2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
Paper ID #39653

A Systematic Literature Review Examining the Impacts of Integrating Computer Science in K-5 Settings

Alaina Katherine Mabie
Monica McGill, CSEdResearch.org

Monica McGill is President & CEO of CSEdResearch.org. Her areas of scholarship are K-12 computer science and cybersecurity education research with a current focus on diversity, equity and inclusion as well as improving the quality of research.

Brenda Huerta, Bradley University

A Systematic Literature Review Examining the Current Landscape of Computer Science Integration in K-5 Settings

Alaina Mabie[1], Monica M. McGill[2], and Brenda Huerta[3]

[1,3]Bradley University
[1,2,3]CSEdResearch.org
[1]amabie@mail.bradley.edu , [2]monica@csedresearch.org,
[4]bhuerta@mail.bradley.edu

Abstract

Problem. Computer Science (CS) is in its early stages of being taught to K-5 students within the United States. It still remains unknown how best to teach CS to students; however, evidence suggests that integrating CS into other subject areas may yield positive learning gains. Further, as teachers struggle to find time to add another subject, teaching integrated CS may provide them them time to teach CS without sacrificing other subject areas.

Research Question. Our research question is: *What does existing literature indicate as promising practices when integrating CS into other subjects?*

Methodology. We conducted a systematic literature review to identify articles that investigate the integration of CT and/or CS in K-5 learning environments.

Findings. We found 243 articles that met our initial requirements, which then were reduced to 29 articles. We extracted the findings from these articles, including the subject areas in which CS was integrated. Our review shows that there are several noted promising practices for integrating CS and computational thinking into K-5 learning environments, with a particularly focus on integrating CS into STEM. We also found that constructive approaches and unplugged activities can enhance learning.

Implications. Based on this literature review, we identified several promising practices for creating curriculum that integrates K-5 into learning environments. These practices provide guidance for curriculum designers and those creating resources and tools for teaching K-5 students CS.

1 Introduction

CS education has been slowly entering the K-12 education system, particularly at the high school level where over half of the schools in the United States offer a CS course [1]. There remain wide gaps in who has access to and is participating in CS education at the high school level [2]. Previous research highlights the fact that marginalized groups can feel that they don't belong in CS as early as 2nd and 3rd grades [3]. One solution is to bring CS education to lower grades so that belongingness (as well as knowledge) can be cultivated in the critical formative years. However, while middle school also has grown, the number of states that require CS in K-5 and the number of schools teaching CS in K-5 still remains low [1].

Some of the key barriers to offering CS to elementary school students that have been found include administrators are not supportive, teachers have not yet received training to teach CS, and resources for adoption remain low [4–7]. A key barrier that has been mentioned in past studies is that teachers have no time to add an additional subject area to their day, particularly since they are immersed in teaching to their state standards which more heavily emphasize language arts and mathematics [4].

Integrating CS and computational thinking (CT) into subjects such as math and language arts has been viewed as a way to mitigate the barrier related to time [8] and to create innovative learning environments [9]. Integration also provides opportunities to link CT and CS more closely to mathematics, engineering and science [9, 10], given the shared learning processes and contexts across the fields. It also recognizes that interdisciplinary education can benefit student learning and is often the core at K-5 learning [11], how integration occurs and how impactful it can be on student learning still remains unexplored. Our research question for this study was: *What does existing literature indicate as promising practices when integrating CS into other subjects?*

To answer this question, we conducted a systematic literature review using the Khan et al. methodology. Systematic literature reviews for integrating CS have also been conducted. For example, Rich et al. conducted a literature review in which the authors explore and catalog CS learning goals believed to be useful to teach students. The researchers explored learning goals which have previously been analyzed and researched within students grades K-8, connecting these two bodies of literature to explore how well expert theories on potential learning goals align with learning goals that have been researched with students.

We provide our results of our systematic literature review here. This research is important for learning what studies currently exist in K-5 integrated CS, so that we can identify emerging patterns and promising practices in this nascent field. CS/CT is highly relevant for engineering work, engineering studies, and engineering ways of knowing – not only in professional practice or at the undergraduate level, but how CS/CT is relevant for engineering learning in K-12 settings.

2 Background

Despite the many benefits integrating CS into existing subjects in elementary schools is thought to have, it also has its challenges. Howard conducted a qualitative study examining five educational technology leaders about the challenges K-5 CS teachers face in the classroom and what they need to be more effective and efficient at integrating CS into subject areas [14]. The authors identified challenges as factors related to teacher motivation and interest and ongoing professional needs of CS teachers so they are fully prepared to do so. Additional challenges included teacher uncertainty about assessment and a need for additional general technology training. The researchers found two main factors that influence teachers' desire to teach CS: student motivation to engage in CS and parental interest in having their children learn CS.

Yadav et al. conducted a study to understand how teachers and CS educators can collaborate to develop pathways to help pre-service teachers become computationally educated. Yadav et al. found that teachers (n=9) struggled to define what CT entailed and most could not differentiate it from technology [15]. As evidenced by the post-surveys, teachers were able to better understand and explain CT after engaging in a training specifically focused on CT. The teachers also stated that they believe CT can be implemented into classrooms universally across all subjects. The authors conclude by emphasizing that the preparation of pre-service teachers through teacher-education programs and collaboration between educators may be a promising practice for formulating better methods for CS integration into the K-12 curriculum.

Sands et al. & Garvin et al. both found that teachers struggle to understand and define the complexities of CT [5, 16]. Garvin et al. also found that approximately half of the participating teachers reported feeling comfortable teaching and integrating CT into the classroom. Despite the high comfort levels, only 43.8% felt they had adequate training in CT.

However, training teachers is not a small task. Rich et al. found that to enable teachers to teach CS and CT in an integrated manner, teachers need support staff, physical resources, willingness to experiment, and the ability to notice student responses. Teachers faced many obstacles that impeded successful integration of CS and engineering into the elementary curriculum. They struggled to justify the time spent on CS and engineering when it was not part of standard, state-mandated curriculum. Despite the obstacles, support staff of PD researchers was a very helpful resource that enabled integration.

