Education 4.0 in higher education and computer science: A systematic review

Journal Item

Link(s) to article on publisher's website:
http://dx.doi.org/doi:10.1002/cae.22643

Received: 2 December 2022 | Revised: 24 February 2023 | Accepted: 5 May 2023

DOI: 10.1002/cae.22643

RESEARCH ARTICLE

WILEY

Education 4.0 in higher education and computer science: A systematic review

Bart Rienties[1] | Rebecca Ferguson[1] | Dalibor Gonda[2] | Goran Hajdin[3] | Christothea Herodotou[1] | Francisco Iniesto[1] | Ariadna Llorens Garcia[4] | Henry Muccini[5] | Julia Sargent[1] | Sirje Virkus[6] | Maria Vittoria Isidori[5]

[1]Institute of Educational Technology, The Open University, Milton Keynes, UK

[2]Zilinska univerzita v Ziline, Zilina, Slovakia

[3]Faculty of Organization and Informatics, University of Zagreb, Zagreb, Croatia

[4]Department of Business Administration, Vilanova i la Geltrú, Barcelona, Spain

[5]Computer Science and Operations Research, Department of Information Engineering Computer Science and Mathematics, University of L'Aquila, L'Aquila, Abruzzo, Italy

[6]School of Digital Technologies, Tallinn University of Technology, Tallinn, Harjumaa, Estonia

Correspondence
Bart Rienties, Institute of Educational Technology, The Open University, Milton Keynes MK76AA, UK.
Email: Bart.rienties@open.ac.uk

Funding information
European Commission, Grant/Award Number: "Accelerating the transition towards Edu 4.0 in HEIs"

Abstract

Education 4.0 is a recently introduced concept focused on innovation, novelty, use of technology and connections with employment and industry. In particular, in engineering disciplines like computer science (CS) it is essential that educators keep up to date with industry developments. Indeed, how CS educators effectively design and implement innovative teaching and learning deserves more systematic attention. This study aims to catalogue and synthesise learning design approaches to teaching and learning within CS: (1) Which innovative pedagogic approaches are used in teaching of CS? (2) Which approaches align with Education 4.0? (3) What skills and competences do educators require to align CS teaching with Education 4.0? Our systematic literature review (SLR) included CS papers published between 2016 and 2020. Two hundred and thirty-one studies were identified of which 66 were included in the final phase, which were coded by a multidisciplinary team. The findings indicated that many CS educators included Education 4.0 learning design elements. We found a clear distinctive three-cluster solution: (1) EDU4 light, (2) project-based/hands-on learning and (3) full EDU4 (refer to Reference [7] conceptualisation, while Education 4.0 refers to our own definition [Reference 71]. These findings suggest three broad flavours when designing innovative CS practices, which might help educators align their practice.

KEYWORDS

computer science, education 4.0, learning design, teaching competencies

1 | INTRODUCTION

The way educators design blended and online courses, in short learning design, has a fundamental impact on how learners engage with learning activities [53, 90]. For example, in a systematic literature review (SLR) of 43 learning design studies, Mangaroska and Giannakos [53] substantial growth in research and application on learning design in higher education has been noted. Indeed, two recent reviews on learning design and learning analytics [51, 90] have indicated that substantial progress has been made on how learning design decisions by educators impact learners in the last 10 years. Nonetheless, there remains a strong need to 'explore how educators plan, implement, and evaluate learning designs' [51], and how students react and engage with these learning designs.

In particular, there is a paucity of research within the broad engineering discipline of computer science (CS) on how educators design for innovative practice [35, 63, 76]. Due to the strategic importance of CS in engineering and supporting the current and next generation of students to develop appropriate computing and data skills [30, 35, 43, 63], we specifically want to explore how CS educators are producing and implementing innovative learning designs.

This study aims to catalogue and synthesise learning design approaches to teaching and learning within CS that are aligned with Education 4.0. There are a wide range of definitions of education 4.0, but common elements include a focus on connections to employment and industry, innovation, novelty, technology use [7, 43, 46, 74, 93]. Given the contemporary conceptualisation of Education 4.0 and the rapid pace of development within CS, in this systematic literature review, we will review CS literature published in the period 2016–2020. Furthermore, as there is continuous change in technology and CS, we are specifically interested in whether (or not) innovative CS approaches refer to the skills CS educators need to be able to teach in an Education 4.0 manner. Therefore, in this study, we will aim to address what innovative pedagogical approaches have been used to support CS teaching (RQ1), which pedagogical approaches are consistent with Education 4.0 (RQ2) and finally what skills and competences do CS educators require to align their CS teaching with Education 4.0 (RQ3)?

2 | LITERATURE REVIEW

CS is a broad subject area that covers many disciplines and overlaps with many others. We use the definition provided by the UK Quality Assurance Agency Subject Benchmark Statement on Computing: 'CS provides the necessary knowledge to understand and build computational systems' [69]. The statement goes on to list the main characteristics of CS and notes that, '[g] enerally, these are expressed in the ability to specify, design and write computer programmes' The breadth of the field means it is able to draw on teaching methods from diverse disciplines and is also used as a way of preparing students for a wide range of professions.

2.1 | Previous systematic literature reviews on innovative approaches in CS

Recently several SLRs have been published on how CS educators design and implement innovative learning and teaching approaches, for example, [35, 63, 76, 86]. For example, Aničić, Divjak and Arbanas [63] conducted a meta-analysis of 155 papers from 1980 to 2014 of how CS educators align their curriculum to career development and developing employable graduates. The findings indicated that the curriculum designs and implications need to be aligned to the needs of the industry. As argued by Aničić, Divjak, and Arbanas [63] 'the literature indicates a need for innovative approaches in curriculum design and delivery, such as designing competency-based programmes that are not restrained by the traditional semester seat-time model, providing flexible curriculum and minimising the time spent in the classroom, or offering courses on not only how to manage innovation, but also on how to innovate'.

In a review of 157 learning designs implemented at The Open University, Toetenel and Rienties [86] found that the majority of educators primarily used two types of learning activities, namely, assimilative activities (e.g., reading, watching videos and listening to audio) and assessment activities. Often educators combined assimilative, productive (e.g., coding) and assessment activities or, alternatively, assimilative, finding and handling information and communication tasks (e.g., working together with peers). In a more recent study by Garousi et al. [35] 34 papers were analysed how software engineering education was aligned with industrial needs. Their findings indicated that to encourage the development of soft skills, educators need to use real-life projects, implement industry-academia collaboration in the design of education and anticipate future trends. Furthermore, educators need to prepare students to deal with those trends [35].

In this journal, Kocdar, Bozkurt and Goru Dogan [46] recently reviewed how engineering through distance education could potentially support effective learning and Education 4.0 in particular. Using an innovative approach of text mining and social network analysis, the 120 studies identified indicated that primarily distance education in engineering was provided via technology-enhanced engineering education, e-learning and m-learning, as well as virtual and remote labs [46]. While these studies provide important insights into how engineering and CS educators in particular implement learning and teaching, none of these studies specifically focused on, mentioned or included Education 4.0 concepts. Furthermore, none of these reviews specifically looked at the way the respective learning designs were used.

2.2 | Education 4.0

This SLR catalogues approaches to teaching and learning within CS that are aligned with Education 4.0. This is a relatively new term—Harkins originally proposed it in 2008 to describe innovation-producing education [40] as opposed to knowledge-producing education. Education 4.0 is related to the view that the current Industry 4.0 is becoming more and more automated, using modern smart technology and the Internet of Things (IoT). Recently, the World Economic Forum [93] positioned a broad range of eight skills to refer to Education 4.0, including global citizenship skills, innovation and creativity, and life-long learning. Other researchers provide more detailed pedagogical descriptions of what kind of learning activities could be present in Education 4.0, including Fisk [33] and later on Hussin [7], leading to nine characteristics associated with EDU 4.0, as illustrated in Figure 1.

The elements identified are all potentially innovative, and currently the Education 4.0 conceptualisation of Hussin [7] is the most cited Education 4.0 paper in Google Scholar. However, Hussin [7]'s focus on students rather than on the broader picture of how innovations are developed and embedded in terms of learning design is a potential caveat. In this study, we propose an alternative definition of Education 4.0 that draws on ideas and descriptions in a range of literature [11, 25, 43, 68, 74, 83, 89, 93]. Building on our initial explorative work [71] how European CS teachers built and designed innovative practice, we defined Education 4.0 as an 'approach to learning and teaching that emphasises the development of skills and competences necessary in a modern workplace using up-to-date technology. The skills and competences developed may relate directly to the technology, or they may be the softer skills (such as team-working and creativity) that are needed to work effectively in such an environment. The approach involves the use of technology and/or pedagogy that is innovative in the context, and therefore requires flexible and creative approaches to its implementation'. Note that in the remainder of this study, when we use EDU4 we refer to the Hussin [7] conceptualisation, while Education 4.0 refers to our own definition [71].

As evidenced by a range of studies [35, 46], being able to design and implement innovative pedagogical approaches requires substantial new and/or updated skills and competences from educators to make use of Education 4.0 approaches. A recurring theme seems to be a shift from CS educators as being a knowledge transmitter to an educator as a facilitator or moderator or consultant of learning [71]. Educators could achieve that by being flexible (adapt to change) [71], supportive, help students to develop ownership of learning, foster an environment where students take risks and share what they do not know about, and where failure is acceptable. This role was often discussed within a flipped classroom implementation [28] that could give control to students to study the teaching material at their own pace and contact the teacher to solve problems and discuss their learning. In such conditions, the teacher is monitoring a student's progress and facilitates understanding through discussions [54]. An increasing number of CS educators have started to implement project-based learning and hands-on experiences in their classroom [4]. However, the specific skills and competencies needed to design, implement and evaluate effective Education 4.0 CS courses has received limited attention.

In our initial work [71] exploring how 20 European CS studies in nine European countries designed and implemented innovative teaching and learning, we found some preliminary evidence that educators indeed design CS practices by incorporating some of the nine EDU4 elements. On average we found 4.10 (out of nine) EDU4 elements in these 20 studies but with substantial variation ($SD = 2.10$) and range (2–9). A preliminary cluster analysis suggested three initial clusters of practices among European CS educators, which we initially labelled as EDU 4.0 light (i.e., incorporating one or two EDU 4.0 characteristics, $n = 8$), Project-based/hands-on learning (i.e., using project-based/hands-on approaches, $n = 6$), and Full EDU 4.0, using a broad range of EDU4 characteristics, $n = 6$). Given the relatively small size of this European data set it was difficult to generalise the findings across the globe, and

1. **Learning any time / anywhere:** Students will be able to learn where and when they choose.
2. **Personalised learning:** Study tools will adapt to the capabilities of the student.
3. **Choice of how to learn:** Students will be able to modify their learning process.
4. **Project-based learning:** Students will learn to apply their skills in a variety of situations.
5. **Hands-on learning:** Students will have authentic experiences and gain real-world skills.
6. **Data interpretation:** Students will learn to interpret and reason with data.
7. **Assessed differently:** Knowledge and skills will be assessed in new ways.
8. **Student ownership of curriculum:** Students will have critical input into their courses.
9. **More independent:** students will become more independent.

