions on Computing Education (TOCE), 20*(2), 1–25.

Ala-Mutka, K. M. (2005). A survey of automated assessment approaches for programming assignments. *Computer Science Education, 15*(2), 83–102.

Albluwi, I. (2018). A Closer Look at the Differences Between Graders in Introductory Computer Science Exams. *IEEE Transactions on Education, 61*(3), 253–260.

Allinjawi, A. A., Al-Nuaim, H. A., & Krause, P. (2014). An Achievement Degree Analysis Approach to Identifying Learning Problems in Object-Oriented Programming. *ACM Transactions on Computing Education (TOCE), 14*(3), 20.

Al-Rifaie, M. M., Yee-King, M., & d'Inverno, M. (2017). Boolean prediction of final grades based on weekly and cumulative activities. In: *2017 Intelligent Systems Conference (IntelliSys)* (pp. 462–469). IEEE.

Ashenafi, M. M., Riccardi, G., & Ronchetti, M. (2015). Predicting students' final exam scores from their course activities. In: *2015 IEEE Frontiers in Education Conference (FIE)* (pp. 1–9). IEEE.

Avella, J. T., Kebritchi, M., Nunn, S. G., & Kanai, T. (2016). Learning analytics methods, benefits, and challenges in higher education: A systematic literature review. *Online Learning, 20*(2), 13–29.

Azcona, D., Hsiao, I. H., & Smeaton, A. F. (2018). Personalizing computer science education by leveraging multimodal learning analytics. In: *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1–9). IEEE.

Carter, A. S., Hundhausen, C. D., & Adesope, O. (2017). Blending measures of programming and social behavior into predictive models of student achievement in early computing courses. *ACM Transactions on Computing Education (TOCE), 17*(3), 12.

Baker, R. S., & Inventado, P. S. (2014). Educational data mining and learning analytics. In: *Learning analytics* (pp. 61–75). Springer.

Berges, M., Striewe, M., Shah, P., Goedicke, M., & Hubwieser, P. (2016). Towards deriving programming competencies from student errors. In: *2016 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)* (pp. 19–23). IEEE.

Bhatia, S., Kohli, P., & Singh, R. (2018). Neuro-symbolic program corrector for introductory programming assignments. In: *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)* (pp. 60–70). IEEE.

Blikstein, P., Worsley, M., Piech, C., Sahami, M., Cooper, S., & Koller, D. (2014). Programming pluralism: Using learning analytics to detect patterns in the learning of computer programming. *Journal of the Learning Sciences, 23*(4), 561–599.

Carter, A. S., Hundhausen, C. D., & Adesope, O. (2015). The normalized programming state model: Predicting student performance in computing courses based on programming behavior. In: *Proceedings of the eleventh annual International Conference on International Computing Education Research* (pp. 141–150). ACM.

Caspari-Sadeghi, S. (2022). Applying learning analytics in online environments: Measuring learners' engagement unobtrusively. In: *Frontiers in Education*, *7*(1).

Castro-Wunsch, K., Ahadi, A., & Petersen, A. (2017). Evaluating neural networks as a method for identifying students in need of assistance. In: *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 111–116). ACM.

Chatti, M. A., Dyckhoff, A. L., Schroeder, U., & Thüs, H. (2012). A reference model for learning analytics. *International Journal of Technology Enhanced Learning, 4*(5–6), 318–331.

Chaweewan, C., Surarerks, A., Rungsawang, A., & Manaskasemsak, B. (2018). Development of Programming capability framework based on aptitude and skill. In: *2018 3rd International Conference on Computer and Communication Systems (ICCCS)* (pp. 104–108). IEEE.

Chung, C. Y., & Hsiao, I. H. (2020). Investigating patterns of study persistence on self-assessment platform of programming problem-solving. In: *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 162–168).

Delev, T., & Gjorgjevikj, D. (2017). Static analysis of source code written by novice programmers. In: *2017 IEEE Global Engineering Education Conference (EDUCON)* (pp. 825–830). IEEE.

Dorodchi, M., Dehbozorgi, N., & Frevert, T. K. (2017)." I wish I could rank my exam's challenge level!": An algorithm of Bloom's taxonomy in teaching CS1. In: *2017 IEEE Frontiers in Education Conference (FIE)* (pp. 1–5). IEEE.