3 Methodology

To mitigate these challenges, it is first important to identify promising practices and techniques so teachers can be trained in those practices. We wanted to learn more about what emergent promising practices might be found in the existing literature. To answer our research question, *What does existing literature indicate as promising practices when integrating CS into other subjects?*, we used the Khan methodology [12] for conducting the systematic literature review. This methodology consists of 5 steps:

Step 1: Framing questions for a review.
Step 2: Identifying relevant work
Step 3: Assessing the quality of studies
Step 4: Summarizing the evidence
Step 5: Interpreting the findings

Steps 1 through 3 belong to the methodology and are discussed in this section. Step 4 (results) and Step 5 (discussion) are each discussed in separate sections.

We also created a PRISMA chart to show our article selection process, including why articles were excluded [18]. This is presented in Figure 1.

3.1 Step 1: Framing the questions for review

Using Khan et al.'s steps, we created a free-form question based on the goals for this study, resulting in *What does existing literature indicate as promising practices when integrating CS into other subjects?*

We formed structured questions from the free-form question, breaking them apart into Khan et al.'s categories. This included the following:

- Population: K-5 students
- Interventions or exposures: Integrated CS

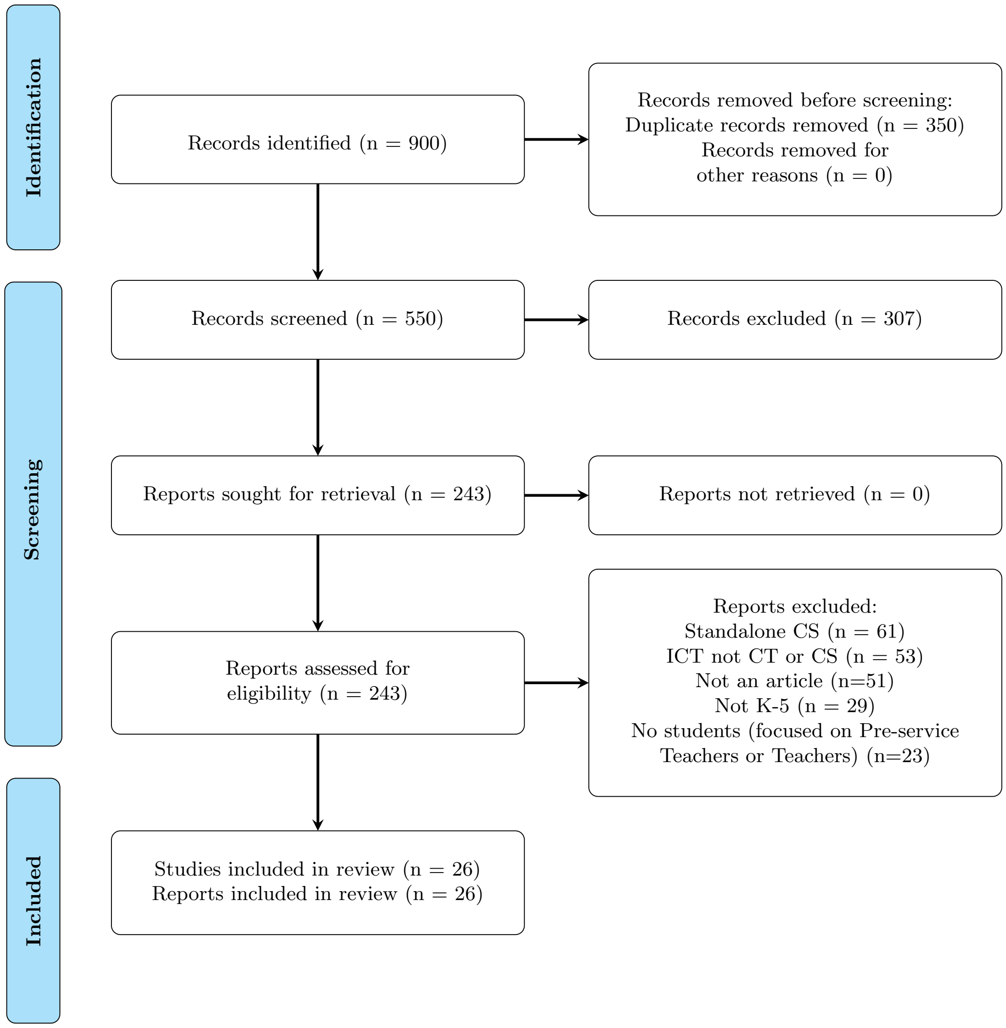

Figure 1: PRISMA chart for article searches [18].

- Outcomes: Students' experiences learning CS
- Study design: Review articles that study integrated CS and student outcomes

3.2 Step 2: Identifying relevant work

To engage in the search, we first established a set of search terms:

Search 1: ”Integrated” and ”Computer Science” and ”K-5”
Search 2: “Integrated” and ”Computer Science” and (”Elementary” or ”primary”)
Search 3: “Integrated” and ”Computer Science” and ”Kindergarten”
Search 4: “Integrated” and “Computing” and “K-5”
Search 5: “Integrated” and ”Computing” and (“Elementary” or “primary”)
Search 6: “Integrated” and ”Computing” and “Kindergarten”
Search 7: ”Computational thinking” and ”K-5”
Search 8: ”Computational thinking” and (“Elementary” or “primary”)
Search 9: ”Computational thinking” and ”Kindergarten”

We used Google Scholar to search for these terms, as it has been shown to be an effective way to conduct searches for systematic literature reviews [19]. We also set additional criteria for our search. Articles in our search had to be:

- Published between 2015 and 2022
- Research focused on integrated K-5 CS and/or CT (all aspects)
- Research with a focus on K-5 students

Given the tens of thousands of articles returned for just the first article search (33,402), it would have been impractical for our team to manually review each of them. There is a law of diminished returns for Google searches, and, for practical reasons, we chose to limit our manual search to the first 100 articles from each keyword search, providing us a robust set of 900 articles to manually review. We decided upon 100 given 1) the newness of the field and our hypothesis that we would not find many articles that meet our search criteria and 2) staffing and project scope.