FIGURE 1 Education 4.0 characteristics according to Hussin [7].

furthermore it did not allow us to conduct more robust statistical analysis.

Therefore, in this follow-up study, we extend our search beyond Europe to synthesise how CS educators design and implement innovative CS practice (RQ1) in countries across the globe, and whether (or not) these align with Education 4.0 (RQ2). Furthermore, by enlarging our sample from 20 to 66 papers it will allow us to conduct a more in-depth synthesis to explore whether these patterns are common to CS, or specific to Europe. Finally, a specific new element previously unexplored is that we would be keen to identify what skills and competencies might be needed for educators to align their practices with Education 4.0 (RQ3). This leads us to the following research questions:

RQ1: What innovative pedagogical approaches have been used to support CS teaching?

RQ2: Which pedagogical approaches are consistent with Education 4.0?

RQ3: What skills and competences do CS educators require to align their CS teaching with Education 4.0?

3 | METHODS

In this SRL, we follow recommendations from [46, 53, 55, 70] and build on our initial approach [71]. In total, four research databases were analysed based upon their ranking and coverage of CS research: Science Direct, Scopus, Web of Science and Wiley InterScience. Papers had to be published in English during the 5-year period 2016–2020, thereby increasing the chance that a particular study used a contemporary and innovative pedagogical approach in CS. We used the following search string: 'computer science' AND education AND teaching AND pedagogy AND ('undergraduate' OR 'postgraduate'). Two hundred and thirty-one unique publications were identified across the four databases.

3.1 | Coding process

We followed the same coding procedure as in Rienties et al. [71], whereby an interdisciplinary team of academics from CS and educational technology from six European countries analysed the data. Initially, all abstracts were manually screened, and 75 studies were excluded as the focus was on elementary/secondary education, subjects other than CS and/or learners instead of teaching. In the first phase, after 1 h of online training and discussion on the coding scheme, 18 members of the TEACH4EDU project read 156 studies in detail. Members coded on average eight studies (range: 3–11), of which 68 studies were included in the subsequent analysis following three inclusion criteria: (1) Is it an 'innovative' application in a CS course? (2) Does it use technology or pedagogy in an innovative way? (3) Is the innovation evaluated, if so how? As notions of innovation and technology can differ, in the second phase we specifically assigned selected papers to coders from different disciplines and geographical contexts to check whether the proposed approach was indeed within these inclusion criteria.

In the second phase, 17 members of the TEACH4EDU project participated in another 1-h online follow-up training and discussion of their 20-variable online coding scheme. A new set of studies was randomly assigned to coders to compare to their initial coding in the first phase, thereby ensuring that at least two coders checked and independently coded each 'innovative' pedagogy in CS. While text mining and bibliometric techniques might be useful to identify common trends in literature, given the complexity of how scholars might write about learning design and Education 4.0, in this study we argue that human verification and validation of innovation are required. Thus for RQ2, we adopted the nine key EDU4 characteristics of Hussin [7] and our Education 4.0 definition for human coding of each included paper. For the analysis, we used both the individual EDU4 scores as well as the aggregate score (e.g., Schäfer [56] was coded as EDU4 1 Learning any time and 9 More independent, leading to an aggregate EDU4 score of 2).

For RQ3, coders indicated whether (or not) any specific skills required by educators to support the teaching of CS to students were mentioned. If yes, coders could use a follow-up open text box to add any description and conceptualisation of educator skills. Following this, we recoded and aggregated the skills.

Based on the coding scheme developed from the RQs, an average of four studies (range: 2–10) were coded per coder. A sample from 15 studies was double-coded to show reliable coding (mean Cohen Kappa EDU4 = 0.84). The coders of Phase 1 then checked the codes of the coders in Phase 2, discussed any differences, and agreed on the final coding (average Cohen Kappa EDU4 = 0.93). If a study did not indicate any EDU4 characteristic, we removed it from further analysis, and therefore we ended up with a total of 66 studies, as illustrated in Figure 2.

3.2 | Data analyses

The vast majority of studies included referred to undergraduate CS students (79%), followed by a mix of undergraduate and postgraduate students. Five studies did not explicitly mention the specific student population,

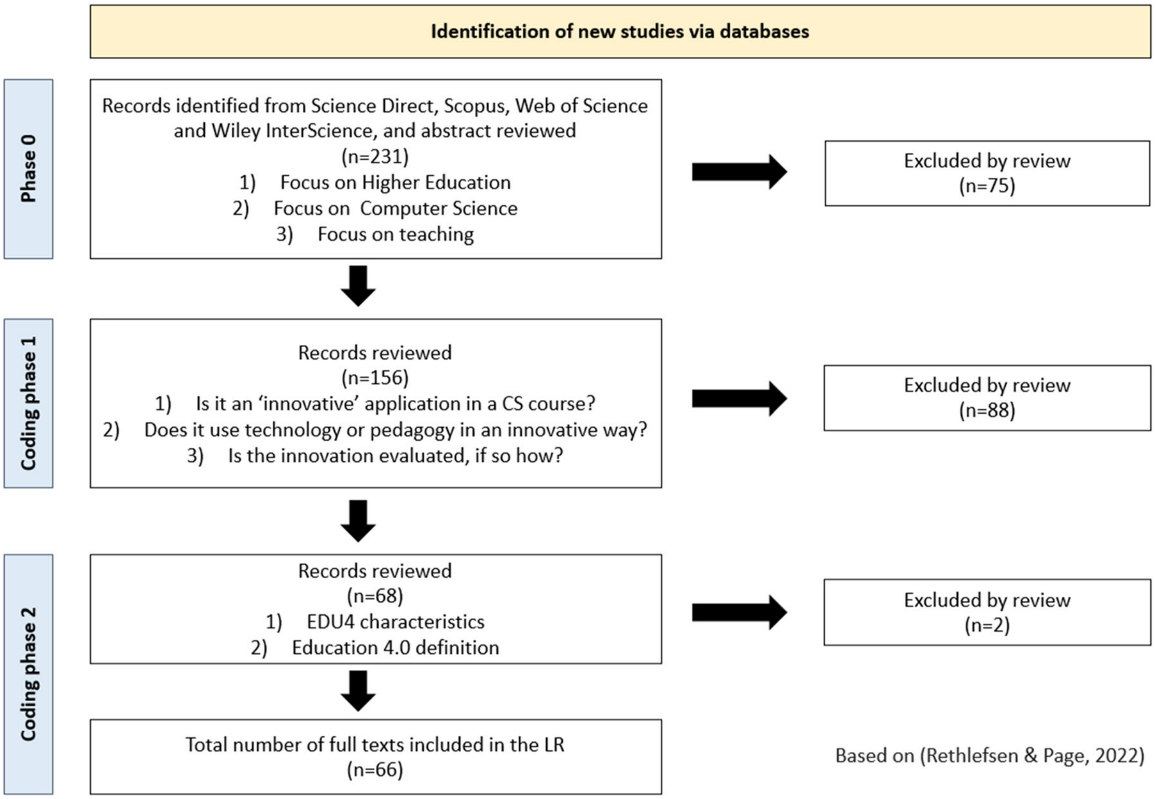


FIGURE 2 Visualisation of selection of included/excluded studies.

and one included teachers only. A total of 36% of studies were from the United States, followed by Spain (9%), Brazil (8%) and Germany (6%). Using the GLOBE geo-cultural regions classification [42], which has clustered countries across 10 broad geo-cultural regions, 47% of studies were conducted by Anglo-Saxon countries, followed by Latin American countries and Latin European (each 12%), Eastern European (8%), Scandinavian and German countries (each 6%), Confucian Asian (5%) and Middle Eastern countries (3%). No studies were identified from African or Southern Asian countries. Using analysis of variance (ANOVA) analyses, no significant differences were found on our key variables and GLOBE, indicating no substantial differences in practices in CS based upon national/geo-cultural regions.

In terms of reporting the findings of RQ1-2, we first explored the overall data, then carried out an exploratory factor analysis (principal component analysis) with direct oblimin rotation to identify a common structure in the EDU characteristics. Mundfrom et al. [56] suggest that a sample size of respondents 40–60 would be appropriate when there is a good level of agreement among items. Perhaps slightly different from normal surveys where participants independently from each other complete a survey, in our study each of the 66 papers were (double) coded by experienced academics/coders, so therefore we argue that an even smaller sample could lead to a reliable factor structure. Multiple factor structures were explored, but a two-factor structure had the best fit. Finally, a k-means cluster analysis was conducted to explore any common patterns in terms of learning designs employed by CS educators. We explored a range of clusters (from 1 to 5), and subsequently analyzed the explained variance using ANOVAs. When more than three clusters were included some of the clusters became too small to be meaningful, while with less than three the explained variance was smaller than those with three clusters. For RQ3, all articles were screened whether (or not) reference was made towards educators' competences and skills to implement a respective innovation. If a study explicitly mentioned this, it was coded and included in an open text box. These open-text boxes were later analyzed by authors C. H. and J. S. to find common patterns.

4 | RESULTS

In terms of RQ1, 66 studies included at least one EDU4 characteristic [7]. Furthermore, in total 54 articles (80%) were considered to fit our own Education 4.0 definition. However, perhaps surprisingly none of the articles explicitly mentioned the terms Education 4.0 or EDU4. As indicated in Figure 3, on average the 66 studies included 4.41 of the nine EDU4 characteristics of Hussin [7], with a substantial variation (SD = 2.30). There seemed to be two peaks in Figure 3, whereby 35% of studies only had two to three EDU4 characteristics, with another peak at seven EDU4 characteristics.

As indicated in Figure 4, the most common EDU4 characteristic was 5) hands-on learning (73%), followed by 9) more independent (67%), 4) project-based learning (61%). Around half of the studies included the characteristic that 1) learning any time/anywhere, while around a third of studies included 7) assessed differently (35%) and 8) student ownership of curriculum (32%). Furthermore, as illustrated by the error bars, there was substantial variation in the 66 CS practices. These findings confirm initial findings reported in Rienties et al. [71].

In terms of RQ2, with our larger data set we were able to explore how the nine EDU4 elements were related to each other. We found a moderately strong correlation (rho = 0.429, $p < .01$) between the aggregate Hussin [7] and our Education 4.0 definition, with the strongest correlation on the EDU4 characteristic 5 (rho = 0.417, $p < .01$). The individual EDU4 characteristics were not all directly and significantly correlated. Therefore, an explorative factor analysis was conducted on the data collected, which indicated the existence of two factors with item loads of 0.45 and more. The first component had an eigenvalue of 2.62 (corresponding to 29% of the explained variance), the second component had an eigenvalue of 1.45 (corresponding to 16% of the explained variance).

