Cognitive Learning Analytics Using Assessment Data and Concept Map: A Framework-Based Approach for Sustainability of Programming Courses

Uzma Omer 1,2,*, Muhammad Shoaib Farooq 1 and Adnan Abid 1

1 Department of Computer Science, University of Management and Technology, Johar Town, Lahore 54782, Pakistan; shoaib.farooq@umt.edu.pk (M.S.F); adnan.abid@umt.edu.pk (A.A.)
2 Department of Information Sciences, University of Education, Lahore, 54770, Pakistan
* Correspondence: f2016288001@umt.edu.pk

Received: 30 July 2020; Accepted: 25 August 2020; Published: 27 August 2020

Abstract: Students of initial level programming courses generally face difficulties while learning the programming concepts. The learning analytics studies, in these courses, are mostly anecdotal on the aspect of assessment as less or no attention is given to assess learning at various cognitive levels of specific concepts. Furthermore, the existing work reflects deficiencies in examining the effect of learners’ cognitive performance on subsequent stages of the course. This gap needs to be addressed by introducing more granular and methodical approaches of cognitive analysis for sustaining the programming courses effectively in computer science and associated disciplines. In this article, a framework-based approach is proposed for cognitive learning analytics on the concepts taught in initial level programming courses. The framework serves as a platform that provides structure to the concept data using the technique of concept mapping and examines learners’ cognitive propagation on related concepts using assessment data. Learners’ performance prediction has been examined on relatively higher-level programming concepts through the metrics established from the cognitive maps of learners, acquired by deploying the related layers of framework. Overall maximum prediction accuracy range obtained was 64.81% to 90.86%, which was better than the prediction accuracies presented in most of the related studies.

Keywords: learning analytics; sustainability; cognition; programming; performance prediction

1. Introduction

The analysis of educational data has been an area of interests for researchers as it reveals information that leads to channelize the efforts of both acquiring and imparting knowledge. Immense effort has been made to analyze learning for finding any hidden information from the streams of data continuously being produced from educational systems [1]. Initial level programming courses are specifically significant as these courses make the basis of higher-level courses in computer science and related study domains. However, most of the students find difficulties in learning the concepts taught in these courses [2]. Learning analytics thus play a vital role to examine the new facets of these courses that could be utilized for improvements in the related domains. Despite the continuous attempts to analyze various aspects of these courses [3], efforts are still required for more granular level analysis covering the detailed aspects of learning in contrast to the analysis relating to the broader aspects. The learning analysis performed through assessment data is mostly anecdotal where the impact of analysis is examined on final exams or grades [4,5] while less work is done to find impact of analysis on subsequent stages of the course covering one or more concepts [6,7]. Furthermore, concept analysis is performed to examine learning at various levels of cognition by applying Bloom’s taxonomy [8,9];
however, work is still required in the direction of analyzing the cognitive propagation of learning of the subsequent higher-level concepts of programming. The precise efforts are needed to explore how cognitive levels of learners on set of concepts could relate to the learners’ cognition on relatively advanced level programming concepts. One of the main hurdles in this direction is the lack of defined structure and connection between the components and stages of the course which are covering different concepts. This study presents a framework for cognitive learning analytics that links components of courses to examine cognitive learning on related concepts. The framework utilizes the technique of concept mapping to provide structure to the course concepts and provides a basis for exploring the cognitive learning relations of the concepts [10]. Furthermore, the framework provides a systematic and structured mechanism for transforming the continuously producing data through assessments of programming courses to find any meaningful information that could be useful for improvements.

The framework is evaluated using the assessment data of two programming courses to predict the cognitive performance of students as they relate to higher level programming concepts. The assessment data are generated at various levels of cognition by applying the revised Bloom’s taxonomy [11], which was originally presented with different sets of verbs and nouns to assess and classify students’ cognition [12]. Another taxonomy of cognitive skills called Structure of the Observed Learning Outcome (SOLO) describes complexities through quantitative and qualitative measures and classifies students’ cognition on the basis of one or more aspects of learning [13]. We chose to use the revised version of Bloom’s taxonomy as in our opinion it presents a comprehensible guide to assess and classify the cognitive progression of learners in terms of lower to higher level thinking.

The rest of this paper has been organized in the following manner. Section 2 highlights the related work while Section 3 presents the cognitive learning analytics framework. Section 4 describes the case study for evaluating the framework. The discussion has been provided in Section 5 while the limitations are elaborated in Section 6. Finally, the conclusion and future implications are presented in Section 7.

2. Related Work

The main motivation behind this study is to discover a framework-based approach for examining the cognitive propagation of learners on related concepts that emphasizes granularity of analysis. We examined literature, most of which relates to the initial level programming courses and perused the coverage of aspects related to this work.

2.1. Frameworks of Learning Analytics and Cognitive Assessment

Various frameworks of learning analytics are presented to analyze learning generally as well as specifically on different cognitive levels. The authors in [14] devised a learning analytics framework, which is comprised of six dimensions: Stakeholders, internal limitations, external constraints, instruments, objectives, and data. The stakeholders include students, teachers, and institutions, internal limitations are based on factors like competencies, external constraints relate to educational conventions and norms, and instruments are used to tailor information that could be used for analysis. Different types of data can be used for analysis, which mainly emerge from the students’ interactions with the system. All the specified dimensions of learning analytics are significant and are applied in their precise forms while evaluating the framework presented in this article. The proposed framework serves as a platform that has potential to analyze learning for examining both reflective and predictive objectives as presented in the related work. The work in [15] presented a framework to analyze learning from different aspects of facilitating the stakeholders. The aspects highlighted for instructors include knowledge assessment, types of errors, engagements, expectations, experimentation, and course design while the student-related aspects include planning, monitoring, and improvements. Tool support is provided to give an insight into the stated aspects by exploring different parameters like submitted assignments, and time spent on certain sections of the tool. The presented approach was based on tool-based analysis, which would help stakeholders to perform their respective roles. The research
in [16] proposed a framework for cognitive assessments, which is based on Bloom’s taxonomy. The framework was meant to examine student learning at various cognitive levels of programming concepts. The significance of assessing learners on specific concepts at various levels of cognition was highlighted. This work complements our approach, which emphasizes the role of various cognitive levels to precisely assess students’ cognitive learning on specific concepts by applying the Bloom’s taxonomy. However, our proposed framework is augmented with the use of a concept map, which establishes an organized structure of concepts, taught at various stages of programming courses. Through this approach, the cognitive dependencies of related concepts could precisely be analyzed at various levels using assessment data.

2.2. Learning Analytics and Predictions in Programming Courses

Learning analytics in programming courses have been performed from various dimensions to find useful information that could be utilized for the improvements of related environments. The data sources of learning analytics and predictions in programming courses generally relate to the students enrolled in courses taught at various institutes where the students’ data of one or more institutes are used in prediction and analysis studies [6,7,17]. Various types of data have been used for analysis of programming courses, which include past academic record [6], social learning data like number of comments and posts [7], log data of submissions and attempts of answering questions [18], and programming data like compilation states and errors [19]. The assessment data of students’ performance in assignments, in-class activities, and mid-term exam is perused to identify a stronger predictor of final exam performance where the performance in mid-term exam turned out to be the strongest predictor among the others [15]. Moreover, the assessment data of peer reviews and comments has also been used for analysis [20]. Furthermore, the cognitive and non-cognitive factors have been examined to find the appropriate set of determinants of students’ performance [21]. Analysis studies, which lead to predict students’ performance, generally, examine impacts of analysis on final exams or grades [17,20,21]. Very few studies target performances on subsequent weeks or forthcoming assignments to examine impact of analysis. The authors in [6] presented an automated predictive model to predict students’ weekly performance. The study in [7] analyzed programming to predict performance by perusing the impact of analysis on students’ assignments.