We chose the years 2015 through 2022 to align with the White House administration's call for CS for all in 2016, hypothesizing that it was after this time the majority of the research would emerge [20]. Each article that met the criteria were included in our results regardless of their publication venue[1] or type and were added to a shared spreadsheet for further screening. This includes articles on studies with in-school and out-of-school settings.

3.3 Step 3: Assessing the quality of studies

For each study, we first determined the type of paper. We excluded experience reports, extended abstracts, and other articles were removed from the set of articles we were to review (n=51). We chose to keep all of the remaining studies in the set of articles to review, rather than extracting only anonymous or dual-anonymous articles or articles that only appeared in journals. This was decided since the field of K-5 CS integration is very nascent, though we expect this field to grow tremendously over the next decade. We further eliminated the following articles focused on:

- Standalone CS (n = 61) - since our focus was CS integrated into other subject areas.
- Information and Communications Technology (ICT) not CT or CS (n = 53)
- No students in the study (focused solely on Pre-service Teachers or Teachers) (n=23)

4 Step 4: Summarizing the Evidence

From our studies, we grouped together findings from the results in several categories: Administration and teacher perspectives as well as the actual subject areas for integrating CS/CT. Within each subject area, we found a mix of articles that had teacher participants, student participants, and sometimes both.

4.1 Math

Strickland et al. conducted a study with 16 teachers in 2018-2020 who went through PD workshops that introduced them to the Scratch environment and Action Fractions materials [21]. The teachers were given twenty-five lessons that integrated mathematics with CS. Sequences, decomposition, repetition, conditionals, variables, and debugging were interspersed throughout the main math curriculum. The lessons were created based on six design principles "...to address

[1] Including ASEE articles that are part of the Google Scholar search corpus

barriers that generalist elementary school teachers may face as they attempt to implement integrated CS curriculum" [21, p. 1153]:

- Mathematics Focus
- Following CS LTs
- Using a familiar lesson structure
- Universal Design for Learning (UDL)
- TIPP& SEE
- Comprehensive supports for lessons

These lessons were then used to teach 362 third and fourth-grade students over 2018-2020. The students kept journals throughout the lessons, and surveys and interviews were given to both students and teachers. Results indicated that integration of mathematics and CS in elementary curriculum is possible despite the challenges they encountered, such as nearly 30% of teachers indicating that they needed more time to complete the lessons. The qualitative analysis indicated that students were engaged and enjoyed the lessons and that students "I can" statements in a pre- and post-test significantly increased.

Niemelä et al.'s study focused on integrating CS into mathematics without hindering the quality of mathematics and using crafts. Year 1-9 teachers (n=206) in Finland participated in curriculum development courses and then used their new curriculum in their classrooms [22]. The curriculum used arithmetic, algebra, and geometry for the mathematics section, and for CT focused on abstraction, automation, and analysis, with variations across the 9 years. Quantitative analysis revealed that 1) geometry was the most popular subject for teachers to integrate CS with 54.7% of teachers preferring geometry and 2) teachers were quite good at creating integrated lesson plans. Some expressed concerns about students not being interested in mathematics and CS because mathematics has a reputation for being hard. The creative aspect of the lessons seemed to be difficult for some of the teachers because they could not see how programming could be creative.

In a single-case qualitative study, Haspekian conducted a case study focused on one fourth-grade teacher with no CS experience who volunteered to use Scratch in their classroom [23]. The goal of the study was to understand the teacher's choices when integrating Scratch. The first lesson developed by the teacher consisted of "free discovery" of Scratch so students could explore and play with the program. The second session consisted of mathematics and CS which was too difficult for the students and the teacher did not have enough understanding of Scratch to support the students. As the teacher spent more time with Scratch, their lessons became more complex and with enough training, they were able to effectively support students.

Bártek conducted a study to explore how often primary and secondary CS teachers used interdisciplinary relationships, primarily in CS and mathematics, and their opinions on appropriate topics for inclusion in the curriculum [24]. Quantitative analysis revealed that teachers (n=123) preferred to use MS excel (99 absolute frequency) and Geogebra or Cabri Geometrie when integrating CS into mathematics. A majority (61%) of teachers felt the need to change the curriculum of CS-related subjects to be able to integrate easier.

Ahmed et al. conducted a qualitative study (2017-2020) on a teacher PD project that enabled them to plan, execute, and evaluate lessons that integrate programming into different subjects, but primarily mathematics [25]. Primary school teachers (n=28) completed all of the lesson plan courses and submitted a detailed presentation of each of their lesson plans, including analog, robot, and block-based programming through Scratch or Scratch Jr. Five categories emerged: didactic methods of the teachers, math content, programming language and tools, opportunities identified in programming instruction, and challenges identified in programming instruction. They found that teaching analog programming before robot or block-based programming allows students to understand the step-by-step process necessary for CS. They also found that the robot was a powerful tool to keep younger students engaged with CS concepts and practices. The robot programming led to student collaboration, increased motivation for mathematics, and an introduction to block-based programming. The various types of programming made integrating CS into mathematics easier.

Luo et al. conducted a year-long qualitative study to explore the impacts of third and fourth-grade students (n=22) that participated in eight math and CT lessons focused on sequencing, conditionals, repetition, and decomposition [26]. Action Fractions lessons along with Scratch were used for these integrated lessons. Teachers were provided with PowerPoint, instructional support videos, and two researchers during the integrated lessons. Students were assessed on their CT knowledge three different times through cognitive interviews using the think-aloud technique, coding scenarios in Scratch, word problems embedded in the math context, and number problems. Analysis revealed that the students' sequence thinking had improved throughout the assessments. Students' understanding of conditionals was inconsistent with number comparisons and students showed no understanding when evaluating conditions using user input. Students showed evidence of understanding and recognizing repetition but struggled with constructing and recognizing repeat instructions in word problems. Students also were able to use decomposition in number problems but had difficulty decomposing the math-integrated word problem.