As indicated in Table 1, EDU4 characteristic 2, 1, 6, 9 and 3 loaded on the first factor, which we will label as 'individual choice and development'. EDU4 characteristic 4, 5 and 7 loaded on the second factor, which we will label as 'intention project-based/hands-on learning'. EDU4 characteristic 8 did not load on any factor. The respective Cronbach Alphas for these two factors were .68 and .62. In other words, educators often combine EDU4 characteristics together when designing and implementing CS courses based upon these two factors.

Follow-up analysis using the k-means cluster method using the 66 instead of 20 studies confirmed the three cluster model initially found in Rienties et al. [71]. As shown in Figure 5 in line with our previous smaller sample of European CS practices, three clusters of studies were identified, which we (re)label as (1) *EDU4 light* ($n = 18$), (2) *project-based/hands-on learning* ($n = 22$) and (3) *full EDU4* ($n = 26$). With the notable exception of EDU4 characteristic 8, using ANOVAs all EDU4 characteristics were significantly different between the three clusters with large effect sizes, indicating clear unique clusters ($p < 01$, η^2 for EDU4 characteristics

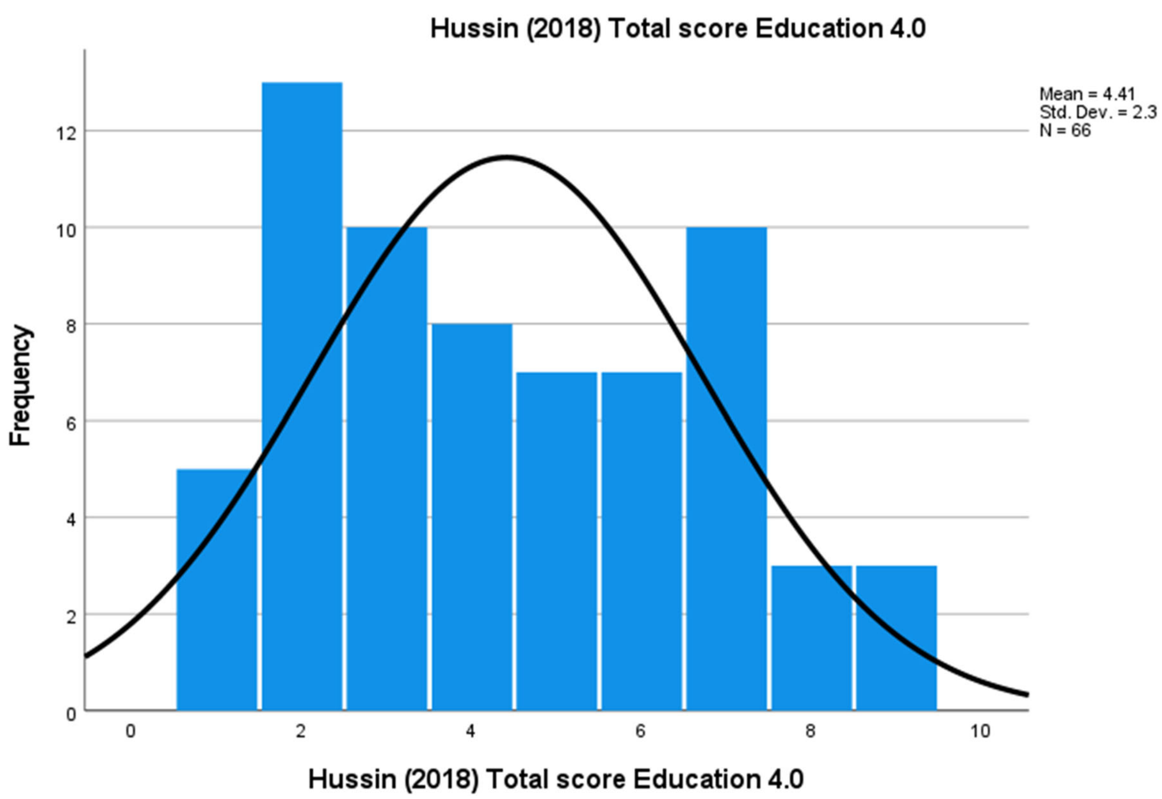


FIGURE 3 Histogram of EDU4 concepts identified in the 66 papers (range: 1–9).

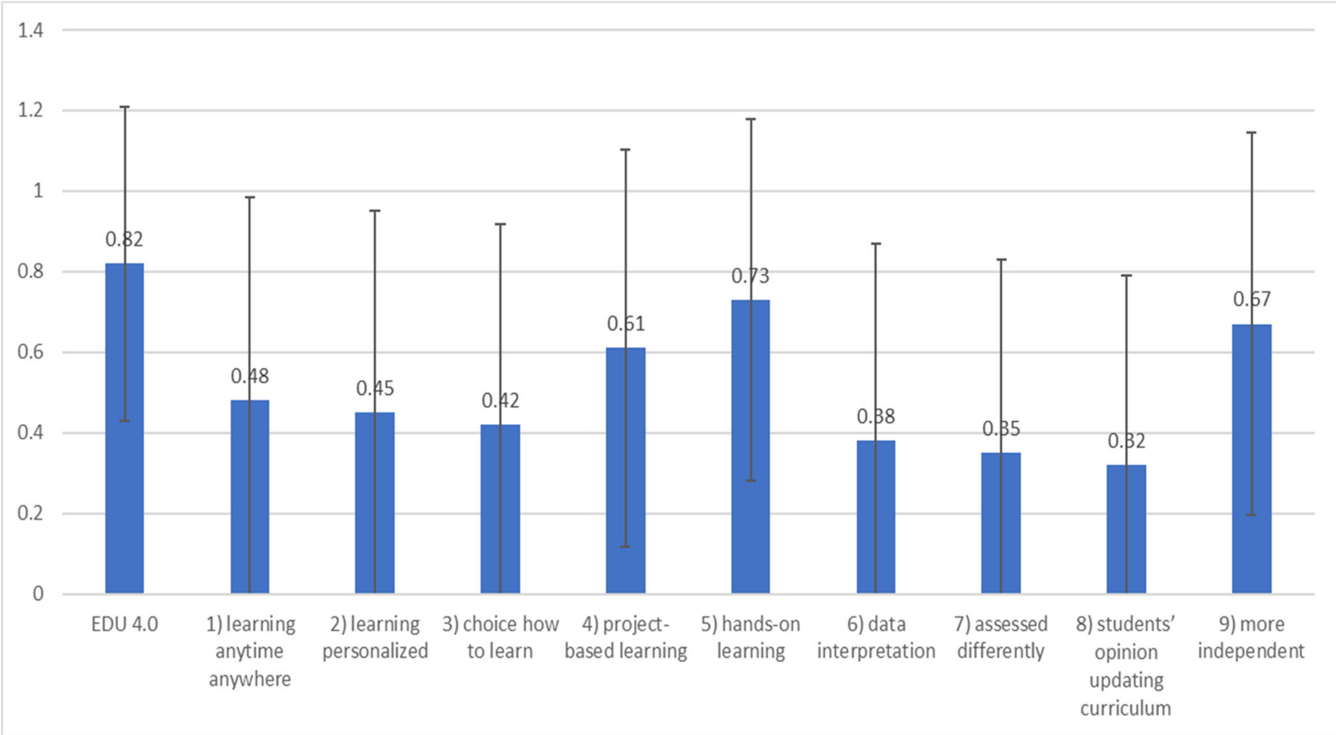


FIGURE 4 EDU4 characteristics present within the 66 papers (with error bars, range: 0 = not included to 1 = included). A total of 73% of papers included 5) hands-on learning, while 32% of papers included 8) students' opinion in updating curriculum.

TABLE 1 Pattern structure of Factor analysis EDU4 Characteristics (order based upon factor loadings).

	1	2
2) learning will be personalised to individual students	0.769	
1) learning can take place anytime anywhere	0.644	
6) students will be exposed to data interpretation in which they are required to apply their theoretical knowledge to numbers and use their reasoning skills to make inferences based on logic and trends from given sets of data	0.618	
9) students will become more independent in their own learning	0.617	
3) students have a choice in determining how they want to learn	0.589	
8) students' opinion will be considered in designing and updating the curriculum		
4) students will be exposed to more project-based learning		0.883
5) students will be exposed to more hands-on learning through field experience (e.g., internships, mentoring projects, collaborative projects)		0.836
7) students will be assessed differently and the conventional platforms to assess students may become irrelevant or insufficient		0.455
Extraction Method: Principal Component Analysis.		
Rotation Method: Oblimin with Kaiser Normalisation.		

ranged from 0.204–0.611). In other words, there appeared to be three distinct innovative pedagogical practices present in published work on CS across the globe in the last 5 years.

As indicated in Figure 6, EDU4 light studies (blue circles) mostly had relatively low total EDU4 scores, and often did not include project-based activities. Therefore, most of these studies in Figure 6 were positioned on the

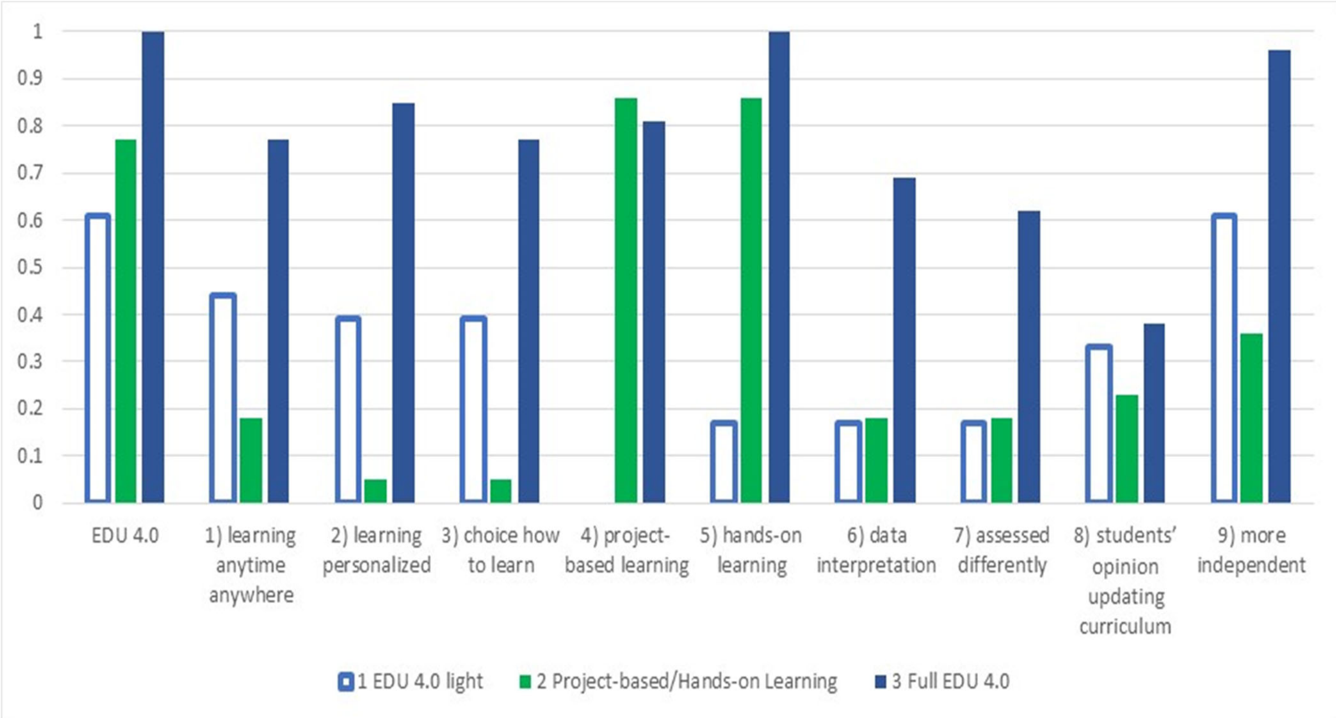


FIGURE 5 Proportion of articles that included the given strategy per cluster (0 = not included, 1 = included).