2.3. Conceptual Analysis

Studies are conducted to analyze programming concepts from various perspectives. The analysis dimensions of programming concepts include investigation of the threshold concepts [22], difficulties students face in understanding specific concepts [23,24], and the choice of programming concepts applied while solving the problems [25]. Difficulty of concepts as one of the investigating areas for devising course related activities is highlighted in [15]. The analysis of individual concepts could give more precise findings than analyzing course data and generalizing the findings for all concepts of the course. The difficult concepts, from the students’ end, need to be examined early so that effective interventions could be planned before teaching the subsequent higher-level concepts.

2.3.1. Cognitive Analysis of Concepts and Related Assessment Data

Assessment data generated by applying Bloom’s taxonomy reflect different levels of learners’ cognition. Studies examined the students’ cognition, through Bloom’s taxonomy, for different purposes. The research in [26] presented rules for auto-identifying the demonstrated cognitive levels of students through their programs. The developed programs were mapped to different cognitive levels on the basis of devised rules. The study was conducted on the concepts of selection, repetition, and modularity. The study in [23] precisely assessed students learning on the concept of programming loops at different levels of cognition. This study helped in identifying students’ weaknesses on the given concept. The study finds an increase in students’ motivation through the specified approach as it helped them in understanding the more detailed insight into their knowledge. The authors in [9] examined the
relationship of written assessment with Bloom’s taxonomy-based final exams and found a strong correlation between the two variables. The work in [27] perused assessment items developed to assess learners on different level of cognition and find most effective assessment items that could assess students learning by analyzing the data of different tests conducted on the basis of developed assessment items.

Although assessment data and cognitive analysis has been examined from various dimensions, yet rare work is seen that examines the cognitive propagation analysis of related programming concepts, which makes an area that needs attention. In this article, we have presented a structured approach of assessing cognitive propagation on a number of programming concepts, which emphasize the granularity and specificity of learning analytics using Bloom’s taxonomy-based assessment data.

2.3.2. Concept Mapping

Establishing linkages between the concepts taught at various stages of conducting the course is significant to analyze the effect of learning a concept on the related concepts. In the literature, the concept mapping technique has been used to link the concepts taught at various stages of the course. The authors of [8] examined the development of higher level thinking on the programming concepts of Abstraction, Decomposition, and Encapsulation. They perused the concept mapping technique to assess learning through peer review process. The authors in [28] analyzed the use of concept mapping in computer science courses through systematic mapping study. The application of concept mapping technique to support teaching and learning by organizing the knowledge has been highlighted. Moreover, the increasing diversity of using concept map in the courses of programming and software engineering is also observed. The concept map is used as an assessment instrument by examining the way students link various concepts. The work in [29] examined personalized learning approaches through pre-requisite concept maps of learners by identifying the learning gaps of specific concepts. The work in [30] used concept mapping technique to find students understanding about the I/O topic. The concept maps for hardware components and I/O communication were developed and used as standard approaches to assess whether students are able to link various concepts appropriately. The research in [31] perused the application of concept map to develop a programming task. The developed concept maps were based on the set of concepts that are required to solve specific programming problem. This technique would help in determining the understanding of all the concepts involved in solving a given problem. The authors in [32] examined the use of concept mapping in teaching physics. The linking phrases among the concepts were kept blank, which were required to be filled in by the students. The concept mapping was identified as a useful technique for removing the misconceptions. The study in [33] presented a survey on the concept map scoring by exploring ways to rank learning through the developed concept maps. Most of the studies used concept map as assessment tool to examine students’ learning while performing some class activities in which students are required to create, review, or refine the concept maps. The work in [8] examined the learners’ cognition at a high level of thinking by letting them improve the created concept map. The authors in [34] used the concept map-based model of object-oriented programming course to check concept effect analysis. The concept effect propagation was analyzed by scrutinizing the understanding of students on set of related concepts required to solve test items in exams. The demonstrated knowledge of the learners was used to examine the extent of achieving the learning outcomes. The research in [35] perused the technique of scoring the students’ knowledge through parameters like linkages of concepts established in the concept maps and patterns of correct and semi-correct propositions established for making the linkages. However, no specific standard is established to rate students’ knowledge through their developed concept maps [28].

Despite using the concept mapping technique to analyze learning, efforts are required to examine how concept mapping can be used to analyze cognitive relationships among various concepts. Since learning a programming concept depends on the learners’ cognition on one or more related concepts, the structuring of conceptual data is specifically significant in programming courses. The proposed
framework structures the concepts data using the technique of concept mapping and then applies the demonstrated cognitive levels of learners acquired from assessments on the concept map. The resulting structure would then be used to examine learning from different dimensions. This approach will serve as a basis for more granular and dynamic analysis of learning.

3. Framework for Cognitive Learning Analytics Using Assessment Data

One of the main motivations behind devising the framework is exploring a systematic approach of transforming the semi-structured data of students’ interactions with the educational system into an organized structure. Figure 1 presents the high-level abstraction of learning analysis aspects that will be covered by the presented framework.

![Figure 1. Abstraction of learning analytics and related components.](image)

Students’ interactions with education systems are generally produced in semi-structured form. Some transformative services are therefore required to convert the semi-structured data in the form that could be usable for systematic analysis of learning. The intent is to use the structure for finding new information that could be utilized by the stakeholders for improving the related learning or teaching processes. The systematic approach of transforming the data into an organized structure will pave the way of utilizing the continuously producing streams of students’ interactions data that are being generated from various iterations of related courses [36]. The cognitive learning analysis framework is augmented with the use of concept map where the relations of various stages of course are established by linking the concepts taught at various stages of course. This structure serves to examine the learners’ knowledge gain on more granular levels. The framework is presented in Figure 2. It is compartmentalized into three layers where each layer plays specific role in analyzing the course data.

3.1. Course Conduct Layer (A)

This layer handles the course material that is used to conduct the course and the data that are generated by performing assessment activities. The concepts data include the set of concepts that are used to teach the course.

The lecture material presents the ways concepts are taught to the students. Assessment data include assessment items and learners’ attempts. The concepts data generally do not expand on various iterations of courses whereas the assessments data keep on growing, as a new set of data appears on every iteration of course, resulting in continuous increase in the volume of big data.
3.2. Transformation and Mapping Layer (B)

This layer transforms and structures the concepts data by establishing linkages of concepts using the technique of concept mapping. The assessment rules for ranking the learning are devised and applied to the learners’ attempts. An appropriate set of assessment rules representing the threshold values that grades learners’ performances, needs to be devised. These rules are required to be applied to learners’ attempts to find out the demonstrated cognitive levels of learners on various concepts. The learners are thus examined at various cognitive levels as per Bloom’s taxonomy where remember is the lowest level and synthesis is the highest level of cognition. For the proposed framework, it is required that the assessment data include the assessment items addressing different cognitive levels of the related concepts. Therefore, the following conditions (1 and 2) related to assessment data need to be fulfilled. Consider $A^s$ a super set that contains all assessment items of a given course and $A^c$ as a set of assessment items for a given concept $c$, then

$$A^c \subset A^s,$$  \hspace{1cm} (1)

Also, $A^i$ is the set of assessment items of cognitive level $i$ for a given concept $c$, then

$$A^i \subseteq A^c \forall i \in \{1, 2, \ldots, L_{max}\}$$  \hspace{1cm} (2)
where $L_{max}$ is the maximum cognitive level of assessing the learners. The concept map of course augmented with the demonstrated cognitive levels of the learners lead to the personalized cognitive maps of the learners. This transformation of concept map to the personalized cognitive level map is represented in Figure 3.