In another study, Luo et al. conducted a mixed methods study that included students from two fourth-grade classes from a Midwestern elementary school that implemented a series of eight CT integrated mathematics lessons [27]. The lessons included CT concepts such as sequences, repetition, decomposition, conditionals, debugging, and variables. Four of the eight lessons focused on variables, which was a focus of this study and included unplugged activities. Using assessments and think-aloud interviews, descriptive statistics revealed that most students had not progressed beyond an understanding of storing data. Qualitative analyses corroborated the quantitative data showing that students had difficulty with articulating how to internalize or update a variable, accept and store user input in a variable, reference data/user input using a variable, and reference variables in a boolean expression to control the program flow.

Kopcha et al. conducted a study focused on training 12 elementary teachers in rural under-resourced areas who went through a week-long summer workshop to learn how to integrate educational robotics (Ozobot robots) into their mathematics curriculum [28]. The teachers also had monthly follow-up training as well as in-classroom support. The PD workshop started with the teachers going through the CS activities as if they were the students so they could understand the problems their students might face. Surveys and assessments were given to the teachers before the PD and ten months later at the end of the school year. The lesson plans teachers developed throughout this study were evaluated on mathematics, cognitive demand, student access to mathematics, student agency, and the assessment. Quantitative analysis revealed that teachers' ability to use robots to teach math, their ability to help students when having difficulty with robots and their confidence in teaching mathematics through robotics had significantly improved after the workshop ($p < .01$).

Rich et al. explore how learning trajectories (LTs) might be used to design variables instruction, a concept associated with computer science (CS) education [29]. To do this, the authors aimed to develop an LT for variables and use it to guide curriculum development for fourth graders working in Scratch in an integrated mathematics CS curriculum. The authors do not make any definitive claims about student learning based on this data, as the student pages and teacher evaluations do not directly measure student learning. However, they present the data as evidence that their proof-of-concept attempt at using a literature-based progression and a process of activity development to create a context-specific LT for variables produced a curriculum that is accessible to students and feasible for use in classrooms.

4.2 Science

With respect to integrating CS into science, we found two articles. Celepkolu et al. conducted a study into upper elementary and middle-grade teachers' perceptions and concerns with CS implementation [30]. Teachers attended a five-day PD workshop to learn how to integrate CS into biology, including incorporating loops, conditionals, expressions, variables, and object-oriented programming through Snap!. Results from the CS Attitude Survey and a knowledge assessment revealed that the teachers' achieved significant improvements in technical skills; however, they still lacked confidence in their understanding.

Luo et al. conducted a case study with two elementary girls who participated in a four-week-long summer CS integrated biology unit. This unit taught about the reproduction cycle of a flowerless plant through Dash robots and the Blockly app [31]. Results indicated that the girls demonstrated behavioral, emotional, and cognitive engagement throughout the sessions despite an initial low interest in science. The girls also were able to use sequences, loops, conditionals, abstraction, decomposition, iteration, and debugging throughout the activities despite not being taught all of the concepts. Integrating CS into their summer biology lessons avoided the challenges often faced in teaching this way in the classroom while still benefiting the students.

4.3 Engineering

Engineering was another area most commonly mentioned for integrating CS into the K-5 classroom. This includes using robots as tools for learning. Sáez López et al. conducted a study that combined programming and robotics to teach basic computational concepts such as sequences, looping, conditional statements, threads, event handling, user interface design, and keyboard input. A variety of tools were used throughout the study (i.e., M-bot, Dash and Dot, Ozobot, Blockly, Scratch). Fifth-grade students (n=107) from three different elementary schools participated and a control group. Data was collected on active learning, understanding of computational concepts, perceived usefulness, and enjoyment through surveys given after the intervention. Findings revealed that there were significant improvements regarding all computational concepts, including statistically significant differences with active participation, efficiency, learning performance, enthusiasm, motivation, sense of fun, and comfort. Statistical significance was not found with perceived usefulness or interest in the subject.

Chalmers conducted a mixed-methods study and worked with teachers (n=4) from four primary schools to implement the WeDo 2.0 robot kits in their classrooms [33]. The teachers were given fifteen robot kits for six weeks without any direction on how to integrate the robots into their classrooms. Teachers completed questionnaires, interviews, journals, and reflections with their perceptions of the robot kits both before and after the integration. The results indicated that "...exploring with and using the robot kits, and activities, helped the teachers build their confidence and knowledge to introduce young students to computational thinking. The study identified that teacher professional development (PD) needs to focus explicitly on how to teach developmentally appropriate robotics based STEM activities that further promote computational concepts, practices, and perspectives." [33, p. 1]

In another study focused on integrating CS with robotics, Sullivan and Bers integrated KIWI robotics kits into a preschool through second-grade curriculum [34]. PreK-2 students (n=60) participated in introductory robotics and programming lessons that were taught once a week for approximately one hour. In the first two lessons, students were able to explore the basic parts of the KIWI robot and learned how to sequence a program. Preschool students spent the whole course reinforcing these concepts. The third lesson focused on sound sensors and students used a wait-for-clap block in their program to incorporate the sensor. Additional lessons focused on loops, conditional statements and creating floor maps of their neighborhood, then programming their robots to drive along the map, stopping at significant locations and carrying out different actions. Students were assessed on their programming knowledge, different KIWI robotics parts and their functions. The results indicated that all the children had 1) a good understanding of the functions of each part of the KIWI robot and 2) improved their programming knowledge of sequencing.

First through fourth grade students from Portugal participated in a study conducted by Brigas and Figueiredo focused on the Hour of Code initiative [35]. This study defined CT as pattern recognition, creating and using algorithms, decomposition, and understanding abstractions. The activities in this workshop focused on solving problems by creating and using algorithms. Findings indicated that students demonstrated the ability to find multiple solutions to problems. The students also worked collaboratively to solve the problems using their CT skills.

4.4 Language Arts

Language arts is a commonly-tested subject and is the subject most taught to students in K-5, either directly or indirectly. We found three studies that described integrating CS into language arts. In the first, Bers describes the pedagogical approach for early childhood CS called Coding as Another Language (CAL) along with six coding stages/learning trajectories that aligned with Piaget's levels of cognitive development [36]. Coding as Another Language is grounded on the principle that learning to program involves learning how to use a new language for communicative and expressive functions. Bers notes that

> CAL is grounded on the principle that learning to program involves learning how to use a new language (a symbolic system of representation) for communicative and expressive functions. This paper proposes that, due to the critical foundational role of language and literacy in the early years, the teaching of computer science can be augmented by models of literacy instruction. CAL supports young children in

transitioning through different six coding stages.