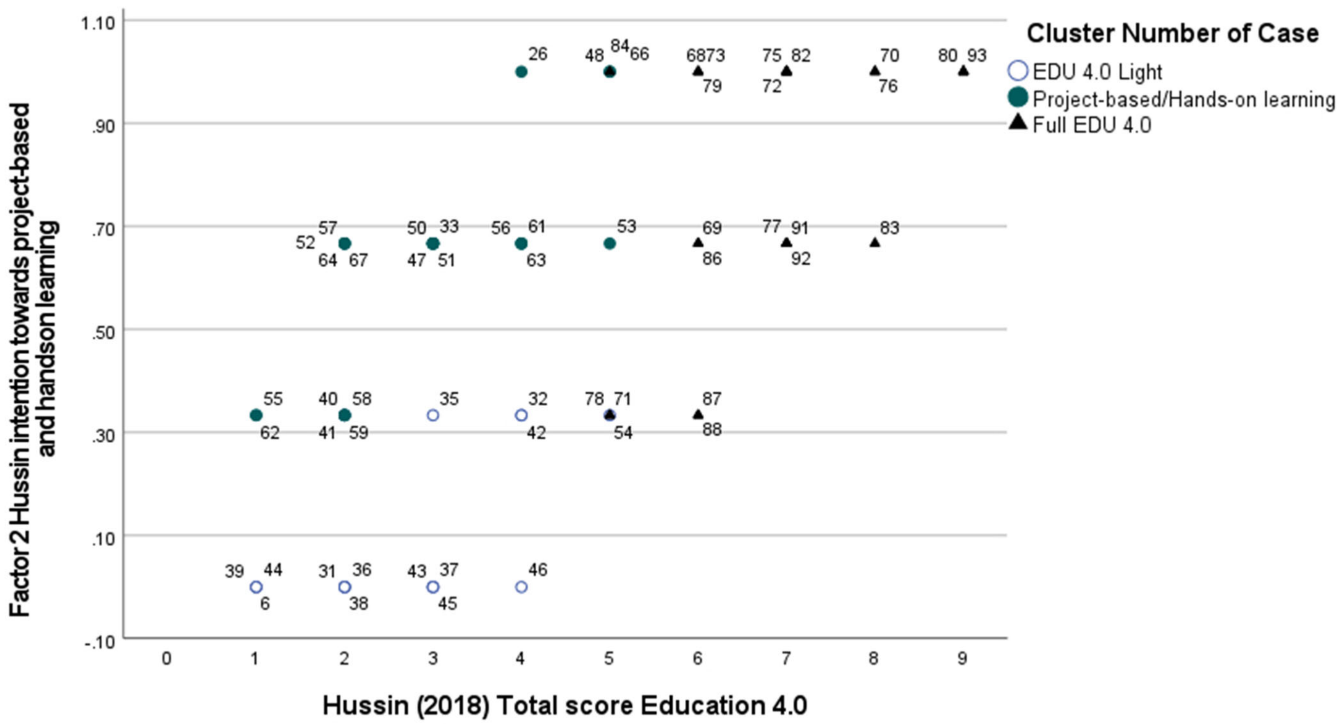


FIGURE 6 Scatterplot of cluster analysis results of 66 studies (EDU 4.0 vs. construct project-based/hands on learning). Note that the numbers in Figure 6 refer to the respective studies. EDU 4.0 score refers to the total number of EDU 4.0 characteristics identified by the coders in each study, while the construct project-based/hands on learning refers to the sum of EDU 4 characteristic 4, 5 and 7.

bottom left. In contrast, while some project-based/hands-on learning studies (green circles) also had relatively low EDU4 scores, in particular for personalised learning and choice how to learn, they had a strong focus on project-based and hands-on learning. Therefore, many of these studies are positioned in the middle to top-left quadrant of Figure 6. Finally, studies that were classified as full EDU4 studies (blue triangles) were mostly positioned in the middle and right of Figure 6 indicating these studies used more and even all EDU4 characteristics in their designs. Note that the numbers in Figure 6 refer to the studies discussed below.

4.1 | EDU4 light

As indicated in Table 2, in EDU4 light studies educators mostly focused on more independent (61%), learning any time/anywhere (44%), personalised learning (39%), and choice how to learn (39%), but with limited hands-on learning (17%) and no project-based learning (0%). For example, Schäfer [77] introduced the concept of a modern C++ course for students of electrical engineering and CS based on an inverted classroom with attractive IoT hardware. The main goal of the new course was to reduce lecture time in favour of practical learning of students through programming.

Burrows and Borowczak [17] explored one group's use of action research and lesson study in three US university-level CS courses to improve undergraduate Engineering student soft skill understandings. The group collaborated to enhance participant engagement by utilising one lesson focused on soft skills over three research lesson iterations in three distinct university semesters. In another study, Park and Kim [60] identified that while assignments on kernel programming are essential parts of operating system courses taught to CS students to provide them a deep understanding of real-world operating systems, often students struggle. In particular, students were routinely flustered by the daunting task of building a practice environment from scratch, and instructors were pressed for time while validating student work that required several kernel installations and reboots. Therefore, Park and Kim [60] provided a cloud-based system that facilitated a high level of accessibility, enabling students to work on their assignments anywhere they want. While each of these studies listed in Table 2 indicated innovative pedagogical enhancements, most of these studies focussed only on a limited number of EDU4 characteristics.

TABLE 2 EDU4 Light studies.

Authors	E1	E2	E3	E4	E5	E6	E7	E8	E9	Country
Apiola, Lokkila and Laakso [6]	Y		Y		Y				Y	Finland
Burrows and Borowczak [17]	Y	Y		Y				Y	Y	USA
Degener, Haak, Gold-Veerkamp and Abke [26]	Y								Y	Germany
Dickson, Dragon and Lee [27]		Y	Y		Y					USA
Dondio and Shaheen [29]								Y	Y	Ireland
Fisher, Rader and Camp [32]	Y	Y						Y		USA
Frevert et al. [34]								Y	Y	USA
Giacaman and De Ruvo [37]			Y							New Zealand
Hosseini, Hartt and Mostafapour [41]							Y		Y	USA, Wales, Canada
Parejo, et al. [59]	Y						Y			Spain
Park and Kim [60]	Y	Y					Y		Y	Korea
Pilkington [66]	Y					Y			Y	South Africa
Scatalon, Garcia and Barbosa [76]		Y								Brazil
Schäfer [77]	Y								Y	Germany
Shi, Min and Zhang [80]			Y							China
Silva, Steinmacher and Conte [82]			Y					Y	Y	Brazil
Tyler and Abdrakhmanova [88]		Y	Y			Y		Y		Kazakhstan

Note: E1–E9 refer to EDU4 characteristics.

4.2 | Project-based/hands-on learning

The second group, which we called project-based learning/hands-on learning, placed a strong emphasis on project-based learning (86%) and hands-on learning (86%), with relatively limited focus on choice how to learn (5%), personalised learning (5%), and learning any time/anywhere (18%), as illustrated in Table 3. For example, Caceffo, Gama and Azevedo [19] assessed the benefits of the use of technology and active learning practices (i.e., Project-Based Learning and Peer Instruction) in the classroom with 25 students to contribute to a more effective and efficient learning environment. Alomari, Ramasamy, Kiper and Potvin [4] described the use of an innovative platform to improve the knowledge of 51 CS students about software testing by providing a set of learning objects and tutorials categorised by difficulty level. This evolved into a collaborative learning environment that included social networking features such as the ability to award virtual points for student social interaction about testing [4].

Another study [19] assessed the benefits of the use of active learning practices teaching algorithms, data structures and programming logic in a CS introductory course with the feedback from two instructors and 24 undergraduate students via interviews and surveys to contribute to a more effective and efficient learning environment. The study indicated that both students and instructors enjoyed the use of new technologies and active learning in the course, but they would like to prioritise two-way communication between students and instructor, and collaboration among students during class.

In another study, Carrascal, del Barrio and Botella [20] proposed an approach for teaching quantum computing which included using classical object-oriented programming to programme a basic quantum simulator. The approach emphasised practical training and was aimed at students who do not necessarily have a deep theoretical grounding in

TABLE 3 Project-based/hands-on learning studies.

Authors	E1	E2	E3	E4	E5	E6	E7	E8	E9	Country
Aghaee and Keller [1]	Y			Y	Y		Y		Y	Sweden
Alasbali and Benatallah [2]				Y					Y	Global
Alegre, et al. [3]			Y	Y	Y					USA
Alomari, Ramasamy, Kiper and Potvin [4]	Y			Y	Y		Y			USA
Berikan and Özdemir [9]				Y	Y	Y				Turkey
Bielefeldt, et al. [10]				Y	Y					USA
Borowczak and Burrows [13]				Y	Y	Y		Y	Y	USA
Burrows and Borowczak [16]					Y		Y	Y		USA
Bushmeleva and Baklashova [18]					Y					Russia
Caceffo, Gama and Azevedo [19]				Y	Y				Y	Brazil
Carrascal, del Barrio and Botella [20]				Y	Y	Y		Y		Spain
Casañ, Alier and Llorens [21]				Y	Y					Spain
Chamberlin et al. [22]	Y				Y					USA
Cobos and Roger [24]				Y				Y		Spain
Fagerholm et al. [31]				Y	Y				Y	Finland
Juárez, Aldeco-Pérez and Velázquez [44]		Y		Y	Y				Y	Mexico
Lewis and Lacher [48]				Y						USA
Liang and Chapa-Martell [49]	Y			Y	Y				Y	Japan
Llorens, Berbegal-Mirabent and Llinas-Audet [50]				Y	Y					Spain
Mäkiö et al. [52]				Y	Y				Y	UK
Santos et al. [75]				Y	Y	Y	Y	Y		Austria, Czech Republic, Slovak Republic, UK
Seyam and McCrickard [78]				Y	Y					USA

Mathematics. After a mandatory introduction that covered the basics, the course made use of problem-based learning. As illustrated in these studies as well as in Table 3, there was an extensive focus on project-based and hands-on learning. However, due to the nature of project-based learning in almost all these studies, flexibility in anytime, anywhere, personalisation and learning choices was relatively low.