![Figure 3](image)

**Figure 3.** Transformation of concept map to personalized cognitive level maps of learners.

### 3.3. Knowledge Discovery and Analysis Layer (C)

This layer is meant to identify the hidden information from the cognitive maps obtained from the previous layer. The cognitive level maps of learners will be used in this layer for knowledge discovery. Educational data mining techniques, which include descriptive and predictive analysis, could also be applied to discover knowledge through streams of data producing from the educational systems [37]. The analysis dimensions for instructors would help instructors determine, for example, the cognitive outcomes of the related concepts obtained by performing some teaching activities during the course execution. This would help in identifying whether related activities or teaching methodologies and materials may need to be continued for the next iterations of the courses. Also, if there are any unexpected outcomes, then those areas of the course would specifically be analyzed. The learners, through this proposed framework, would be able to identify weak areas of their understanding levels e.g., if a particular learner demonstrated good performance up to the level of “apply” as per Bloom’s taxonomy but are unable to perform good on the levels beyond that, which sub-concepts he/she needs to improve. This framework will help course evaluators, who are responsible for quality assurance and enhancement purposes. This would help them in identifying for example, whether the intended course objectives based on students’ cognitive learning are achieved up to the expected levels and the specific iterations of the courses in which the course objectives were achieved above or below the expected levels.

### 4. Case Study: Cognitive Performance Prediction

To evaluate the proposed framework, the data of two instances of CS2 course, titled Object Oriented Programming (OOP), are used. The courses at both institutes relate to the semester-based system where each semester spans over 16 weeks.

#### 4.1. Course Conduct (First Layer-A)

The first layer is based on the components related to conducting the course. These components are finalized while planning the course conduct activities and pre-exist in the system. The courses were conducted at two different institutes in the semester of Spring 2019 and have an enrollment of 53 students at institute 1 and 48 students at institute 2. This layer mainly presents the concepts data as well as the data of conducted assessment.

#### 4.1.1. Concepts Data

A number of concepts are included in the both instances of the OOP course. However, based on the past data of the respective courses, the commonly identified concepts on which students’ performances were observed as weaker than the other course concepts, were selected for analysis.
The shortlisted concepts include some higher-level concepts and some basic programming concepts that are prerequisites of OOP course. The shortlisted programming concepts for this study include polymorphism, inheritance, class, overloading, overriding, abstract class, concrete class, methods, variables, scope, signature, and data type.

4.1.2. Assessment Data

The assessment data are generated at various cognitive levels of all the concepts involved in analysis. Unlike most of the related studies, which analyzed students learning at maximum of 2 or 3 concepts involving all levels of Bloom’s taxonomy [23,26], this case study analyzed students on 12 concepts, therefore, we are limiting the application of Bloom’s taxonomy until the fourth level. The assessment items are devised to examine students’ cognition at the first four levels of Bloom’s taxonomy: Remember, understand, apply, and analyze. The remember level relates to recalling, recognizing, or retrieving of knowledge; the understand level depicts comprehending, classifying, explaining, and comparing the concepts or aspects; the apply level examines executing or implementing knowledge; and the analyze level is linked to attributing, differentiating, and organizing the concepts, ideas, or aspects [38]. Considering various approaches of developing and classifying the assessment items for different levels of Bloom’s taxonomy [23,26,39], the following approach is observed for developing the assessment items against the respective cognitive levels of Bloom’s taxonomy:

- **Remember**: For the cognitive level of remember, the assessment items were developed to examine the recalling of the specified programming concepts in which students generally need to answer through narration or basic syntax recall.

- **Understand**: In case of the assessment items for understand, students were required to answer by comprehending the respective concept or related programming aspect.

- **Apply**: For the cognitive level of apply, the assessment items were developed to solve programming problems by applying programming concepts or providing information about the working of the program.

- **Analyze**: The assessment items for the cognitive level of analyze were developed to identify and correct errors by analyzing the given code.

The assessment items were reviewed by two different instructors where the reviewers were required to rate assessment items’ capabilities of assessing students’ cognition at specified levels. The Kappa analysis was performed to find the inter-rater reliability, which turned out to be 0.79, which represents a high degree of correlation.

To assess each target concept at institute 1, the descriptive assessment items of uniform marks were developed for all the four cognitive levels. Hence, a total of 48 assessment items, of descriptive type, were developed to assess 12 concepts at each of the four cognitive levels. In addition to the descriptive type of assessment items, the Multiple-Choice Questions (MCQs) are considered fairly appropriate to assess students learning in programming courses [40] at different levels of Bloom’s taxonomy [39]. To assess the target concepts at institute 2, four MCQs of 1 mark each were created to examine a particular cognitive level of a given concept. Since the cognitive learning at four levels were required to be examined, 16 MCQs for each concept were prepared. Hence, a total of 192 assessment items were developed to test students’ cognition at first four levels of Bloom’s taxonomy in 12 concepts. The learner’s cognition on the target concepts were assessed through a planned series of quizzes conducted at regular intervals during the course. Since lower-level concepts in the concept map are taught early in the course, the quizzes of these concepts are conducted before the subsequent higher-level concepts. As part of the data cleansing process, the data of students who had not attempted all quizzes were removed from the dataset. Hence, the data of 77 students were finalized for analysis. It includes 40 students from institute 1 and 37 students from institute 2.
4.2. Transformation and Mapping (Second Layer-B)

The second layer performs a two-step transformation; first by transforming the semi-structured data of concepts into the organized structure of concept map, and second by converting the concept map into personalized cognitive level maps of individual learners. This step is performed by devising and applying the assessment rules on students’ attempts and mapping the demonstrated cognition of students to the concept map.

4.2.1. Constructing Concept Map

The hierarchical technique is used to construct the concept map for the shortlisted concepts of OOP course for analysis. A hierarchical concept map follows top-down approach of structuring concepts where more general concepts are on top and more specific concepts are at the bottom [41]. This technique for concept map constructions is adopted as programming concepts are based on one or more set of lesser complex concepts e.g., the concept of class is developed through the lesser complex concepts of methods and variables. Similarly, the concept of variables is developed through the concepts of data types and scope. Furthermore, the concepts are arranged according to the way these concepts are taught in the specific OOP course. This means that lesser complex concepts lie on lower levels of the concept map and have been taught at earlier stages of the course as compared to the relatively higher level concepts, which are taught at later stage of the course. The concept map developed for OOP course, that structures the targeted concepts for analysis, is presented in Figure 4.

The presented concept map is built and finalized for experimentation after going through a multi-cycle review and refinement process performed among two instructors who have been teaching OOP course in different institutes. Some of the sub-concepts are taught as part of the higher concepts like abstract, concrete classes, and inheritance. In some cases, the sub-concepts are taught as a whole before teaching the higher-level concepts like methods and variables before class.