Bers then investigated the use of CAL through six case studies which demonstrated its effectiveness.

Another study focused on how CS can be integrated into literacy through digital storytelling [37]. Whyte et al.'s study consisted of a six-week after school program for fifth graders (n=12) that had eight structured tasks ranging from introducing a character to coordinating multiple narrative elements simultaneously. The ninth and open-ended task required that the student create an original narrative project that included events, actions, plot, etc. Students were encouraged to write short programs following the structured tasks; however, most of the students used wait commands and coordinated each element to sequence appropriate to their narrative. This resulted in participants taking much longer to complete their projects and required persistent testing and debugging of script. Overall results indicated that 1) integrated programming into literacy activities supports novice learners' understanding of algorithm design and program execution and 2) learners use of programming supported their ability to create digital stories by visualizing narrative structure.

A very similar study also conducted by Whyte et al. also consisted of a six-week afterschool club that focused on integrating CS with literacy through Scratch [38]. This study involved teaching eight to eleven year old students (n=10) six tasks that focused on algorithm design, program execution and loops for programming, representation choice, and structural and visual design for storytelling ranging from deciding on representation and content for specific purposes to programming an original multimodal story. Qualitative analysis revealed that students could achieve their narrative goals and programming goals. The task-oriented design allowed students to focus on the designated concept.

In a study investigating two third-grade teachers integrating CS into existing engineering curriculum, Sullivan et al. studied the lesson plans that the teachers developed. The plans focused on writing instructions for building an engineering structure as a 3D model. The teachers focused on two disciplinary activities that support learning through productive frustration and the importance of precision. Precision in CS is emphasized in algorithms and engineering (in the development of blueprints and structures). The students who participated in the unit needed help writing precise instructions, which is where they experienced both frustration and failure. Researchers noted these experiences as valuable for learning and improving the students' problem-solving skills.

4.5 Multi-Subjects

Several studies reached beyond integrating CS into one subject area, and instead took a more multidisciplinary approach. Leonard et al., for example, conducted a study using a unique approach to CS integration, integrating several subjects including biology, dance, and language arts [40]. This multi-modal approach proved to be a successful method of integration with high student engagement and increased student CT skills. The lessons allowed students to explore parts of the cell, choreography, and CS from different perspectives. The curriculum included sessions taught by a team of researchers once a week for eleven weeks [40]. Fifth graders (n-44) were taught choreography inspired by the parts of a cell such as a cell membrane, nucleus, vacuoles, and cytoplasm. The students were also taught choreography for a poem about cells, then how to program a virtual character to dance the choreography that they learned. Findings revealed that students' CT skills increased significantly between the pre- and post-test and students were engaged in and enjoyed the learning process.

Playton et al. conducted a case study focused on whether increasing student exposure to STEM integrated with CS (STEM+C) positively impacts their attitudes, interests, engagement, and knowledge of CS careers [41]. Rural fourth graders (n=34) were given 18 contact hours over a year with three inquiry-based STEM+C units. The units included designing and testing a moving object, creating sculptures with circuitry, and developing an ecosystem video game. They assessed the students' engagement and attitudes towards STEM+C. Paired T-tests revealed significant positive increases in girls' attitudes toward mathematics and science. Significant increases were also observed in all students' attitudes toward science. S-STEM results indicated that students' perceptions of their math and science performance increased. Students also gained an understanding of STEM+C careers with the largest gains in girls' knowledge of scientists and computer scientists. Findings also revealed that students reported very high to high levels of affective and overall engagement.

Lin conducted a study focused on how commercially available technology toys such as littleBits and KIBO can help promote the development of CT in K-3 elementary students [42]. This study assessed if the environment impacts students' engagement with the technology toys, including a child-friendly laboratory space using littleBits and KIBO, a college classroom using littleBits (girls only), and a kindergarten classroom at an elementary school using littleBits and LEGO WeDo. Data collection for this study included pictures, videos, transcriptions, pre-task questionnaires by the parents, post-task interviews from the participants, and teacher interviews. Results indicated that in the lab setting, all participants were able to engage in learning while using littleBits and KIBO. They showed signs of reflection, coordination, problem-solving, and excitement. Participants showed more disengagement with the KIBO. Results from the second location indicated that littleBits can help girls in early elementary school engage in play and learn CT skills. They also demonstrated problem-solving, reflection, engagement, and collaboration. Results from the final location indicated that littleBits can be used in the classroom as a learning tool that promotes learning, engagement, complexity, and collaboration. The students demonstrated problem-solving, CT, and reflection.

Century et al. conducted a study focusing on a district's strategy for implementing CS into its curriculum [8]. Broward County Public Schools in Florida created six transdisciplinary modules that included problem-based language arts, science, and social studies lessons associated with CS lessons from Code.org "Fundamentals" program and Scratch activities for grades 3-5. Sixteen elementary schools were enrolled in the study with 321 teachers and 5,791 students. The teachers and support staff participated in a three-day PD institute before implementing the modules and attended a two-day institute during the school year. An example of one of the modules was the fourth graders' Invasive Species. This module focused on the invasion of Burmese pythons into the Everglades and integrates science by looking into the basic needs of living things, social studies with local and state governments and civic engagement, CS with crowd-sourcing, sequencing, conditionals, events, functions, programming, debugging, and models/simulations, and English with researching, reading, writing, and presenting.

Teachers had assigned grade-level CS lessons but they were also allowed to implement more CS activities into their classroom or to teach non-grade level assigned lessons. Qualitative and quantitative data revealed no significant difference in attitudinal or academic achievement outcomes. However, there was a positive association between teaching a higher amount of non-grade level assigned CS lessons with higher academic achievement outcomes in language arts and mathematics. Teachers' innovativeness was positively associated with student CS identity attitudes. The study also found a negative association between the percentage of grade-level assigned CS lessons completed and Achieve3000 scores and teachers' resourcefulness and coping, and students' science scores.