4.3 | Full EDU4

The full EDU4 cluster was strongly focused on hands-on learning (100%), more independent (96%), personalised learning (85%), learning any time/anywhere (77%) and choice how to learn (77%). The lowest EDU4 characteristic was student ownership of curriculum (38%), as illustrated in Table 4, although this was substantially higher than the other two clusters.

For example, Pivkina [67] described the experience of using an undergraduate student as a peer learning assistant (PLA), supporting 80 students in three different undergraduate CS courses with 20 students. PLAs held office hours, helped with labs and tutorials and facilitated student group work in class. They were therefore practicing tutoring their peer students. The impact of the experience was measured by comparing student

TABLE 4 Full EDU4 studies.

Authors	E1	E2	E3	E4	E5	E6	E7	E8	E9	Country
Alsaif, Li, Soh and Alraddady [5]	Y		Y	Y	Y		Y		Y	Saudi Arabia
Behnke, Kos and Bennett [8]	Y		Y	Y	Y	Y			Y	USA
Borge, Ong and Goggins [12]	Y	Y		Y	Y	Y	Y	Y	Y	USA
Broisin, Venant and Vidal [14]	Y	Y	Y		Y				Y	France
Buffardi and Valdivia [15]			Y	Y	Y	Y	Y	Y	Y	USA
Charlton and Avramides [23]		Y	Y	Y	Y		Y		Y	UK
Corritore and Love [47]	Y		Y	Y	Y		Y	Y	Y	USA
Gestwicki and McNely [36]		Y	Y	Y	Y		Y	Y	Y	USA
Gonçalves, von Wangenheim, Hauck and Zanella [38]	Y	Y	Y	Y	Y	Y	Y		Y	Brazil
Goumopoulos, Nicopolitidis, Gavalas and Kameas [39]	Y	Y	Y	Y	Y	Y			Y	Greece
Knobelsdorf, Frede, Böhne and Kreitz [45]	Y	Y			Y	Y			Y	Germany
Munkvold [57]		Y	Y	Y	Y		Y		Y	Norway
Paschoal, Oliveira, Nakagawa and Souza [61]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Brazil
Pawelczak [62]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Germany
Peng [64]	Y	Y	Y	Y	Y	Y	Y			USA
Peteranetz, Flanigan, Shell and Soh [65]	Y	Y	Y	Y	Y	Y		Y	Y	USA
Pivkina [67]		Y		Y	Y		Y		Y	Mexico
Ruiz, Serral Asensio and Snoeck [72]	Y	Y		Y	Y	Y	Y		Y	Cuba
Salem, Damaj, Hamandi and Zantout [73]	Y	Y		Y	Y	Y			Y	Lebanon
Seyamet al. [79]	Y	Y	Y		Y	Y			Y	USA, Korea
Silva, Polo and Crosby [81]		Y	Y		Y	Y		Y	Y	USA
Tanaka, Ferreira da Silva and Casanova [84]	Y	Y		Y	Y	Y	Y		Y	USA
Tlili, Essalmi, Jemni and Kinshuk [85]	Y	Y	Y	Y	Y	Y			Y	Tunisia
Troussas, Krouska and Sgouropoulou [87]	Y	Y	Y		Y		Y	Y	Y	Greece
Winiecki and Salzman [91]	Y	Y	Y	Y	Y	Y			Y	USA
Wood et al. [92]	Y	Y	Y	Y	Y	Y	Y	Y	Y	USA

interactions between PLAs and regular university teaching assistants, showing a preference of students to interact with their PLAs. While that research offers a case for practical learning, it does not offer any choice of learning anytime or anywhere, limiting the experience to the interaction with the PLAs.

A similar case was found in Seyam et al. [79] who evaluated whether pair programming (an agile software development practice, used in both industry and education, which enforces a role-based approach to learning new programming concepts) would help 53 students during five sessions in a mobile development course to better understand mobile programming. For the evaluation of the experience, observations and questionnaires were used to show a rich experience where programming for mobile devices goes beyond merely writing code. This study shows a practical hands-on learning case for the students in a real-life software development environment, but students did not have the option to choose the way to learn or personalise their learning experience.

Active learning strategies based on students' practical work combined with continuous feedback (such as the Inspection-based strategy based on doing and reflection) are preferred by students in their education. According to Silva, Polo and Crosby [81] using these active learning strategies was not appropriate at the beginning of a course, as their use could confuse students. Students need to have prior knowledge about the content to use active learning strategies.

4.4 | Skills for educators in CS to deliver EDU4

In terms of RQ3, nearly half of the studies ($n = 30$) reviewed made an explicit reference to skills and competences CS educators should have or develop to align their CS teaching to innovative practice. As indicated in Figure 7, studies that referred to skills and competences of educators on average had higher scores on nearly all EDU4 characteristics, with the notable exception of 6) data interpretation. A follow-up ANOVA analysis indicated significant differences with a medium effect size between studies that did and did not mention skills of educators on 3) choice how to learn ($F = 4.758$, $p < .05$, $\eta^2 = 0.069$); 5) hands-on learning ($F = 5.689$, $p < .05$, $\eta^2 = 0.082$); 8) including students' opinion in updating curriculum ($F = 5.922$, $p < .05$, $\eta^2 = 0.085$); and 9) more independent learning ($F = 7.422$, $p < .01$, $\eta^2 = 0.104$).

Furthermore, the aggregate EDU4 score was substantially higher in studies that mentioned skills and competencies of educators ($M = 5.30$, $SD = 2.20$,

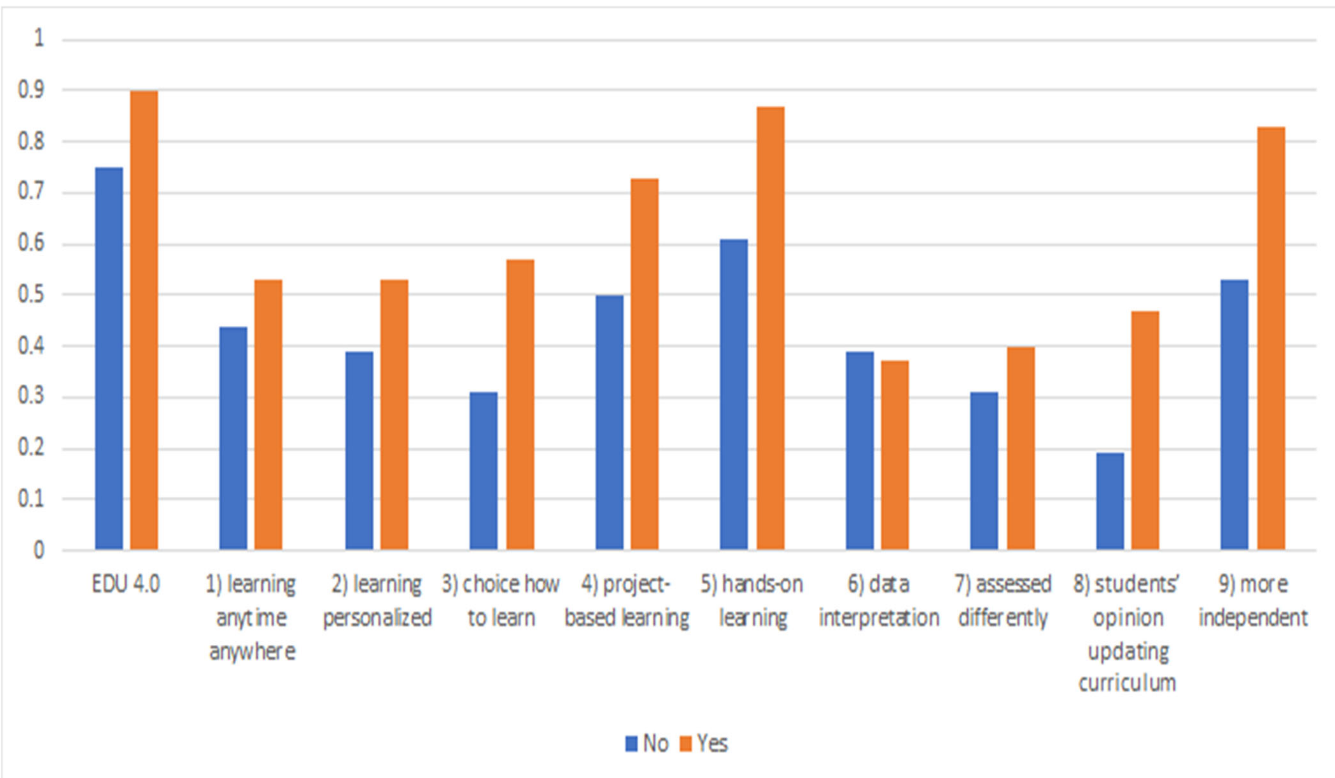


FIGURE 7 Average reference to skills and competencies of educators and EDU4 in 66 studies (0 = not mentioned, 1 = mentioned). n not mentioned skills and competences of educators = 36, n mentioned = 30.

$F = 9.304$, $p < .01$, $\eta^2 = 0.127$) relative to those who did not mention those skills ($M = 3.67$, SD = 2.14). Although no significant differences across the three clusters were found between studies that did or did not mention skills and competencies of educators, 53% of studies who did were part of the full EDU4 cluster, while only 28% of studies who did not were part of that cluster. In other words, those studies that explicitly referred to skills and competencies for educators seemed to be more explicit and innovative in terms of pedagogies and EDU4 elements. An alternative explanation could be that the authors who employed more innovative pedagogical approaches and full EDU4 modes provided more narratives about how educators could effectively support these innovative approaches.

The studies that made an explicit reference to the skills and competencies of HE educators discussed the issue of educators' skills in relation to the implementation and assessment of an innovative learning intervention, which was the main focus of the article. A reference to or discussion of skills and competencies was often presented as an implication of the proposed study rather than being examined as the starting point of a given article. This could be explained by the fact that innovative teaching approaches or interventions are more likely to require educators to develop new skills and competences, and thus such a discussion was seen as very relevant. This observation could explain some of the insights of the quantitative analysis, in particular the observed higher scores on the EDU4 characteristics in studies where skills and competences of educators are discussed.