4.2.2. Devising and Applying Assessment Rules

The demonstrated cognitive levels of learners are determined by deploying the devised assessment criteria on student’s attempts of assessment items. In case of descriptive assessment items of a given concept, the criteria for grading students attempt were devised by giving weightage to the sub-concepts (of the concept map) or aspects of programing involved for solving the problem. The grading of answers was assessed through Cronbach’s alpha test [42]. It was performed by randomly picking and re-evaluating 10 assessment items where alpha value depicts greater than 0.77, which represents an acceptable number [43]. In case of MCQs, there was only one correct choice, and the marks are given for selecting that among the given choices.
4.3. Knowledge Discovery and Analysis (Third Layer-C)—Cognitive Performance Prediction

The cognitive level maps obtained through layer 2 of the framework is used to examine learners’ cognition from various aspects. For this case study, the metrics are devised to examine and predict the learners’ performance at subsequent stages of the course covering relatively higher-level concepts. In this regard, we defined the metrics of 1) Cognitive Score (CS), 2) Aggregate Cognitive Score (ACS), and 3) Aggregate Level-wise Cognitive Score (ALCS). The CS is calculated against each concept through the involvement of learners demonstrated cognitions of various levels. It is the sum of weighted percentage of the demonstrated cognitive levels of students on particular concept. The motivation behind establishing the CS metric is to consider students’ cognition at various cognitive levels, instead of one particular cognitive level, while ranking their knowledge gain on particular concepts. Cognitive score of a given concept is the sum of weighted % of marks obtained at each cognitive level. Consider a given concept $x$, the CS of $x$ will be calculated as (3).

$$\text{CS}_x = \sum_{i=1}^{n} \left( \%M_i \times \frac{1}{n} \right)$$ (3)

The value of $n$ ranges from 1 to 6 where 1 represents lowest level of cognition (remember) and 6 represents the highest level of cognition (create) as per Bloom’s taxonomy. In the courses under experiment, the concepts are assessed until the fourth level of Bloom’s taxonomy; therefore, the maximum number of $n$ for these courses is 4.

ACS of a concept $y$ is calculated as (4). It is based on the CS of immediate predecessors of $y$. The motivation behind devising this metric is to examine the ability of sub-ordinate concepts to predict performance of higher-level concepts by involving students’ cognition at various levels. ACS is calculated as the sum of weighted cognitive scores of immediate predecessor sub-concepts $k$. The $m$ is the number of immediate predecessor concepts, which could range from one to less than the total number of concepts in the course.

$$\text{ACS}_y = \sum_{k=1}^{m} \left( \text{CS}_k \times \frac{1}{m} \right)$$ (4)

On the basis of cognitive performance of low level concepts, the ACS is calculated to predict performance at subsequent higher level concepts e.g., to calculate ACS for the concept of Class, it first needs to calculate the cognitive score of each of the immediate predecessor sub-concepts of Class (Methods and Variables) in the concept map. The ACS of Class is then calculated by multiplying cognitive score of Methods and Variables by half. Considering the concept of Class has two immediate predecessor sub-concepts, the value of $m$ is 2.

ALCS of a concept $y$ is calculated through its immediate sub-concepts in the concept map. It is calculated by adding the weighted obtained %marks at particular cognitive level $L$ of all the sub-concepts of $y$. Let $\%MS1$ is % of obtained marks in sub-concept 1 of concept $y$, $\%MS2$ is % of obtained marks in sub-concept 2, and so on; then, ALCS of a concept $y$ at first cognitive level will be calculated as (5), where $u$ could be any number from 2 to one less than the total number of concepts in the course and $L$ presents the cognitive level 1 (remember). Similarly, the ALCS for level 2 (understand), 3 (apply), and 4 (analyze) will be calculated. The weight is driven by the value of $u$ and calculated as $1/u$.

$$\text{ALCS}_y = \sum_{j=1}^{u} \left( \%MS_j^L \times \frac{1}{u} \right)$$ (5)

The ALCS of concept of class at remember level is calculated by adding % of marks obtained in the sub-concepts of methods and variables on the same level where the value of $u$ is 2. ALCS is calculated where there exist more than one sub-concept as immediate predecessors.
Application of Metrics to Predict Performance on Relatively Higher-Level Concepts

The ACS and ALCS are examined as predictors to predict performance of individual cognitive levels of higher-level concepts. The linear regression technique has been used for prediction because it is considered as one of the most appropriate approaches in terms of predicting the value of one variable through the other. The prediction studies used different techniques, to calculate the prediction accuracies, which include root mean square error [20], precision, recall, F1 score [17,44], and variance [7]; however, we calculated the prediction accuracies through Mean Absolute Error (MAE) as it is considered as one of the simplified and comprehensible measures in terms of identifying the differences of actual and predicted values. The prediction accuracies acquired through both ACS and ALCS were compared to determine the cognitive level of target concepts (among the four cognitive levels i.e., remember, understand, apply, analyze) that exhibit lesser MAE or higher accuracy. After calculating the ACS and ALCS for cognitive performance prediction of seven target concepts (inheritance, polymorphism, class, overloading, overriding, methods, and variables), the values of MAE are calculated and examined. The MAE1 is calculated on the complete data while MAE2 presents improvements acquired by removing the extreme values from the dataset.

Table 1 presents the overall prediction accuracies of both institutes acquired from ACS and ALCS. In case of ACS, the overall prediction accuracy ranges from 67.9% to 86.40% while it is observed as 74.43% to 89.88% after improvements acquired through MAE2. The ALCS presents overall prediction accuracy ranges between 60.41% to 86.36%, which is obtained as 64.81% to 90.86% through MAE2.

| Target Concept for Performance Prediction | Target Cognitive Levels | Overall (Both Institutes) |
|-----------------------------------------|-------------------------|--------------------------|
|                                         |                         | ACS          | ALCS          |
|                                         | MAE1 | MAE2 | MAE1 | MAE2 |
| Inheritance                             |      |      |      |      |
| Remember                                | 28.34 | 22.03 | 25.60 | 23.46 |
| Understand                              | 28.28 | 20.66 | 27.68 | 19.28 |
| Apply                                   | 31.04 | 25.57 | 26.63 | 20.16 |
| Analyze                                 | 31.08 | 24.30 | 18.79 | 9.14  |
| Polymorphism                            |      |      |      |      |
| Remember                                | 21.96 | 18.50 | 23.59 | 19.67 |
| Understand                              | 28.26 | 23.26 | 23.66 | 18.47 |
| Apply                                   | 32.10 | 24.62 | 26.85 | 22.72 |
| Analyze                                 | 29.84 | 25.27 | 24.88 | 20.9  |
| Class                                   |      |      |      |      |
| Remember                                | 18.20 | 14.38 | 18.43 | 16.11 |
| Understand                              | 24.76 | 20.00 | 29.42 | 25.2  |
| Apply                                   | 22.90 | 17.06 | 27.80 | 25.00 |
| Analyze                                 | 20.84 | 14.01 | 28.06 | 22.98 |
| Overloading                              |      |      |      |      |
| Remember                                | 23.42 | 17.76 | 25.56 | 21.74 |
| Understand                              | 24.70 | 19.53 | 25.36 | 22.18 |
| Apply                                   | 27.31 | 20.79 | 25.42 | 21.16 |
| Analyze                                 | 23.2  | 16.50 | 21.98 | 17.56 |
| Overriding                              |      |      |      |      |
| Remember                                | 26.49 | 21.98 | 28.06 | 26.01 |
| Understand                              | 13.60 | 10.12 | 13.64 | 11.10 |
| Apply                                   | 19.16 | 13.59 | 18.64 | 14.36 |
| Analyze                                 | 23.65 | 19.34 | 20.24 | 17.10 |
| Methods                                 |      |      |      |      |
| Remember                                | 16.15 | 12.77 | 15.46 | 10.04 |
| Understand                              | 15.60 | 11.66 | 17.75 | 14.66 |
| Apply                                   | 14.81 | 10.59 | 15.71 | 9.47  |
| Analyze                                 | 30.72 | 22.76 | 39.59 | 35.19 |
| Variables                               |      |      |      |      |
| Remember                                | 28.09 | 22.88 | 30.60 | 27.31 |
| Understand                              | 24.88 | 20.40 | 27.25 | 26.42 |
| Apply                                   | 15.08 | 11.78 | 14.22 | 12.03 |
| Analyze                                 | 21.48 | 16.57 | 24.675| 27.68 |
Figure 5 presents the comparison of prediction accuracies acquired at various cognitive levels of target concepts. In case of ACS, cognitive learning analysis of two concepts demonstrated maximum prediction at each of the cognitive level of remember, understand, and apply while for one concept, the maximum prediction accuracy is presented at the cognitive level of analyze.