Jurado et al. conducted a study with four elementary teachers in three schools in Catalonia who were trained to introduce robotics in the classroom to 75 students [43]. CS learning was integrated using the KIBO robot 18 Kit. Each school acquired 1 to 4 units of this educational robot, which were then used in groups of 4 to 15 students. The curriculum consisted of 16 sessions lasting between 45 and 60 min, organized around CS concepts introduced along the way, such as instruction, sequence, iteration, and conditional. In addition, each of the 16 sessions had activities related to interdisciplinary STEAM learning and was closely inspired by the results of a previous pilot experience and by scientific literature related to the robotic tool. Students' learning was higher in general and particularly so in the classes where the teachers had a higher initial interest in the training. Interestingly, children with lower marks improved especially in conduct and creativity but did not improve in collaboration and communication; children with higher marks improved in collaboration, communication, and creativity and decreased in content creation.

In terms of student learning: robotics helped all the students in their problem-solving abilities, small groups of fifteen students were more suitable than larger groups of 30 students during the robotic session, psycho-motive and assembly-related activities were successful among the students, activities involving repeating patterns presented difficulty, and collaborative learning was complex to implement since some students monopolized the robot. Teachers perceived the training to be supportive and useful and ended the school year feeling confident with the used robotic platform (KIBO). Student increases in problem-solving skills is aligned with findings from prior research indicating that the use of robotics can lead to improvements in problem-solving skills.

In a study illustrating how a school started with no computing or CT and developed many successful integrated computing lessons despite the various challenges and barriers involved in the implementation of CS, Israel et al. focused on how teachers with limited CS experience integrate CT into their classrooms [44]. A cross-case analysis with each teacher serving as a case explored the different teachers' methodologies for integrating CT into their lessons and the barriers and challenges they faced. Teachers (n=7) participated in a week-long summer workshop that included a basic introduction to computing and CT, modeling computing tools, and time for teachers to practice using the tools. They also attended a follow-up workshop held during winter break. The teachers implemented computing into the content areas they already taught, including library/media, technology, art, 3rd grade, enrichment, 4th/5th grade, and 2nd grade. Observational data and field notes were collected when the teachers implemented computing into their classrooms. Results indicated that the teachers were generally willing to learn about computing and integrate computing into their instruction. Five of the seven teachers used whole-class instruction when integrating computing into the classroom. The teachers faced six major barriers when integrating computing:

- access to technology,
- access to expert support in the classroom,
- computing access issues due to poverty and disability,
- limited instruction time,
- lack of students' computing experience, and
- lack of classroom space.

Students from low-income families and those with disabilities required additional support from teachers. Some students who struggled academically found success with computing and were able to help other students and take on a leadership role. Computing lessons also encouraged student collaboration and minimized the role of the teacher as an expert.

Hladik investigated how to teach teachers to teach CT [45]. Four scaffolded CT activities were developed and included Language Arts, Physical Education, Music, and Art integrated into four different classrooms. The first activity was an unplugged activity for grades 1-3, discussing what a computer consists of, including smartphones, laptops, robotics, and computer systems in cars, and further discussing sequencing, conditional statements, and debugging. Students were then given materials to create a maze and asked to write instructions for an animal figurine to complete the maze. The second activity focused on sequencing and loops and required students to write down instructions for a popular dance. Once instructions were placed in the correct order, students performed the dance. The third activity integrated Language Arts with Scratch Stories for students to develop their own creative stories, including programming the plot and designing the scenes, characters, and actions. In the final activity, students used shapes, color, movement, and loops to design their own original art piece using Processing's pixel grid system. This activity used sequences, loops, conditionals, events, and variables. Findings of the study include an increase in students perceiving that coding is not ”hard”, that coding is creative, and that coding can be used to design ”cool things”. Increases in student perceptions were also found in their belief that they can use coding to learn other subjects and that coding teaches problem solving skills.

5 Step 5: Interpreting the Findings

While we recognize that K-5 is a large grade band, we would have liked to analyze the results in the context of the grade bands K-2 (typically pre- and early-readers) and 3-5. Since the results were more heavily aligned with 3-5, we added a note about these in each.

5.1 Perspectives

With respect to how to integrate CS into other subjects, both curricular co-design and elementary school CS education barriers exist that prohibit CS integration into schools [4]. Further, limited instructional time, budget, and background knowledge that elementary school teachers have with respect to CS have all been found to be additional barriers [4].

5.2 Subject Synthesis

5.2.1 Math

Across the articles we reviewed, the subject that was most commonly used to integrate CS was mathematics. This is no surprise, since mathematics and CS have overlapping concepts such as problem-solving, abstraction, and logic. The studies raised several positive outcomes related to students and teachers:

- Increase in student engagement [21]
- Increase in student learning [21]
- Improvement in students' sequential thinking [26]
- Increase in student motivation to learn mathematics [25]
- Improvement in students' ability to decompose number problems [26]
- Teachers' confidence in teaching mathematics had significantly increased through training [15, 25, 28]

The studies indicated that designing flexible, integrated curriculum enables teachers to adapt the curriculum as needed [24] and teaching analog programming prior to robot or block-based programming is beneficial [25].

Despite their similarities, however, the integration of CS into mathematics still has its challenges, including general challenges similar to other subjects: limited time for integration, limited training to teach integrated CS, limited support for teachers teaching integrated CS, and the perception of hindering the learning outcomes for the original subject curriculum [21–23, 28]. Additional challenges raised that were more specific to mathematics included:

- Students had difficulty decomposing the math-integrated word problem [26].
- Students struggled with constructing and recognizing repeat instructions in word problems [26].
- Students did not gain an understanding in storing data, articulating how to use variables, and control flow [27].

Teachers expressed beliefs that the mathematics curriculum needs to change to be able to integrate CS into the curriculum [24]. One teacher also noted that the perception that mathematics already has a reputation for being "hard", adding CS into the curriculum would make students perceive the class as even more difficult [22].