Doshi, J. C., Christian, M., & Trivedi, B. H. (2014). Effect of conceptual cue based (CCB) practical exam evaluation of learning and evaluation approaches: a case for use in process-based pedagogy. In: *2014 IEEE sixth international conference on technology for education* (pp. 90–94). IEEE.

Du, X., Yang, J., Shelton, B. E., Hung, J. L., & Zhang, M. (2021). A systematic meta-review and analysis of learning analytics research. *Behaviour & Information Technology, 40*(1), 49–62.

Echeverría, L., Cobos, R., Machuca, L., & Claros, I. (2017). Using collaborative learning scenarios to teach programming to non-CS majors. *Computer Applications in Engineering Education, 25*(5), 719–731.

Edwards, S. H., Tilden, D. S., & Allevato, A. (2014a). Pythy: improving the introductory python programming experience. In: *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 641–646). ACM.

Edwards, S. H., Shams, Z., & Estep, C. (2014b). Adaptively identifying non-terminating code when testing student programs. In: *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 15–20). ACM.

Effenberger, T., Pelánek, R., & Cechák, J. (2020). Exploration of the robustness and generalizability of the additive factors model. In: *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge* (pp. 472–479).

España-Boquera, S., Guerrero-López, D., Hermida-Pérez, A., Silva, J., & Benlloch-Dualde, J. V. (2017). Analyzing the learning process (in Programming) by using data collected from an online IDE. In: *2017 16th International Conference on Information Technology Based Higher Education and Training (ITHET)* (pp. 1–4). IEEE.

Estey, A., Keuning, H., & Coady, Y. (2017). Automatically classifying students in need of support by detecting changes in programming behaviour. In: *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 189–194). ACM.

Esteero, R., Khan, M., Mohamed, M., Zhang, L. Y., & Zingaro, D. (2018). Recursion or iteration: Does it matter what students choose?. In: *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 1011–1016). ACM.

Farooq, M. S., Hamid, A., Alvi, A., & Omer, U. (2022). Blended Learning Models, Curricula, and Gamification in Project Management Education. *IEEE Access, 10*, 60341–60361. https://doi.org/10.1109/ACCESS.2022.3180355

Farooq, M. S., Omer, U., Tehseen, R., & Nisah, I. U. (2021). Software project management education: a systematic review. *VFAST Transactions on Software Engineering*, *9*(3), 102–119.

Funabiki, N., Ishihara, N., & Kao, W. C. (2016). Analysis of fill-in-blank problem solution results in Java programming course. In: *2016 IEEE 5th Global Conference on Consumer Electronics* (pp. 1–2). IEEE.

Fu, X., Shimada, A., Ogata, H., Taniguchi, Y., & Suehiro, D. (2017). Real-time learning analytics for c programming language courses. In: *Proceedings of the Seventh International Learning Analytics & Knowledge Conference* (pp. 280–288). ACM.

Gomes, A., Correia, F. B., & Abreu, P. H. (2016). Types of assessing student-programming knowledge. In: *2016 IEEE Frontiers in Education Conference (FIE)* (pp. 1–8). IEEE.

Gomes, A., & Correia, F. B. (2018). Bloom's taxonomy based approach to learn basic programming loops. In: *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1–5). IEEE.

Grawemeyer, B., Halloran, J., England, M., & Croft, D. (2022). Feedback and engagement on an introductory programming module. In: *Computing Education Practice 2022* (pp. 17–20).

Guzmán-Valenzuela, C., Gómez-González, C., Rojas-Murphy Tagle, A., & Lorca-Vyhmeister, A. (2021). Learning analytics in higher education: A preponderance of analytics but very little learning? *International Journal of Educational Technology in Higher Education, 18*(1), 1–19.

Hashima, A. S., Hamoud, A. K., & Awadh, W. A. (2018). Analyzing students' answers using association rule mining based on feature selection. *Journal of Southwest Jiaotong University*, 53(5).

Hellings, J., & Haelermans, C. (2020). The effect of providing learning analytics on student behaviour and performance in programming: a randomised controlled experiment. *Higher Education*, 1–18.

Heinonen, K., Hirvikoski, K., Luukkainen, M., & Vihavainen, A. (2014). Using CodeBrowser to seek differences between novice programmers. In: *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 229–234). ACM.

Hijon-Neira, R., Velázquez-Iturbide, Á., Pizarro-Romero, C., & Carriço, L. (2014). Merlin-know, an interactive virtual teacher for improving learning in Moodle. In: *2014 IEEE Frontiers in Education Conference (FIE) Proceedings* (pp. 1–8). IEEE.