Figure 5. Comparison of overall cognitive prediction accuracies of both institutes.

The improvements in ACS predictions shifted the maximum prediction accuracy of one concept from remember to analyze and one from understand to remember. In the case of ALCS, MAE1 of three out of seven concepts presented maximum prediction accuracy at cognitive level of remember, two of the concepts at the level of analyze, and one concept for each of the cognitive levels of apply and understand. The findings are almost similar in case of MAE2 for ALCS except for two concepts in which the maximum accuracy of prediction is shifted from remember to apply and analyze.

The prediction accuracies, acquired through ACS and ALCS, were further analyzed individually for each of the institutes. Table 2 illustrates the prediction accuracy acquired by applying ACS to the course data of each institute separately.
Table 2. Prediction accuracy through ACS.

| Target Concept for Performance Prediction | Target Cognitive Levels | ACS |
|-------------------------------------------|-------------------------|-----|
|                                           | Institution 1 | Institution 2 |
|                                           | MAE1 | MAE2 | MAE1 | MAE2 |
| Inheritance                               |      |      |      |      |
| Remember                                  | 22.24 | 15.80 | 34.44 | 28.26 |
| Understand                                | 23.93 | 15.79 | 32.62 | 25.54 |
| Apply                                     | 40.53 | 34.30 | 21.55 | 16.85 |
| Analyze                                   | 37.31 | 29.53 | 24.86 | 19.06 |
| Polymorphism                              |      |      |      |      |
| Remember                                  | 24.84 | 20.67 | 19.08 | 16.32 |
| Understand                                | 35.91 | 29.11 | 20.60 | 17.41 |
| Apply                                     | 41.69 | 32.47 | 22.52 | 16.76 |
| Analyze                                   | 40.57 | 34.58 | 19.12 | 15.96 |
| Class                                     |      |      |      |      |
| Remember                                  | 22.01 | 18.60 | 14.39 | 10.16 |
| Understand                                | 27.78 | 23.26 | 21.75 | 16.74 |
| Apply                                     | 30.00 | 22.55 | 15.80 | 11.58 |
| Analyze                                   | 26.80 | 17.82 | 14.89 | 10.20 |
| Overloading                               |      |      |      |      |
| Remember                                  | 28.80 | 21.28 | 18.04 | 14.23 |
| Understand                                | 25.71 | 19.56 | 23.70 | 19.50 |
| Apply                                     | 37.77 | 28.38 | 16.84 | 13.20 |
| Analyze                                   | 28.01 | 18.23 | 18.39 | 14.76 |
| Overriding                                |      |      |      |      |
| Remember                                  | 32.41 | 27.00 | 20.57 | 16.96 |
| Understand                                | 14.98 | 10.54 | 12.21 | 9.70  |
| Apply                                     | 19.49 | 11.49 | 18.83 | 15.69 |
| Analyze                                   | 17.91 | 12.99 | 29.39 | 25.68 |
| Methods                                   |      |      |      |      |
| Remember                                  | 13.12 | 9.46  | 19.19 | 16.08 |
| Understand                                | 16.09 | 11.11 | 15.12 | 12.22 |
| Apply                                     | 14.25 | 10.77 | 15.37 | 10.41 |
| Analyze                                   | 37.08 | 25.60 | 24.36 | 19.92 |
| Variables                                 |      |      |      |      |
| Remember                                  | 33.68 | 28.86 | 22.51 | 16.91 |
| Understand                                | 32.89 | 25.69 | 16.86 | 15.12 |
| Apply                                     | 11.59 | 7.56  | 18.57 | 16.00 |
| Analyze                                   | 36.08 | 29.68 | 6.88  | 3.46  |

For the course of institute 1, the MAE1 ranges from 11.59 to 41.69, which makes the range of prediction accuracy from 58.31% to 88.41% considering all cognitive levels of target concepts. For same course, the accuracy of ACS acquired through MAE2 ranges from 65.48% to 92.44%. For the course at institute 2 the MAE1 through ACS is from 6.88 to 34.44 making the range of prediction accuracy 65.56% to 93.12% while through MAE2, the prediction accuracy for the same course turned out to be 71.74% to 96.54%. Figure 6 presents the cognitive prediction accuracy of target concepts acquired through ACS.

In case of institute 1, the ACS could be able to predict the cognitive performance of inheritance, polymorphism, class, and methods with maximum prediction accuracy acquired at the cognitive level of remember. The maximum cognitive prediction accuracy in overloading and overriding is obtained at the cognitive level of understand while for the concept variables, it is obtained at the cognitive level of apply. Refinements through MAE2 did not change the findings as far as the maximum predictable cognitive level of target concepts is concerned except for the concepts of inheritance and class, which becomes the levels of understand and analyze, respectively. In case of institute 2, the analysis of ACS reflects arbitrary results in terms of maximum prediction accuracies acquired at various cognitive levels of target concepts. The concepts of polymorphism and class presented maximum accuracy at remember while overriding and methods depicted the maximum accuracy at the level of understand. The concepts of inheritance and overloading presented maximum accuracy at the level of apply where variables presented the maximum accuracy at analyze. The MAE2 analysis depicts that the situation
remains same as far as maximum predictable cognitive level is concerned except for polymorphism and method, which shifted to analyze and apply, respectively.