In terms of the educators' skills discussed in these studies, some of them could be seen as generic, such as the creation of student-centred environments. Others were more concrete, such as the use of specific mobile games in teaching. A recurring theme we identified was the educator as facilitator, moderator or learning consultant [12, 36, 44] as opposed to a educator controlling or being the centre of the learning process [47]. Educators could achieve that by being flexible (adapting to change) [36], supportive, helping students to develop ownership of learning [47], fostering an environment where students take risks and share what they do not know about and where failure is acceptable [12].

This facilitative role was often discussed within a flipped classroom implementation [47] that could give control to students to study the teaching material at their own pace and contact the educator to solve problems and discuss their learning. In such conditions, the educator was monitoring student progress and facilitating understanding through discussions [61]. A educator as facilitator was also seen as the person strengthening communication, ethics, leadership, security and software skills [44]. These conditions point to educators as the agents in charge of developing student-centred learning environments [87].

Educators' skills and competencies were also discussed in relation to the development of more specific expertise, including the use of social network analysis techniques to understand social relationships when students are part of an online network or community [12]. Furthermore, several articles referred to the use of a peer learning assistance approach, that is, having peers to hold office hours, help with labs and facilitate student group work, as they were shown to better support learning than teaching assistants [67], and the use of specific educational games [15, 87] and remote laboratories [14] that could support CS education. In terms of game-based approaches to CS, educators should have the skills to provide tailored and personalised feedback [87] and assign students to game roles within a course management system [15].

5 | DISCUSSION

This study aimed to catalogue and synthesise 'innovative' teaching and learning approaches within CS by aligning it to the relatively new concept of Education 4.0 [7, 79]. Building on a small-scale initial study conducted among European CS educators [71], this review used a three-step coding process to review and synthesise 66 CS studies from across the globe in the period 2016–2020. In line with our initial study [71], in terms of RQ1-2 it seems that across the globe there are roughly three different design flavours (i.e., EDU4 light, project/hands-on, Full EDU4) with regard to innovative practice in teaching and learning in CS.

While none of the 66 studies explicitly mentioned 'education 4.0', which may be due to its recent conceptualisation, all these studies used some form of Education 4.0 or EDU4 characteristics by Hussin [7]. EDU4 light studies mostly focussed on more independent learning any time/anywhere, personalised learning, and choice of how to learn. As the descriptions of these studies show, a large number of technical and teaching innovations have been introduced into the CS curriculum, although they were mainly focused on one or two EDU4 rather than the full nine features. This may be related to the educator's willingness to make some innovations based on the specific problems found in a course, that is to 'update' part of the teaching method instead of completely redesigning a CS course [53, 63].

The second cluster had a strong focus on project-based learning and hands-on learning. These studies primarily used collaborative and project-based learning approaches with some interesting innovations, such as where CS students were considered as potential entrepreneurs [31]. In all 21 studies, great emphasis was placed on practice and project-based learning, enabling CS graduates to develop strong programming and soft skills, usually working together as a team. However, due to the nature of project-based learning, flexibility in anytime, anywhere, personalisation and learning options were often relatively low.

The third cluster encompassed elements from both the first and second clusters and was mainly centred on hands-on learning, more independent, personalised learning, learning any time/anywhere and choice of how to learn. The lowest characteristic of EDU4 in this cluster was the student's ownership of curriculum, although this is significantly higher than the other two clusters. Several innovative and integrated perspectives used game-based learning [87], flipped classrooms [62] and online lab work [14], showing how CS educators can help students to develop coding, project and team skills.

While the three clusters might suggest clear differences in flavour in terms of the EDU4 characteristics, as is evident from both Tables 2–4 and the narratives of the illustrated studies substantial differences were present within each of the clusters in terms of focus, CS skills, technologies implemented and pedagogical innovations. Despite the substantial differences within the practices in each cluster, perhaps the relatively clear cluster formations might indicate several different stages of development and innovation in a respective context. Indeed, in some learning design research, there is emergent evidence of common design practices [53, 58] when comparing different disciplines. In other words, for some contexts, just a small tinkering to a model that mostly works might be appropriate, while in other contexts CS educators might go for a full overhaul of their practice.

In terms of RQ3, about half of the studies made an explicit reference to skills and competencies CS educators should have or develop to align their CS teaching to Education 4.0. Perhaps interestingly, those studies that did refer to the skills and competencies of educators on average had significantly higher scores on nearly all EDU4 characteristics. This might indicate that CS authors who employed more innovative pedagogical approaches, in particular when implementing flipped classrooms or interactive games or lab-exercises, felt the need to provide more detailed narratives about their peer educators, and needed to be aware of the need for additional skills and competences to implement these innovative approaches.

Based upon our systematic literature review, we encourage researchers to investigate how CS courses could be transformed to include more (and perhaps even all) Education 4.0 characteristics, such as those identified by World Economic Forum [93], and whether this would lead to improved skills of CS graduates. It is important for engineering to follow trends in industry, while also providing future anticipation of possible changes. Furthermore, there is an urgent need to critically assess whether the concept of Education 4.0 is useful (or not) for CS and engineering in general. While some of the concepts of active learning, empowering students and hands-on learning are increasingly common and well-supported by robust evidence, more research is needed to explore whether all elements of Education 4.0 are necessarily beneficial for learning or not.

5.1 | Limitations and future research

There is an inherent systemic bias in terms of published outputs, as it is more likely that successful innovations and experiments are published than unsuccessful innovations, as well as 'business as usual' approaches. Furthermore, with the rapid changes in engineering and CS in particular, and the shift in practice due to coronavirus disease 2019 (COVID-19), the reported findings might evolve over time, and perhaps (sub) disciplinary differences might be observed within engineering and CS.

Another limitation is the search string that was used, whereby different key terms of search strings might have resulted in different outcomes. Nonetheless, using a robust 3 phase coding strategy, we believe that we are the first to systematically review the pedagogical learning design decisions that CS educators make when designing innovative practices. Future research should establish which of these common design practices work well for which groups of CS students, and for which specific knowledge, skills and competencies. This will help to strengthen our evidence base and understanding of how to effectively design innovative CS courses that help to empower Education 4.0 in Industry 4.0.

5.2 | Implications for teaching and learning

By using the Education 4.0 characteristics our findings suggest three common flavours that CS educators use to design their practice. As evidenced from this study learning practices associated with Education 4.0 require considerable time for preparation compared with the traditional lecture-based class and this may be overwhelming for some teachers, especially those who have not practised Education 4.0 approaches in the past. Therefore, support should be provided through, for example, teaching assistants, fellow teachers or the reuse of existing activities to help teachers to gradually develop the proposed skills and competences. For teachers comfortable with implementing innovative designs adopting some of the applications and approaches developed within studies of the project-based/hands-on cluster or perhaps even the full EDU4 cluster might be appropriate. For teachers less familiar with innovative design, implementing some of the design elements of EDU4 light studies might be useful.

ACKNOWLEDGEMENTS

The authors are grateful to the two reviewers and chief editor for their detailed and useful feedback. This study was conducted within the project "TEACH4EDU4 Accelerating the transition towards Edu 4.0 in HEIs", financed from the Erasmus+ programme—KA2—Cooperation for innovation and the exchange of good practices.

DATA AVAILABILITY STATEMENT

The data and syntax that support the findings of this study are openly available in Open Research Data Online at 10.21954/ou. rd.22786511.

ORCID

Bart Rienties http://orcid.org/0000-0003-3749-9629
Francisco Iniesto http://orcid.org/0000-0003-3946-3056

REFERENCES

1. N. Aghaee and C. Keller, *Ict-supported peer interaction among learners in bachelor's and master's thesis courses*, Comput. Educ. **94** (2016), 276–297
2. N. Alasbali and B. Benatallah, "Open source as an innovative approach in computer science education a systematic review of advantages and challenges," *2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE)*, 2015, pp. 278–283.
3. F. Alegre, J. Moreno, T. Dawson, E. E. Tanjong, and D. H. Kirshner, Computational thinking for stem teacher leadership training at louisiana state university 1. 2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT), 2020, pp. 1–2.
4. H. W. Alomari, V. Ramasamy, J. D. Kiper, and G. Potvin, *A user interface (ui) and user experience (ux) evaluation framework for cyberlearning environments in computer science and software engineering education*, Heliyon **6** (2020), no. 5, e03917.
5. S. Alsaif, A. S. Li, B. Soh, and S. Alraddady, *The efficacy of facebook in teaching and learning: Studied via content analysis of web log data*, Proc. Comput. Sci. **161** (2019), 493–501.
6. M. Apiola, E. Lokkila, and M.-J. Laakso, *Digital learning approaches in an intermediate-level computer science course*, Int. J. Inf. Learn. Technol. **36** (2019), no. 5, 467–484.
7. A. Aziz Hussin, *Education 4.0 made simple: Ideas for teaching*, Int. J. Educ. Lit. Stud. **6** (2018), no. 3, 92–98.
8. K. A. Behnke, B. A. Kos, and J. K. Bennett, "Computer science principles: Impacting student motivation & learning within and beyond the classroom," *Proceedings of the 2016 ACM Conference on International Computing Education Research*, Association for Computing Machinery, Melbourne, VIC, Australia, 2016, pp. 171–180.
9. B. Berikan and S. Özdemir, *Investigating "problem-solving with datasets" as an implementation of computational thinking: A literature review*, J. Educ. Comput. Res. **58** (2020), no. 2, 502–534.
10. A. R. Bielefeldt, M. Polmear, C. Swan, D. Knight, and N. Canney, "An overview of the microethics and macroethics education of computing students in the united states," *2017 IEEE Frontiers in Education Conference (FIE)*, 2017, pp. 1–9.
11. C. A. Bonfield, M. Salter, A. Longmuir, M. Benson, and C. Adachi, *Transformation or evolution?: Education 4.0, teaching and learning in the digital age*, High. Educ. Pedagog. **5** (2020), no. 1, 223–246.
12. M. Borge, Y. S. Ong, and S. Goggins, *A sociocultural approach to using social networking sites as learning tools*, Educ. Technol. Res. Dev. **68** (2020), no. 3, 1089–1120.
13. M. Borowczak and A. C. Burrows, "Interactive web notebooks using the cloud to enable cs in k-16+ classrooms and pds," Paper presented at 2017 ASEE Annual Conference & Exposition, Columbus, Ohio, 2017. https://doi.org/10.18260/1-2-28571
14. J. Broisin, R. Venant, and P. Vidal, *Lab4ce: A remote laboratory for computer education*, Int. J. Artif. Intell. Educ. **27** (2017), no. 1, 154–180.
15. K. Buffardi and P. Valdivia, "Bug hide-and-seek: An educational game for investigating verification accuracy in software tests," *2018 IEEE Frontiers in Education Conference (FIE)*, 2018, pp. 1–8.
16. A. C. Burrows and M. Borowczak, "Hardening freshman engineering student soft skills," *Session W1A First Year Engineering Experience (FYEE) Conference*, 2017, pp. 1–5.
17. A. C. Burrows and M. Borowczak, *Computer science and engineering: Utilizing action research and lesson study*, Educ. Action Res. **27** (2019), no. 4, 631–646.
18. N. A. Bushmeleva and T. A. Baklashova, *Methodological teaching system of mathematical foundations of formal languages as a means of fundamentalization of education*, EURASIA J. Math. Sci. Technol. Educ. **13** (2017), no. 8, 5141–5155.
19. R. Caceffo, G. Gama, and R. Azevedo, "*Exploring active learning approaches to computer science classes*," *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, Association for Computing Machinery, Baltimore, Maryland, USA, 2018, pp. 922–927.
20. G. Carrascal, A. A. del Barrio, and G. Botella, *First experiences of teaching quantum computing*, J. Supercomput. **77** (2021), no. 3, 2770–2799.
21. M. J. Casañ, M. Alier, and A. Llorens, *Teaching ethics and sustainability to informatics engineering students, an almost 30 years' experience*, Sustainability **12** (2020), no. 14, 5499.
22. J. Chamberlin, J. Hussey, B. Klimkowski, W. Moody, and C. Morrell, "The impact of virtualized technology on