Hilton, S., & Rague, B. (2015). Is video feedback more effective than written feedback?. In: *2015 IEEE Frontiers in Education Conference (FIE)* (pp. 1–6). IEEE.

Hsiao, I. H., Huang, P. K., & Murphy, H. (2017). Integrating programming learning analytics across physical and digital space. *IEEE Transactions on Emerging Topics in Computing*.

Hsu, W. C., & Mimura, Y. (2017). Understanding the secondary digital gap: Learning challenges and performance in college introductory programming courses. In: *2017 IEEE 9th International Conference on Engineering Education (ICEED)* (pp. 59–64). IEEE.

Hundhausen, C. D., Olivares, D. M., & Carter, A. S. (2017). IDE-based learning analytics for computing education: A process model, critical review, and research agenda. *ACM Transactions on Computing Education (TOCE), 17*(3), 1–26.

Ihantola, P., Vihavainen, A., Ahadi, A., Butler, M., Börstler, J., Edwards, S. H., ... & Rubio, M. Á. (2015). Educational data mining and learning analytics in programming: Literature review and case studies. In: *Proceedings of the 2015 ITiCSE on Working Group Reports* (pp. 41–63). ACM.

Iqbal Malik, S., & Coldwell-Neilson, J. (2017). Impact of a new teaching and learning approach in an introductory programming course. *Journal of Educational Computing Research, 55*(6), 789–819.

Jamjoom, M., Alabdulkreem, E., Hadjouni, M., Karim, F., & Qarh, M. (2021). Early prediction for at-risk students in an introductory programming course based on student self-efficacy. *Informatica*, *45*(6).

Janke, E., Brune, P., & Wagner, S. (2015). Does outside-in teaching improve the learning of object-oriented programming?. In: *Proceedings of the 37th International Conference on Software Engineering* (Volume 2, pp. 408–417). IEEE Press.

Khalil, M., Prinsloo, P., & Slade, S. (2022). A Comparison of learning analytics frameworks: a systematic review. In: *LAK22: 12th International Learning Analytics and Knowledge Conference* (pp. 152–163).

Khan, I., Ahmad, A. R., Jabeur, N., & Mahdi, M. N. (2021). Machine learning prediction and recommendation framework to support introductory programming course. *International Journal of Emerging Technologies in Learning*, *16*(17).

Kaliisa, R., Kluge, A., & Mørch, A. I. (2022). Overcoming challenges to the adoption of learning analytics at the practitioner level: A critical analysis of 18 learning analytics frameworks. *Scandinavian Journal of Educational Research, 66*(3), 367–381.

King, C. E. (2018). Feasibility and acceptability of peer assessment for coding assignments in large lecture based programming engineering courses. In: *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1–9). IEEE.

Kitchenham, B., & Charters S. (2007). Guidelines for performing systematic literature reviews in software engine,ering. Retrieved from https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf. Accessed 7 Jun 2022

Koong, C. S., Tsai, H. Y., Hsu, Y. Y., & Chen, Y. C. (2018). The Learning effectiveness analysis of JAVA programming with automatic grading system. In: *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 2, pp. 99–104). IEEE.

Kumar, A. N. (2017). Learning styles of computer science I students. In: *2017 IEEE Frontiers in Education Conference (FIE)* (pp. 1–6). IEEE.

Lagus, J., Longi, K., Klami, A., & Hellas, A. (2018). Transfer-Learning Methods in Programming Course Outcome Prediction. *ACM Transactions on Computing Education (TOCE), 18*(4), 19.

Liao, S. N., Zingaro, D., Thai, K., Alvarado, C., Griswold, W. G., & Porter, L. (2019). A robust machine learning technique to predict low-performing students. *ACM Transactions on Computing Education (TOCE), 19*(3), 18.

Lin, C. C., Liu, Z. C., Chang, C. L., & Lin, Y. W. (2018). A Genetic algorithm-based personalized remedial learning system for learning object-oriented concepts of Java. *IEEE Transactions on Education*.

Mangaroska, K., & Giannakos, M. N. (2018). *Learning analytics for learning design: A systematic literature review of analytics-driven design to enhance learning*. IEEE Transactions on Learning Technologies.

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2016). CMX: The effects of an educational MMORPG on learning and teaching computer programming. *IEEE Transactions on Learning Technologies, 10*(2), 219–235.