![Figure 6. Comparison of cognitive prediction accuracies of target concepts through ACS.](image)

Table 3 presents the individual results obtained by applying ALCS for each of the institute. For institute 1, through ALCS the MAE1 ranges from 9.18 to 40.78, which makes the prediction accuracy range 59.22% to 90.82% at all cognitive levels of target concepts while through MAE2 the range is from 59.55% to 96.02%. For the course at institute 2, through ALCS the MAE1 ranges from 8.57 to 39.98 which makes the prediction accuracy range 60.02% to 91.43% while through MAE2 the range of prediction accuracy acquired is 60.96% to 95.09%.

| Target Concept for Performance Prediction | Target Cognitive Levels | ALCS | |
|-----------------------------------------|-------------------------|------|------|
| Inheritance | Remember | 19.87 | 17.75 | 31.33 | 29.17 |
| | Understand | 25.34 | 17.84 | 30.02 | 20.71 |
| | Apply | 27.83 | 21.29 | 25.43 | 19.02 |
| | Analyze | 14.01 | 3.98 | 23.57 | 14.31 |
| Polymorphism | Remember | 28.28 | 22.41 | 18.89 | 16.94 |
| | Understand | 26.37 | 19.56 | 20.96 | 17.38 |
| | Apply | 30.96 | 28.34 | 22.74 | 17.10 |
| | Analyze | 29.27 | 23.38 | 20.50 | 18.42 |
| Class | Remember | 22.32 | 21.02 | 14.54 | 11.20 |
| | Understand | 37.31 | 32.87 | 21.53 | 17.53 |
| | Apply | 39.94 | 38.24 | 15.65 | 11.77 |
| | Analyze | 29.45 | 23.32 | 26.68 | 22.63 |
Table 3. Cont.

| Target Concept for Performance Prediction | Target Cognitive Levels | ALCS |
|------------------------------------------|------------------------|------|
|                                          |                        | Institute 1 | Institute 2 |
|                                          |                         | MAE1 | MAE2 | MAE1 | MAE2 |
| Overloading                              | Remember                | 34.44 | 30.54 | 16.69 | 12.94 |
|                                          | Understand              | 27.20 | 23.98 | 23.52 | 20.38 |
|                                          | Apply                   | 35.02 | 30.54 | 15.81 | 11.78 |
|                                          | Analyze                 | 25.85 | 20.11 | 18.11 | 15.02 |
| Overriding                               | Remember                | 34.42 | 33.50 | 21.73 | 18.51 |
|                                          | Understand              | 14.40 | 11.86 | 12.88 | 10.33 |
|                                          | Apply                   | 15.92 | 11.19 | 21.37 | 17.50 |
|                                          | Analyze                 | 9.18  | 7.49  | 31.30 | 26.70 |
| Methods                                  | Remember                | 11.36 | 3.90  | 19.57 | 16.19 |
|                                          | Understand              | 18.13 | 15.57 | 17.37 | 13.75 |
|                                          | Apply                   | 12.97 | 5.42  | 18.44 | 13.52 |
|                                          | Analyze                 | 39.20 | 31.34 | 39.98 | 39.04 |
| Variables                                | Remember                | 39.21 | 38.80 | 21.98 | 15.82 |
|                                          | Understand              | 35.34 | 37.24 | 19.07 | 15.60 |
|                                          | Apply                   | 12.29 | 10.76 | 16.16 | 13.29 |
|                                          | Analyze                 | 40.78 | 40.45 | 8.57  | 4.91  |

Figure 7 presents the cognitive prediction accuracy of target concepts obtained through ALCS. In the case of institute 1, the ALCS for the cognitive performance prediction in the concepts of inheritance, overloading, and overriding showed maximum prediction accuracy at cognitive level of analyze, for class and methods the maximum accuracy is observed at cognitive level of remember. For variables it turned out to be apply and for polymorphism the maximum accuracy is acquired at understand. For MAE2, maximum predicting level becomes apply in case of methods where the other findings turned out to be similar as MAE1.

Figure 7. Comparison of cognitive prediction accuracies of target concepts through ALCS.
In the case of institute 2, the ALCS for the cognitive performance prediction on inheritance and variables showed maximum prediction accuracy at cognitive level of analyze while the cognitive performance prediction on the concepts of polymorphism and class showed maximum accuracy at the level of remember. The concept of overloading presented it at apply and method at understand. For MAE2 the findings are almost the same except for methods in which the maximum prediction accuracy has been shifted to the level of apply with very minute difference, which was understand in case of MAE1.

5. Discussion

The role of education is essential for sustainable development of the related environments [45]. Initial level programming courses are specifically significant in computer science and related disciplines as these courses provide basis to various other courses in undergraduate study programs. Due to high dropout rates, the sustainability of these courses is a concern and therefore makes it an area that needs to be examined from various dimensions [3]. Learning analytics studies are being conducted for initial level programming courses to unveil any hidden aspect that could be used to optimize the efficiencies of related learning environments. However, more granular analytical techniques still need to be explored in order the find detailed aspects that could be utilized for the betterment of related systems. Since learning higher-level programming concepts depends on the understanding of lesser complexed concepts, the early prediction of performance can be explored through the performance of lesser complexed programming concepts. This will pave the way for more meaningful predictions, which would help in devising the effective interventions [46]. In case of programming course, it will specifically assist instructors by giving an insight into the learning gaps before teaching more complex programming concepts. In the literature, most of the studies performed analysis and prediction through anecdotal approaches where the effect of analysis is examined on final stages of the course [17,20,21]; however, information is rarely given on the topic of assessment items in the final exams and the cognitive levels of the assessment items. In this article, we presented a framework-based approach for cognitive learning analytics to peruse students learning more precisely and predict students’ cognitive performances at higher-level programming concepts taught at subsequent stages of the courses. The presented framework emphasizes cognitive learning analytics by going through two transformation steps of semi-structured course data: a) Transforming the concepts data into organized structure of concept map and b) the mapping of concept map and students demonstrated cognitive learning acquired through the assessment data, which resulted into the structure of cognitive maps of learners

The conceptual structure of cognitive map is used to examine cognitive propagation of learning on related higher-level programming concepts. It supported devising two metrics ACS and ALCS, which were analyzed as predictors of cognitive performance of learners on higher-level programming concepts. The cognitive predictions are examined on four levels of Bloom’s taxonomy in the two instances of an OOP course conducted at two different institutes. ACS and ALCS were examined to check how cognitive learning propagates at various levels of cognition of target concepts. The accuracy of predictions is examined through MAE1 and the improvements are performed by removing the extreme values, which give improved accuracy presented as MAE2. It was identified that the devised metrics performed better than most of the related studies; however, there is no particular cognitive level of higher-level programming concepts, which could more precisely be predicted among the others.

We examined the capability of presented metrics to predict cognitive performance by investigating:

(1) How precisely the devised metrics could predict cognitive performance of learners at higher level concepts?

For ACS, the overall prediction accuracy ranges from 67.9% to 86.40% while it is observed as 74.43% to 89.88% after improvements acquired through MAE2. In the case of ALCS, the overall prediction accuracy was observed between 60.41% to 86.36% which is obtained as 64.81% to 90.86%
through MAE2. No significant difference is observed on comparing the overall prediction accuracies obtained through ACS and ALCS. The prediction accuracies in the related studies include ranges of 55% to 81.03% [6], 70% to 96% [17], 36% to 67% [19], and 80% to 85% [47]. Results presented in the related work and those acquired through ACS and ALCS are comparable where ACS and ALCS showed better prediction accuracies in many cases. This finding encourages to replicate and explore the applicability of the study in the similar scenarios.

(2) Are their specific cognitive levels of higher-level concepts that could more precisely be predicted?

No specific cognitive level is identified as the most predictable cognitive level of the target concepts. However, in most of the target concepts, the prediction accuracies at various cognitive level present minute differences, therefore the level depicting maximum accuracy does not reflect any obvious differences among the other levels. The overall ranges of prediction accuracy of all the cognitive levels could thus be considered for performance predictions accuracies of specific higher-level concepts.

6. Limitations

The developed concept map includes those concepts on which we aim to evaluate the framework through cognitive propagation analysis of initial level programming courses. More concepts could be included into the concept map; however, we chose the concepts, identified as common weak concepts, from the data of the previous iterations of the same course. We followed the hierarchical approach of developing concept map, which is based on top-down technique of linking the concepts in which more complexed concepts are placed at higher levels and lesser complexed concepts are placed towards bottom of the concept map. The development process of the concept map includes review cycles conducted by two different reviewers who have been teaching the initial level programming courses at different institutes.