5.2.2 Science

We found that in the two studies where CS was integrated into science, it was integrated through biology. In these studies, CS was implemented into the many cycles seen within biology such as cell cycles, life cycles, or reproduction cycles of plants. Commonalities between the studies include students showing high engagement with the lessons, helping students with problem-solving, critical thinking, and CT [30, 31]. Engagement was most often measured by students' attention, excitement, and time trying to improve their programs [31].

The studies found high student engagement and increased student knowledge [30, 31]. However, researchers have noted that CS integration into K-5 biology instruction is not without its challenges. Challenges teachers faced while integrating CS into biology were that they were not confident in their understanding of CS, there were time constraints on lessons and there was also a lack of students' prior experience and availability to resources.

5.2.3 Engineering

Integration of CS into engineering most often took the form of robotics. Many of the studies included educational robotics kits such as the KIWI robotics kit, WeDo 2.0 kit, M-bot, and Ozobot. [32–34]. The kits kept students engaged and excited about learning [32, 33]. Additional positive outcomes included:

- Robots are powerful tools to keep kids engaged with CS concepts and practices [25]
- Robots led to student collaboration [25, 35]
- Robots led to increased learning [32]
- Increased motivation and enthusiasm [32]
- Increased motivation for mathematics [25]

Integrating robotics into CS faces many of the same challenges other subjects face such as limited integration time and a lack of teacher confidence in the CS. An obstacle that was more commonly reported within engineering integration was a lack of resources [6, 32–34].

5.2.4 Language Arts

Integration of CS into Language Arts was quite fluid due to the sequencing of storytelling and programming. Various studies have indicated that integrating CS into language arts can be done with positive learning outcomes. Research has shown that teaching CS through storytelling or as another language helps introduce CS effectively and efficiently for student understanding. A few studies used tasks to direct students through storytelling and CS concepts that allowed the students to build on their skills and end up with an original narrative that they had programmed [37, 38]. The students also showed high engagement through completing their tasks, excitement and attentive listening [36, 38, 46].

5.2.5 Multi-Subject

Due to the wide range of articles with multi-subject integration, the commonalities focus on the need for proper professional training and limited integration time. There were also mixed results on if the multi-subject integration of CS were successful at improving student knowledge of CS and the other subjects [8, 40].

Research into multi-subject integration of CS faces many of the same challenges as single-subject integration of CS. Teachers still require PD to be able to understand and feel comfortable teaching CT and CS. Mixed results were found when determining if multi-subject integration is a technique for eliminating the issue of not having enough time for integrating CS [8, 41, 44]. Some studies have found that students improve their knowledge of all subjects while some studies show knowledge gain in one subject. Despite all of the common barriers to CS integration, multi-subject integration tends to use technology toys and tools more often than single-subject integration. Technology toys and tools used in integrating CS have been shown to keep students engaged in learning activities that develop their problem-solving, reflection, and collaboration skills [42].

5.2.6 Overall Synthesis

Our review investigated a set of papers that studied CS integration into numerous subject areas in K-5. Our results, however, indicated that there remains a dearth of research on CS integrated into all subjects, but more so with respect to subjects such as fine arts, physical education, history, and social studies.

The research in our literature review indicated some common themes, including that introducing students to CS via integration into other subjects can be beneficial to students':

- Problem-solving
- Logical thinking,
- Collaboration skills, and
- Engagement and excitement.

The use of CS educational tools (e.g., Scratch, robotics toolkits) also can lead to increased student engagement and excitement.

Many of the studies also reiterated the challenges and barriers to integrating CS which included limited lesson time, limited PD training, limited resources, and students' prior experience with CS.

6 Conclusion

This systematic literature review considers the different papers found when considering the results from 900 papers on K-5 computer science education. Articles reviewed for this study show some overlap, but are primarily distinct, making it difficult to draw significant conclusions. However, some emerging trends indicate that training teachers on how to integrate CS into subject areas while also providing supports during their implementation can be a promising first step towards successful student outcomes.

As the field of K-5 CS education grows, we expect that additional patterns will emerge, indicating promising practices for this age group. In the meantime, a significant amount of additional research is needed on how best to implement CS in to primary school classrooms.

Acknowledgements

This material is based upon work supported by Code.org. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Code.org.

References

[1] Code.org, CSTA, & ECEP Alliance. 2022 State of Computer Science Education, 2022.

[2] Carol L Fletcher, Bryan Cox, Leigh Ann DeLyser, and Amy J Ko. Applying cape to assess equitable cs education. In *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, pages 1–2. IEEE, 2021.

[3] Sapna Cheryan, Allison Master, and Andrew N Meltzoff. Cultural stereotypes as gatekeepers: Increasing girls' interest in computer science and engineering by diversifying stereotypes. *Frontiers in psychology*, page 49, 2015.

[4] Victor R Lee Jody Clarke-Midura and Jessica Shumway Mimi Recker. “design for co-design” in a computer science curriculum research-practice partnership. In *Proceedings of the 16th International Conference of the Learning Sciences-ICLS 2022*, 2022.

[5] Megean Garvin, Heather Killen, Jan Plane, and David Weintrop. Primary school teachers' conceptions of computational thinking. In *Proceedings of the 50th ACM technical symposium on computer science education*, pages 899–905, 2019.

[6] Ahmad Khanlari and Fatemeh Mansourkiaie. Using robotics for stem education in primary/elementary schools: Teachers' perceptions. In *2015 10th International Conference on Computer Science & Education (ICCSE)*, pages 3–7. IEEE, 2015.

[7] Ahmad Khanlari. Teachers' perceptions of the benefits and the challenges of integrating educational robots into primary/elementary curricula. *European Journal of Engineering Education*, 41(3):320–330, 2016.

[8] Jeanne Century, Kaitlyn A Ferris, and Huifang Zuo. Finding time for computer science in the elementary school day: a quasi-experimental study of a transdisciplinary problem-based learning approach. *International Journal of STEM Education*, 7(1):1–16, 2020.

[9] Bailey Braaten and Arnulfo Perez. Integrating stem and computer science in algebra: Teachers' computational thinking dispostions. In *2017 ASEE Annual Conference & Exposition*, 2017.