Marcolino, A. S., Santos, A., Schaefer, M., & Barbosa, E. F. (2018). Towards a Catalog of Gestures for M-learning Applications for the Teaching of Programming. In: *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1–9). IEEE.

McCall, D., & Kölling, M. (2019). A New Look at Novice Programmer Errors. *ACM Transactions on Computing Education (TOCE), 19*(4), 38.

Ninrutsirikun, U., Imai, H., Watanapa, B., & Arpnikanondt, C. (2020). Principal Component clustered factors for determining study performance in computer programming class. *Wireless Personal Communications*, 1–20.

Omer, U., Farooq, M. S., & Abid, A. (2020). Cognitive Learning Analytics Using Assessment Data and Concept Map: A Framework-Based Approach for Sustainability of Programming Courses. *Sustainability, 12*(17), 6990.

Omer, U., Farooq, M. S., & Abid, A. (2021). Introductory programming course: Review and future implications. *PeerJ Computer Science, 7*, e647.

Ouhbi, S., Idri, A., Fernández-Alemán, J. L., & Toval, A. (2015). Requirements engineering education: A systematic mapping study. *Requirements Engineering, 20*(2), 119–138.

Pardo, A. (2014). Designing learning analytics experiences. Learning analytics: From research to practice (pp. 15–38). New York: Springer.

Pereira, F. D., Fonseca, S. C., Oliveira, E. H., Cristea, A. I., Bellhäuser, H., Rodrigues, L., ... & Carvalho, L. S. (2021). Explaining individual and collective programming students' behavior by interpreting a black-box predictive Model. *IEEE Access*, *9*, 117097–117119.

Pereira, F. D., Oliveira, E. H., Oliveira, D. B., Cristea, A. I., Carvalho, L. S., Fonseca, S. C., ... & Isotani, S. (2020). Using learning analytics in the Amazonas: understanding students' behaviour in introductory programming. *British Journal of Educational Technology*.

Premchaiswadi, W., Porouhan, P., & Premchaiswadi, N. (2018). Process modeling, behavior analytics and group performance assessment of e-learning logs via fuzzy miner algorithm. In: *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 2, pp. 304–309). IEEE.

Qazdar, A., Er-Raha, B., Cherkaoui, C., & Mammass, D. (2019). A machine learning algorithm framework for predicting students performance: A case study of baccalaureate students in Morocco. *Education and Information Technologies, 24*(6), 3577–3589.

Rojas-López, A., Rincón-Flores, E. G., Mena, J., García-Peñalvo, F. J., & Ramírez-Montoya, M. S. (2019). Engagement in the course of programming in higher education through the use of gamification. *Universal Access in the Information Society, 18*(3), 583–597.

Romero, C., & Ventura, S. (2020). Educational data mining and learning analytics: An updated survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10*(3), e1355.

Rosiene, C. P., & Rosiene, J. A. (2015). Flipping a programming course: The good, the bad, and the ugly. In; *2015 IEEE Frontiers in Education Conference (FIE)* (pp. 1–3). IEEE.

Rubio, M. A., Romero-Zaliz, R., Mañoso, C., & Angel, P. (2014). Enhancing an introductory programming course with physical computing modules. In: *2014 IEEE Frontiers in Education Conference (FIE) Proceedings* (pp. 1–8). IEEE.

Santana, B. L., Figueredo, J. S. L., & Bittencourt, R. A. (2018). Motivation of engineering students with a mixed-contexts approach to introductory programming. In: *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1–9). IEEE.

Scott, M. J., Counsell, S., Lauria, S., Swift, S., Tucker, A., Shepperd, M., & Ghinea, G. (2015). Enhancing practice and achievement in introductory programming with a robot Olympics. *IEEE Transactions on Education, 58*(4), 249–254.

Seeling, P., & Eickholt, J. (2017). Levels of active learning in programming skill acquisition: From lecture to active learning rooms. In: *2017 IEEE Frontiers in Education Conference (FIE)* (pp. 1–5). IEEE.

Seeling, P. (2016). Evolving an introductory programming course: Impacts of student self-empowerment, guided hands-on times, and self-directed training. In: *2016 IEEE Frontiers in Education Conference (FIE)* (pp. 1–5). IEEE.

Seanosky, J., Guillot, I., Boulanger, D., Guillot, R., Guillot, C., Kumar, V., ... & Munshi, A. (2017). Real-time visual feedback: a study in coding analytics. In: *2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)* (pp. 264–266). IEEE.