undergraduate computer networking education," *Proceedings of the 18th Annual Conference on Information Technology Education*, Association for Computing Machinery, Rochester, New York, USA, 2017, pp. 109–114.
23. P. Charlton and K. Avramides, *Knowledge construction in computer science and engineering when learning through making*, IEEE Trans. Learn. Technol. **9** (2016), no. 4, 379–390.
24. M. Cobos and S. Roger, *Sart3d: A matlab toolbox for spatial audio and signal processing education*, Comput. Appl. Eng. Educ. **27** (2019), no. 4, 971–985.
25. G. J. M. d. Costa, N. Silva, and T. Fonseca, "*Moral reasoning in knowledge authoring: An e-learning 4.0 analysis!*", Computers in education (Editor) S. Abramovich, Nova Science Publishers, 2012, pp. 135–154.
26. P. Degener, V. Haak, C. Gold-Veerkamp, and J. Abke, "Towards the vision of an lms integrated, browser-based simulation to program lego mindstorms ev3s in ansi-c," *2019 IEEE Global Engineering Education Conference (EDUCON)*, 2019, pp. 89–94.
27. P. E. Dickson, T. Dragon, and A. Lee, "*Using undergraduate teaching assistants in small classes*," *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, Association for Computing Machinery, Seattle, Washington, USA, 2017, pp. 165–170.
28. B. Divjak, B. Rienties, F. Iniesto, P. Vondra, and M. Žižak, *Flipped classrooms in higher education during the covid-19 pandemic: Findings and future research recommendations*, Int. J. Educ. Technol. High. Educ. **19** (2022), no. 1, 9.
29. P. Dondio and S. Shaheen, "Is stackoverflow an effective complement to gaining practical knowledge compared to traditional computer science learning?," *Proceedings of the 2019 11th International Conference on Education Technology and Computers*, Association for Computing Machinery, Amsterdam, Netherlands, 2019, pp. 132–138.
30. European Commission, "Teaching careers in europe: Access, progression and support," Eurydice Report, Luxembourg, 2018.
31. F. Fagerholm, A. Hellas, M. Luukkainen, K. Kyllönen, S. Yaman, and H. Mäenpää, *Designing and implementing an environment for software start-up education: Patterns and anti-patterns*, J. Syst. Softw. **146** (2018), 1–13.
32. W. Fisher, C. Rader, and T. Camp, "Online programming tutors or paper study guides?," *2016 IEEE Frontiers in Education Conference (FIE)*, 2016, pp. 1–6.
33. P. Fisk, "Education 4.0 … the future of learning will be dramatically different, in school and throughout life," vol. 20 January 2021, 2017. https://www.thegeniusworks.com/2017/01/future-education-young-everyone-taught-together/
34. T. Frevert, A. Rorrer, D. J. Davis, C. Latulipe, M. L. Maher, B. Cukic, L. Mays, and S. Rogelberg, "Sustainable educational innovation through engaged pedagogy and organizational change," *2018 IEEE Frontiers in Education Conference (FIE)*, 2018, pp. 1–5.
35. V. Garousi, G. Giray, E. Tüzün, C. Catal, and M. Felderer, *Aligning software engineering education with industrial needs: A meta-analysis*, J. Syst. Softw. **156** (2019), 65–83.
36. P. Gestwicki and B. McNely, *Interdisciplinary projects in the academic studio*, ACM Trans. Comput. Educ. **16** (2016), no. 2, 8.
37. N. Giacaman and G. De Ruvo, *Bridging theory and practice in programming lectures with active classroom programmer*, IEEE Trans. Educ. **61** (2018), no. 3, 177–186.
38. R. Q. Goncalves, von C. A. G. Wangenheim, J. C. R. Hauck, and A. Zanella, *An instructional feedback technique for teaching project management tools aligned with pmbok*, IEEE Trans. Educ. **61** (2018), no. 2, 143–150.
39. C. Goumopoulos, P. Nicopolitidis, D. Gavalas, and A. Kameas, *A distance learning curriculum on pervasive computing*, Int. J. Contin. Eng. Educ. Life-Long Learn. **27** (2017), no. 1–2, 122–146.
40. A. M. Harkins, *Leapfrog principles and practices: Core components of education 3.0 and 4.0*, Futures Res. Quart. **24** (2008), no. 1, 19–31.
41. H. Hosseini, M. Hartt, and M. Mostafapour, *Learning is child's play: Game-based learning in computer science education*, ACM Trans. Comput. Educ. **19** (2019), no. 3, 22.
42. R. J. House, P. J. Hanges, M. Javidan, P. W. Dorfman, and V. Gupta, Culture, leadership and organizations: The globe study of 62 societies, Sage Publications Inc, Thousand Oaks, CA, 2004.
43. Jisc, "Education 4.0 – transforming the future of education (through advanced technology)," 2019. Retrieved January 20, 2021 from https://www.youtube.com/watch?v=aVWHp8FsV1w
44. E. Juárez, R. Aldeco-Pérez, and J. M. Velázquez, *Academic approach to transform organisations: One engineer at a time*, IET Softw. **14** (2020), no. 2, 106–114.
45. M. Knobelsdorf, C. Frede, S. Böhne, and C. Kreitz, "Theorem provers as a learning tool in theory of computation," *Proceedings of the 2017 ACM Conference on International Computing Education Research*, Association for Computing Machinery, Tacoma, Washington, USA, 2017, 83–92.
46. S. Kocdar, A. Bozkurt, and T. Goru Dogan, *Engineering through distance education in the time of the fourth industrial revolution: Reflections from three decades of peer reviewed studies*, Comput. Appl. Eng. Educ. **29** (2021), no. 4, 931–949.
47. C. L Corritore and B. Love, *Redesigning an introductory programming course to facilitate effective student learning: A case study*, J. Inform. Technol. Educ. Innov. Pract. **19** (2020), 091–135.
48. M. C. Lewis and L. L. Lacher, "Teaching modern multi-threading in cs2 with actors," *2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2020, pp. 292–299.
49. Z. Liang and M. A. Chapa-Martell, "A top-down approach to teaching web development in the cloud," *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 2018, pp. 32–39.
50. A. Llorens, J. Berbegal-Mirabent, and X. Llinas-Audet, "*Aligning professional skills and active learning methods: An application for information and communications technology engineering*,", Eur. J. Eng. Educ. **42** (2017), no. 4, 382–395. https://doi.org/10.1080/03043797.2016.1189880
51. L. P. Macfadyen, L. Lockyer, and B. Rienties, *Learning design and learning analytics: Snapshot 2020*, J. Learn. Anal. **7** (2020), no. 3, 6–12.
52. E. Mäkiö, E. Yablochnikov, A. W. Colombo, J. Mäkiö, and C. Harrison, "Applying task-centric holistic teaching approach in education of industrial cyber physical systems," *2020 IEEE*