Both descriptive and MCQs types of assessment items were used for assessing students at different levels of cognition. To devise assessment items, the maximum cognitive level up until which course concepts are taught needs to be considered. The developed assessment items were reviewed by two instructors who ranked each assessment item based on its capability of assessing learners on specific level of cognition. The Kappa analysis for inter-rater reliability turned out to be 0.79, which represents a good value to consider the developed assessment items for assessing students at various levels of cognition. However, a better inter-rater correlation value could lead to results with improved precision.

Since a great number of assessment items needed to be developed in order to analyze learners’ cognition at various levels of a number of concepts, we limited the cognitive analysis until the fourth level of cognition as per Bloom’s taxonomy. As solving a programming problem requires applying one or more programming concepts, the assessment items for higher level concepts are prepared in a way that mainly involve the specific sub-concepts in the concept map. The reliability of grading is assessed through Cronbach’s alpha test [42]. It was performed by randomly picking and re-evaluating 10 assessment items. The value of alpha turned out to be greater than 0.77, which represents an acceptable number [43].

7. Conclusions and Future Implications

Sustainability of initial level programming courses, in computer science and related study programs, is specifically significant due to the dependencies of subsequent courses on the programming concepts taught in these courses. Most of the learning analytics studies in these courses are anecdotal on the aspect of analyzing learners at various levels of cognition. The main objective of this work was to present a methodical approach for cognitive learning analytics that emphasizes the granularity of analysis to examine learners’ performance in subsequent stages of the courses. In order to address this objective, a cognitive learning analytics framework is proposed to serve as a platform for systematically analyzing the cognitive propagation of learners on various concepts. The framework provides structure to the concepts data and links the related concepts taught at various stages of the course using the
technique of concept mapping. It then applies the assessment data, generated by using Bloom’s taxonomy, to the concept map. The resulting structure is used to examine the cognitive propagation of learners on the related concepts. The framework is evaluated, in relation to two instances of OOP course taught at two different institutes, by predicting the cognitive performance of learners on the subsequent higher-level programming concepts. The identified prediction accuracy ranges were better than those presented in most of the related work, which has led to the inference that the study can be replicated for other instances of the programming courses.

The application of framework will be explored by examining more analysis dimensions, which could support instructors, learners, and course evaluators to perform their respective roles in educational systems. The framework could potentially assist instructors to find the commonality of cognitive learning patterns to find different performance classes. This could help instructors in devising the optimized interventions for each performance class of students. The learners might have been interested to find individual cognitive learning gaps that could provide them basis to deal with their personalized learning needs. The course evaluators could be interested to know the extent of cognitive learning that will support in achieving the course and program objectives to specific levels.

This work will further be extended to examine the effectiveness of proposed framework by devising metrics that will cover a larger spectrum of analysis. Based on the proposed framework, an automated solution for cognitive learning analytics will also be developed. A personalized learning and feedback system will be modeled to address the individual learning gaps identified by applying the cognitive analysis techniques. Furthermore, the framework applicability could also be explored in the similar domains.

Author Contributions: U.O. Conceptualization, methodology, writing—original draft, writing—review and editing, M.S.F. Methodology, supervision, writing—review and editing, A.A. Methodology, supervision, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mangaroska, K.; Giannakos, M. Learning analytics for learning design: A systematic literature review of analytics-driven design to enhance learning. *IEEE Trans. Learn. Technol.* 2018, 12, 516–534. [CrossRef]
2. Watson, C.; Li, F.W. Failure rates in introductory programming revisited. In Proceedings of the 2014 Conference on Innovation and Technology in Computer Science Education, Uppsala, Sweden, 23–25 June 2014; ACM: New York, NY, USA, 2014. [CrossRef]
3. Luxton-Reilly, A.; Albluwi, I.; Becker, B.A.; Giannakos, M.; Kumar, A.N.; Ott, L.; Paterson, J.; Scott, M.J.; Sheard, J.; Szabo, C.; et al. Introductory programming: A systematic literature review. In Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, Larnaca, Cyprus, 2–4 July 2018; Association for Computing Machinery: New York, NY, USA, 2018.
4. Liao, S.N.; Zingaro, D.; Thai, K.; Alvarado, C.; Griswold, W.G.; Porter, L. A robust machine learning technique to predict low-performing students. *ACM Trans. Comput. Educ. (TOCE)* 2019, 19, 1–19. [CrossRef]
5. Koong, C.S.; Tsai, H.Y.; Hsu, Y.Y.; Chen, Y.C. The Learning Effectiveness Analysis of JAVA Programming with Automatic Grading System. In Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 23–27 July 2018; IEEE: Piscataway, NJ, USA, 2018. Available online: https://ieeexplore.ieee.org/document/8377837 (accessed on 6 July 2019).
6. Azcona, D.; Hsiao, I.H.; Smeaton, A.F. Personalizing computer science education by leveraging multimodal learning analytics. In Proceedings of the 2018 IEEE Frontiers in Education Conference (FIE), San Jose, CA, USA, 3–6 October 2018; IEEE: Piscataway, NJ, USA, 2018.
7. Carter, A.S.; Hundhausen, C.D.; Adesope, O. Blending measures of programming and social behavior into predictive models of student achievement in early computing courses. *ACM Trans. Comput. Educ.* 2017, 17, 1–20. [CrossRef]
8. Turner, S.A.; Pérez-Quiñones, M.A.; Edwards, S.H. Peer Review in CS2: Conceptual Learning and High-Level Thinking. *ACM Trans. Comput. Educ.* 2018, 18, 1–37. [CrossRef]

9. Ahmad, R.; Sarlan, A.; Hashim, A.S.; Hassan, M.F. Relationship between Hands-on and Written Coursework Assessments with Critical Thinking Skills in Structured Programming Course. In Proceedings of the 2017 7th World Engineering Education Forum (WEEF), Kuala Lumpur, Malaysia, 13–16 November 2017; IEEE: Piscataway, NJ, USA, 2017.

10. Novak, J.D.; Musonda, D.A. Twelve-Year Longitudinal Study of Science Concept Learning. *Am. Educ. Res. J.* 1991, 28, 117–153. [CrossRef]

11. Anderson, L.W.; Bloom, B.S. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*; Pearson Education: New York, NY, USA, 2001; pp. 25–195.

12. Bloom, B.S.; Engelhart, M.D.; Furst, E.J.; Hill, W.H.; Krathwohl, D.R. *Taxonomy of Educational Objectives: Handbook 1 Cognitive Domain*; Longmans, Green and Co Ltd.: London, UK, 1956; pp. 18–24.

13. Biggs, J.B.; Collis, K.F. *Evaluating the Quality of Learning: The SOLO Taxonomy (Structure of the Observed Learning Outcome)*; Academic Press: New York, NY, USA, 1982; pp. 1–255.

14. Greller, W.; Drachsler, H. Translating learning into numbers: A generic framework for learning analytics. *J. Educ. Technol. Soc.* 2012, 15, 42–57.

15. Hui, B.; Farvolden, S. How Can Learning Analytics Improve a Course? In Proceedings of the 22nd Western Canadian Conference on Computing Education, Abbotsford, BC, Canada, 5–6 May 2017; ACM: New York, NY, USA, 2017.