Simkins, D., & Decker, A. (2016). Examining the intermediate programmers understanding of the learning process. In: *2016 IEEE Frontiers in Education Conference (FIE)* (pp. 1–4). IEEE.

Society for Learning Analytics Research. About, (2012). Retrieved from http://www.solaresearch.org/about/. Accessed 10 Mar 2012

Su, X., Wang, T., Qiu, J., & Zhao, L. (2015). Motivating students with new mechanisms of online assignments and examination to meet the MOOC challenges for programming. In; *2015 IEEE Frontiers in Education Conference (FIE)* (pp. 1–6). IEEE.

Tempelaar, D. (2021). *Learning analytics and its data sources: Why we need to foster all of them*. International Conference on Cognition and Exploratory Learning in Digital Age (CELDA).

Turner, S. A., Pérez-Quiñones, M. A., & Edwards, S. H. (2018). Peer Review in CS2: Conceptual Learning and High-Level Thinking. *ACM Transactions on Computing Education (TOCE), 18*(3), 13.

Ullah, Z., Lajis, A., Jamjoom, M., Altalhi, A. H., Shah, J., & Saleem, F. (2019). A Rule-Based Method for Cognitive Competency Assessment in Computer Programming Using Bloom's Taxonomy. *IEEE Access, 7*, 64663–64675.

Ureel, L. C., & Wallace, C. (2015). WebTA: Automated iterative critique of student programming assignments. In: *2015 IEEE Frontiers in Education Conference (FIE)* (pp. 1–9). IEEE.

Ureel II, L. C., & Wallace, C. (2019). Automated critique of early programming antipatterns. In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 738–744). ACM.

Veerasamy, A. K., Laakso, M. J., & D'Souza, D. (2022). Formative assessment tasks as indicators of student engagement for predicting at-risk students in programming courses. *Informatics in Education, 21*(2), 375–393.

Viberg, O., Hatakka, M., Bälter, O., & Mavroudi, A. (2018). The current landscape of learning analytics in higher education. *Computers in Human Behavior, 89*, 98–110.

Wainer, J., & Xavier, E. C. (2018). A Controlled Experiment on Python vs C for an Introductory Programming Course: Students' Outcomes. *ACM Transactions on Computing Education (TOCE), 18*(3), 12.

Watson, C., & Li, F. W. (2014). Failure rates in introductory programming revisited. In: *Proceedings of the 2014 conference on Innovation & technology in computer science education* (pp. 39–44). ACM.

Watson, C., Li, F. W., & Godwin, J. L. (2014). *No tests required: comparing traditional and dynamic predictors of programming success*. Association for Computing Machinery (ACM).

Wood, Z., & Keen, A. (2015). Building worlds: Bridging imperative-first and object-oriented programming in CS1-CS2. In: *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 144–149). ACM.

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: *Proceedings of the 18th international conference on evaluation and assessment in software engineering* (pp. 1–10).

Xinogalos, S. (2015). Object-oriented design and programming: An investigation of novices' conceptions on objects and classes. *ACM Transactions on Computing Education (TOCE), 15*(3), 13.

Yeomans, L., Zschaler, S., & Coate, K. (2019). Transformative and Troublesome? Students' and Professional Programmers' Perspectives on Difficult Concepts in Programming. *ACM Transactions on Computing Education (TOCE), 19*(3), 23.

Zheng, L., Zhen, Y., Niu, J., & Zhong, L. (2022). An exploratory study on fade-in versus fade-out scaffolding for novice programmers in online collaborative programming settings. *Journal of Computing in Higher Education*, 1–28.

Zur, E., & Vilner, T. (2014). Assessing the assessment—Insights into CS1 exams. In: *2014 IEEE Frontiers in Education Conference (FIE) Proceedings* (pp. 1–7). IEEE.

Authors and Affiliations

Uzma Omer[1] · Rabia Tehseen[2] · Muhammad Shoaib Farooq[3] · Adnan Abid[3,4]

Rabia Tehseen
rabia.tehseen@ucp.edu.pk

Muhammad Shoaib Farooq
shoaib.farooq@umt.edu.pk

Adnan Abid
adnan.abid@vu.edu.pk

1 University of Education, Township, Lahore, Pakistan

2 University of Central Punjab, Johar Town, Lahore, Pakistan

3 University of Management and Technology, Johar Town, Lahore, Pakistan

4 Virtual University of Pakistan, Lahore, Pakistan