Conference on Industrial Cyberphysical Systems (ICPS), **1**, 2020, pp. 359–364.
53. K. Mangaroska and M. Giannakos, *Learning analytics for learning design: A systematic literature review of analytics-driven design to enhance learning*, IEEE Trans. Learn. Technol. **12** (2019), no. 4, 516–534.
54. R. P. Medeiros, G. L. Ramalho, and T. P. Falcao, *A systematic literature review on teaching and learning introductory programming in higher education*, IEEE Trans. Educ. **62** (2019), no. 2, 77–90.
55. D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, *Preferred reporting items for systematic reviews and meta-analyses: The prisma statement*, Int. J. Surg. **8** (2010), no. 5, 336–341.
56. D. J. Mundfrom, D. G. Shaw, and T. L. Ke, *Minimum sample size recommendations for conducting factor analyses*, Int. J. Test. **5** (2005), no. 2, 159–168.
57. R. I. Munkvold, "Game lab: A practical learning approach for game development," *European Conference on Games Based Learning*, Academic Conferences International Limited, 2017, pp. 472–479.
58. Q. Nguyen, B. Rienties, and D. Whitelock, "*Informing learning design in online education using learning analytics of student engagement*,", Open world learning: Research, innovation and the challenges of high-quality education (Editors) B. Rienties, R. Hampel, E. Scanlon, and D. Whitelock, Routledge, London, 2022, pp. 189–207.
59. J. A. Parejo, J. Troya, S. Segura, A. Del-Rio-Ortega, A. Gamez-Diaz, and A. E. Marquez-Chamorro, *Flipping laboratory sessions: An experience in computer science*, IEEE Rev. Iberoam. de Tecnol. del Aprendiz. **15** (2020), no. 3, 183–191.
60. H. Park, and Y. Kim, *Clik: Cloud-based linux kernel practice environment and judgment system*, Comput. Appl. Eng. Educ. **28** (2020), no. 5, 1137–1153.
61. L. N. Paschoal, B. R. N. Oliveira, E. Y. Nakagawa, and S. R. S. Souza, "Can we use the flipped classroom model to teach black-box testing to computer students?," *Proceedings of the XVIII Brazilian Symposium on Software Quality*, Association for Computing Machinery, Fortaleza, Brazil, 2019, pp. 158–167.
62. D. Pawelczak, Comparison of traditional lecture and flipped classroom for teaching programming, Proceedings of the 3rd International Conference on Higher Education Advances, Editorial Universitat Politècnica de València, 2017, pp. 391–398.
63. K. Pazur Anicic, B. Divjak, and K. Arbanas, *Preparing ict graduates for real-world challenges: Results of a meta-analysis*, IEEE Trans. Educ. **60** (2017), no. 3, 191–197.
64. C. Peng, "Introductory game development course: A mix of programming and art," *2015 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2015, pp. 271–276.
65. M. S. Peteranetz, A. E. Flanigan, D. F. Shell, and L. K. Soh, *Computational creativity exercises: An avenue for promoting learning in computer science*, IEEE Trans. Educ. **60** (2017), no. 4, 305–313.
66. C. Pilkington, Questioning the value of vodcasts in a distance learning theoretical computer science course. Cham: Springer International Publishing. (2017), 83–98
67. I. Pivkina, "Peer learning assistants in undergraduate computer science courses," *2016 IEEE Frontiers in Education Conference (FIE)*, 2016, pp. 1–4.
68. V. Puncreobutr, *Education 4.0: New challenge of learning*, St. Theresa J. Humanit. Soc. Sci. **2** (2016), no. 2, 92–97.
69. Quality Assurance Agency, "Subject benchmark statement on computing," Quality Assurance Agency, Gloucester, UK, 2019.
70. M. L. Rethlefsen and M. J. Page, *Prisma 2020 and prisma-s: Common questions on tracking records and the flow diagram*, J. Med. Libr. Assoc. JMLA **110** (2022), no. 2, 253–257.
71. B. Rienties, R. Ferguson, C. Herodotou, F. Iniesto, J. Sargent, I. Balaban, H. Muccini and S. Virkus, "Education 4.0 and computer science: A european perspective," *32nd Central European Conference on Information and Intelligent Systems CECIIS 2021*, N. Vrček, E. Pergler and P. Grd (Editors), Faculty of Organization and Informatics, University of Zagreb, Varaždin, Croatia, 2021, pp. 139–146.
72. J. Ruiz, E. Serral Asensio, and M. Snoeck, *Learning ui functional design principles through simulation with feedback*, IEEE Trans. Learn. Technol. **13** (2020), no. 4, 833–846.
73. F. A. Salem, I. W. Damaj, L. Hamandi, and R. N. Zantout, *Effective assessment of computer science capstone projects and student outcomes*, iJEP **10** (2020), no. 2, 72–93.
74. G. Salmon, *May the fourth be with you: Creating education 4.0*, J. Learn. Dev. **6** (2019), no. 2, 95–115.
75. B. S. Santos, J.-M. Dischler, V. Adzhiev, E. F. Anderson, A. Ferko, O. Fryazinov, M. Ilčík, I. Ilčíková, P. Slavik, V. Sundstedt, L. Svobodova, M. Wimmer, and J. Zara, *Distinctive approaches to computer graphics education*, Comput. Graphics Forum **37** (2018), no. 1, 403–412.
76. L. P. Scatalon, R. E. Garcia, and E. F. Barbosa, "Teaching practices of software testing in programming education," *2020 IEEE Frontiers in Education Conference (FIE)*, 2020. pp.1–9.
77. U. Schäfer, "Teaching modern c++ with flipped classroom and enjoyable iot hardware," *2019 IEEE Global Engineering Education Conference (EDUCON)*, 2019, pp. 910–919.
78. M. Seyam and D. S. McCrickard, "*Teaching mobile development with pair programming*," *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, Association for Computing Machinery, Memphis, Tennessee, USA, 2016, pp. 96–101.
79. M. Seyam, D. S. McCrickard, S. Niu, A. Esakia, and W. Kim, "Teaching mobile application development through lectures, interactive tutorials, and pair programming," *2016 IEEE Frontiers in Education Conference (FIE)*, 2016, pp. 1–9.
80. N. Shi, Z. Min, and P. Zhang, *Effects of visualizing roles of variables with animation and ide in novice program construction*, Telemat. Inform. **34** (2017), no. 5, 743–754.
81. P. A. Silva, B. J. Polo, and M. E. Crosby, *Adapting the studio based learning methodology to computer science education*, New directions for computing education: Embedding computing across disciplines (Editors) S. B. Fee, A. M. Holland-Minkley, and T. E. Lombardi, Springer International Publishing, Cham, 2017, pp. 119–142.
82. W. Silva, I. Steinmacher, and T. Conte, *Students' and instructors' perceptions of five different active learning strategies used to teach software modeling*, IEEE Access **7** (2019), 184063–184077.
83. S. Suhaimi, *Education 4.0: The impact of computer architecture and organization course on students' computer anxiety and computer self-efficacy*, Int. J. Adv. Trends Comput. Sci. Eng. **8** (2019), 3022–3025.

84. R. Tanaka, R. Ferreira da Silva, and H. Casanova, "*Teaching parallel and distributed computing concepts in simulation with wrench*," *2019 IEEE/ACM Workshop on Education for High-Performance Computing (EduHPC)*, 2019, pp. 1–9.
85. A. Tlili, F. Essalmi, M. Jemni, and Kinshuk, *Towards applying keller's arcs model and learning by doing strategy in classroom courses*, Springer, Singapore, 2017, pp. 189–198.
86. L. Toetenel and B. Rienties, *Analysing 157 learning designs using learning analytic approaches as a means to evaluate the impact of pedagogical decision-making*, Br. J. Educ. Technol. **47** (2016), no. 5, 981–992.
87. C. Troussas, A. Krouska, and C. Sgouropoulou, *Collaboration and fuzzy-modeled personalization for mobile game-based learning in higher education*, Computers & Education **144** (2020), 103698.
88. B. Tyler and M. Abdrakhmanova, "Flipping the cs1 and cs2 classrooms in central Asia," *2016 IEEE Frontiers in Education Conference (FIE)*, 2016, pp. 1–5.
89. T. Wallner and G. Wagner, Academic education 4.0, International Conference on Education and New Developments, 2016, pp. 155–159.
90. B. Wasson and P. A. Kirschner, *Learning design: European approaches*, TechTrends **64** (2020), 815–827.
91. D. Winiecki and N. Salzman, "*Teaching professional morality & ethics to undergraduate computer science students through cognitive apprenticeships & case studies: Experiences in cs-hu 130 'foundational values'*," *2019 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, 2019, pp. 1–3.
92. Z. J. Wood, J. Clements, Z. Peterson, D. Janzen, H. Smith, M. Haungs, J. Workman, J. Bellardo, and B. DeBruhl, "*Mixed approaches to cs0: Exploring topic and pedagogy variance after six years of cs0*," *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, Association for Computing Machinery, Baltimore, Maryland, USA, 2018, pp. 20–25.
93. World Economic Forum, *Schools of the future: Defining new models of education for the fourth industrial revolution*, World Economic Forum, 2020. https://www3.weforum.org/docs/WEF_Schools_of_the_Future_Report_2019.pdf

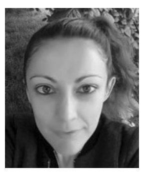
AUTHOR BIOGRAPHIES

Bart Rienties received his PhD at Maastricht University in the Netherlands in 2010. From 2010–2013 he worked as (senior) lecturer at University of Surrey, and since 2014 has worked at the Institute of Educational Technology at the Open University UK. He has been a Professor of Learning Analytics since 2017 and has published over 300 articles and conference proceedings on learning analytics, learning design, computer-supported collaborative learning, internationalisation and professional development.

Rebecca Ferguson received her PhD from The Open University in the UK in 2010. Since then, she has worked at The Open University and was appointed as Professor of Learning Futures in 2020. Her publications include Educational Visions: The Lessons from 40 Years of Innovation (London 2019) and Augmented Education (New York 2014). Her research focuses on innovations in pedagogy, online learning, learning at scale and learning analytics.

Dalibor Gonda obtained a PhD degree at the University of Constantine the Philosopher in Nitra, Slovakia. Since 2016, he has been working as an assistant at the University of Žilina in Žilina, where he lectures on statistics and algebra. He researches various aspects of teaching mathematics, statistical data processing and the use of spatial statistics in location tasks.

Goran Hajdin received his PhD at the University of Zagreb in Croatia in 2014. From 2007 he works at Faculty of Organisation and Informatics, University of Zagreb on the courses related to pedagogy, didactics and methods of teaching informatics. He is a head of lifelong learning module for teacher education since 2017. He publishes research on topics related to pedagogy, educational technology and teaching methods with special focus on informatics.



Christothea Herodotou received her PhD from the Institute of Education, UCL in 2009. She worked as a primary teacher for 7 years and since 2013 as a postdoctoral researcher at the Open University UK. In 2021, she became a Professor of Learning Technologies and Social Justice. She is now leading two award-winning university-wide initiatives about community citizen science (nQuire) and predictive learning analytics (Early Alert Indicators dashboard).

Francisco Iniesto received his PhD at The Open University in the UK in 2020. His background is as a Computer Engineer with extensive experience in IT consulting and software development. From 2018 he works as a Research Associate and Associate Lecturer at the Institute of Educational Technology at the Open University. His areas of research and publications are in inclusive design, accessible educational technology and open education.

Ariadna Llorens Garcia (Member, IEEE) is an associate lecturer in the Management Department at the Universitat Politècnica de Catalunya–Barcelona Tech (UPC). She is an Industrial Engineer and holds a PhD in Management from UPC and a MBA in Administration from ESADE. Her research interests are in the fields of engineering education, university-business cooperation and innovative learning methodologies.

Henry Muccini has a PhD in Computer Science and is a Professor at the University of L'Aquila, Italy. His research interests are in the Software Engineering field, and specifically on software architecture descriptions and analysis, machine learning for architecting quality systems, software migration to microservices and model-driven software architectures.

Julia Sargent received her PhD at Loughborough University in the UK in 2018. From 2018 she worked as a Research Fellow at the Open University and since 2019 has worked as a Lecturer at the Institute of Educational Technology at the Open University UK. She currently works in the curriculum part of IET. She publishes research on aspects such as educational technology, physical education and pedagogy.

Sirje Virkus received her PhD at Manchester Metropolitan University in the UK in 2011. From 1985 she has worked at Tallinn University as a lecturer and since 2012 as a Professor of Information Science. She is the Head of the Study Area of Information Sciences at the School of Digital Technologies of Tallinn University. Her research interests are focused on the development of information-related competencies, ICT innovation in education and internationalisation. She has written more than 200 research publications.

Maria Vittoria Isidori received her cognitive neuroscience PhD at the Padova/L'Aquila University in Italy in 1998. From 2008 she works at Department of Human Studies University of L'Aquila on the courses related to pedagogy, didactics and methods of inclusive teaching. She is a head of lifelong learning module for teacher education since 2014. She publishes research on topics related to pedagogy, learning process, inclusive didactic, educational and teaching methods with special focus on Special Need SEN.

How to cite this article: B. Rienties, R. Ferguson, D. Gonda, G. Hajdin, C. Herodotou, F. Iniesto, A. Llorens, H. Muccini, J. Sargent, S. Virkus, and M. V. Isidori, *Education 4.0 in higher education and Computer Science: A systematic review*, Comput. Appl. Eng. Educ. (2023), 1–19. https://doi.org/10.1002/cae.22643