16. Lajis, A.; Nasir, H.M.; Aziz, N.A. Proposed assessment framework based on bloom taxonomy cognitive competency: Introduction to programming. In Proceedings of the 2018 7th International Conference on Software and Computer Applications, Kuantan, Malaysia, 8–10 February 2018; ACM: New York, NY, USA, 2018.

17. Lagus, J.; Longi, K.; Klami, A.; Hellas, A. Transfer-Learning Methods in Programming Course Outcome Prediction. *ACM Trans. Comput. Educ.* 2018, 18, 1–18. [CrossRef]

18. Chung, C.Y.; Hsiao, I.H. 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, Portland, OR, USA, 11–14 March 2020.

19. Carter, A.S.; Hundhausen, C.D.; Adesope, O. 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, Omaha, NE, USA, 9–13 August 2015.

20. Ashenafi, M.M.; Ronchetti, M.; Riccardi, G. Predicting Student Progress from Peer-Assessment Data. *Int. Educ. Data Min. Soc.* 2016, 1, 270–275.

21. Ninruitsirikun, U.; Imai, H.; Watanapa, B.; Arpnikanondt, C. Principal Component Clustered Factors for Determining Study Performance in Computer Programming Class. *Wirel. Pers. Commun.* 2020, 110, 1–20. [CrossRef]

22. Yeomans, L.; Zschaler, S.; Coate, K. Transformative and Troublesome? Students’ and Professional Programmers’ Perspectives on Basic Concepts in Programming. *ACM Trans. Comput. Educ.* 2019, 19, 1–27. [CrossRef]

23. Gomes, A.; Correia, F.B. Bloom’s Taxonomy Based Approach to Learn Basic Programming Loops. In Proceedings of the 2018 IEEE Frontiers in Education Conference (FIE), San Jose, CA, USA, 3–6 October 2018; IEEE: Piscataway, NJ, USA, 2018.

24. Xinogalos, S. Object-oriented design and programming: An investigation of novices’ conceptions on objects and classes. *ACM Trans. Comput. Educ.* 2015, 15, 1–21. [CrossRef]

25. Esteero, R.; Khan, M.; Mohamed, M.; Zhang, L.Y.; Zingaro, D. Recursion or Iteration: Does it Matter What Students Choose? In Proceedings of the 49th ACM Technical Symposium on Computer Science Education, Baltimore, MD, USA, 21–24 February 2018.

26. Ullah, Z.; Lajis, A.; Jamjoom, M.; Altalhi, A.H.; Shah, J.; Saleem, F. A rule-based method for cognitive competency assessment in computer programming using Bloom’s taxonomy. *IEEE Access* 2019, 7, 64663–64675. [CrossRef]

27. Dorodchi, M.; Dehbozorgi, N.; Frevert, T.K. “I wish I could rank my exam’s challenge level!”: An algorithm of Bloom’s taxonomy in teaching CS1. In Proceedings of the 2017 IEEE Frontiers in Education Conference (FIE), Indianapolis, IN, USA, 18–21 October 2017; IEEE: Piscataway, NJ, USA, 2017.
28. Dos Santos, V.; De Souza, É.F.; Felizardo, K.R.; Vijaykumar, N.L. Analyzing the Use of Concept Maps in Computer Science: A Systematic Mapping Study. *Inform. Educ.* 2017, 16, 257–288. [CrossRef]

29. Wang, S.; Liu, L. Prerequisite concept maps extraction for automatic assessment. In Proceedings of the 25th International Conference Companion on World Wide Web, Montreal, QC, Canada, 11–15 April 2016; International World Wide Web Conferences Steering Committee: Geneva, Switzerland, 2016.

30. Larraza-Mendiluze, E.; Garay-Vitoria, N. A comparison between lecturers’ and students’ concept maps related to the input/output topic in computer architecture. In Proceedings of the 12th Koli Calling International Conference on Computing Education Research, Koli, Finland, 15–18 November 2012.

31. Li, L.; Mao, H.; Xu, L. Application of concept maps-based anchored instruction in programming course. In Proceedings of the 2010 10th IEEE International Conference on Computer and Information Technology, Bradford, West Yorkshire, UK, 29 June–1 July 2010; IEEE: Piscataway, NJ, USA, 2010.

32. Aşıksoy, G. Computer-based concept mapping as a method for enhancing the effectiveness of concept learning in technology-enhanced learning. *Sustainability* 2019, 11, 1005. [CrossRef]

33. Keppens, J.; Hay, D. Concept map assessment for teaching computer programming. *Comput. Sci. Educ.* 2008, 18, 31–42. [CrossRef]

34. Allinjawi, A.A.; Al-Nuaim, H.A.; Krause, P. An Achievement Degree Analysis Approach to Identifying Learning Problems in Object-Oriented Programming. *ACM Trans. Comput. Educ.* 2014, 14, 1–15. [CrossRef]

35. Anohina-Naumeca, A.; Strautmane, M.; Grundspenkis, J. Development of the scoring mechanism for the concept map based intelligent knowledge assessment system. In Proceedings of the 11th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing on International Conference on Computer Systems and Technologies, Sofia, Bulgaria, 17–18 June 2010.

36. Tulasi, B. Significance of Big Data and analytics in higher education. *Int. J. Comput. Appl.* 2013, 68, 21–23. [CrossRef]

37. Siemens, G.; Baker, R.S.D. Learning analytics and educational data mining: Towards communication and collaboration. In Proceedings of the 2nd international conference on learning analytics and knowledge, Vancouver, BC, Canada, 29 April–2 May 2012.

38. Michael, O. *Emerging Perspectives on Learning, Teaching, and Technology*; CreateSpace North Charleston: North Charleston, SC, USA, 2010; pp. 47–56.

39. Designing Effective Multiple Choice Questions. Available online: https://www.mcgill.ca/skillsets/files/skillsets/mcq_handout3.pdf (accessed on 29 July 2020).

40. Abreu, P.H.; Silva, D.C.; Gomes, A. Multiple-Choice Questions in Programming Courses: Can We Use Them and Are Students Motivated by Them? *ACM Trans. Comput. Educ.* 2018, 19, 1–16. [CrossRef]

41. Stice, C.F.; Alvarez, M.C. Hierarchical concept mapping in the early grades. *Child. Educ.* 1987, 64, 86–96. [CrossRef]

42. Cronbach, L.J. Coefficient alpha and the internal structure of tests. *Psychometrika* 1951, 16, 297–334. [CrossRef]

43. Garson, G.D. *Testing Statistical Assumptions*; Statistical Associates Publishing: Asheboro, NC, USA, 2012; pp. 15–17.

44. Ahadi, A.; Hellas, A.; Lister, R. A contingency table derived method for analyzing course data. *ACM Trans. Comput. Educ.* 2017, 17, 1–19. [CrossRef]

45. Mora, H.; Pujol-López, F.A.; Mendoza-Tello, J.C.; Morales-Morales, M. An education-based approach for enabling the sustainable development gear. *Comput. Hum. Behav.* 2020, 107, 105775. [CrossRef]

46. McMahon, B.M.; Sembiante, S.F. Re-envisioning the purpose of early warning systems: Shifting the mindset from student identification to meaningful prediction and intervention. *Rev. Educ.* 2020, 8, 266–301. [CrossRef]

47. Castro-Wunsch, K.; Ahadi, A.; Petersen, A. 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, Seattle, WA, USA, 8–11 March 2017.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).