[10] Mahmoud Quweider, Juan Iglesias, and Amajd Zaim. Educational computer science fun projects for integrating multidisciplinary concepts of mathematics, science, and engineering. In *2007 Annual Conference & Exposition*, pages 12–573, 2007.

[11] Selma Deneme and Selen Ada. On applying the interdisciplinary approach in primary schools. *Procedia-Social and Behavioral Sciences*, 46:885–889, 2012.

[12] Khalid S Khan, Regina Kunz, Jos Kleijnen, and Gerd Antes. Five steps to conducting a systematic review. *Journal of the royal society of medicine*, 96(3):118–121, 2003.

[13] Kathryn Rich, Carla Strickland, and Diana Franklin. A literature review through the lens of computer science learning goals theorized and explored in research. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, pages 495–500, 2017.

[14] Nicol R Howard. Edtech leaders' beliefs: How are k-5 teachers supported with the integration of computer science in k-5 classrooms? *Technology, Knowledge and Learning*, 24(2):203–217, 2019.

[15] Aman Yadav, Christina Krist, Jon Good, and Elisa Nadire Caeli. Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. *Computer Science Education*, 28(4):371–400, 2018.

[16] Phil Sands, Aman Yadav, and Jon Good. Computational thinking in k-12: In-service teacher perceptions of computational thinking. *Computational thinking in the STEM disciplines: Foundations and research highlights*, pages 151–164, 2018.

[17] Peter Rich, Olga Belikov, Emily Yoshikawa, and McKay Perkins. Enablers and inhibitors to integrating computing and engineering lessons in elementary education. *Journal of Technology and Teacher Education*, 26 (3):437–469, 2018.

[18] Matthew J Page, Joanne E McKenzie, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, et al. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *Systematic reviews*, 10(1):1–11, 2021.

[19] Alberto Martín-Martín, Mike Thelwall, Enrique Orduna-Malea, and Emilio Delgado López-Cózar. Google scholar, microsoft academic, scopus, dimensions, web of science, and opencitations' coci: a multidisciplinary comparison of coverage via citations. *Scientometrics*, 126(1):871–906, 2021.

[20] U.S. White House. Computer Science for All, 2016. URL https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all.

[21] Carla Strickland, Kathryn M Rich, Donna Eatinger, Todd Lash, Andy Isaacs, Maya Israel, and Diana Franklin. Action fractions: The design and pilot of an integrated math+ cs elementary curriculum based on learning trajectories. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pages 1149–1155, 2021.

[22] Pia Niemelä, Tiina Partanen, Maarit Harsu, Leo Leppänen, and Petri Ihantola. Computational thinking as an emergent learning trajectory of mathematics. In *Proceedings of the 17th Koli Calling international conference on computing education research*, pages 70–79, 2017.

[23] Mariam Haspekian. Computer science in mathematics new curricula at primary school: New tools, new teaching practices. In *Proceedings of the 13th International Conference on Technology in Mathematics Teaching (ICTMT 13)*, pages 23–31, 2017.

[24] Květoslav Bártek. Cross-curricular links between informatics and mathematics in contemporary czech school reality. In *ICERI2017 Proceedings*, pages 4286–4289. IATED, 2017.

[25] Gashawa Ahmed, Jalal Nouri, LeChen Zhang, and Eva Norén. Didactic methods of integrating programming in mathematics in primary school: Findings from a swedish national project. In *Proceedings of the 51st ACM technical symposium on computer science education*, pages 261–267, 2020.

[26] Feiya Luo, Maya Israel, and Brian Gane. Elementary computational thinking instruction and assessment: A learning trajectory perspective. *ACM Transactions on Computing Education (TOCE)*, 22(2):1–26, 2022.

[27] Feiya Luo, Wei Yan, Ruohan Liu, and Maya Israel. Elementary students' understanding of variables in computational thinking-integrated instruction: A mixed methods study. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1*, pages 523–529, 2022.

[28] Theodore J Kopcha, Cheryl Y Wilson, and Dayae Yang. Improving teacher use of educational robotics to teach computer science in k-5 mathematics. In *Computational Thinking in PreK-5: Empirical Evidence for Integration and Future Directions*, pages 47–54. 2022.

[29] Kathryn M Rich, Diana Franklin, Carla Strickland, Andy Isaacs, and Donna Eatinger. A learning trajectory for variables based in computational thinking literature: Using levels of thinking to develop instruction. *Computer Science Education*, 32(2):213–234, 2022.

[30] Mehmet Celepkolu, Erin O'Halloran, and Kristy Elizabeth Boyer. Upper elementary and middle grade teachers' perceptions, concerns, and goals for integrating cs into classrooms. In *Proceedings of the 51st ACM technical symposium on computer science education*, pages 965–970, 2020.

[31] Feiya Luo, Pavlo D Antonenko, and E Christine Davis. Exploring the evolution of two girls' conceptions and practices in computational thinking in science. *Computers & Education*, 146:103759, 2020.

[32] José Manuel Sáez López, Rogelio Buceta Otero, and Sebastián De Lara García-Cervigón. Introducing robotics and block programming in elementary education. *RIED. Revista Iberoamericana de Educación a Distancia*, 24 (1):95–113, 2021.

[33] Christina Chalmers. Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction*, 17:93–100, 2018.

[34] Amanda Sullivan and Marina Umaschi Bers. Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, 26:3–20, 2016.

[35] Carlos Jorge Brigas and José Alberto Quitério Figueiredo. " the hour of the code": Computational thinking workshop in a primary school in guarda, portugal. *Research in Social Sciences and Technology*, 4(2):129–136, 2019.

[36] Marina Umaschi Bers. Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education*, 6(4):499–528, 2019.

[37] Robert Whyte, Shaaron Ainsworth, and Jane Medwell. Designing for integrated k-5 computing and literacy through story-making activities. In *Proceedings of the 2019 ACM Conference on International Computing Education Research*, pages 167–175, 2019.

[38] Robert Whyte, Shaaron Ainsworth, and Jane Medwell. Designing multimodal composition activities for integrated k-5 programming and storytelling. 2020.

[39] Florence Sullivan, Enrique Suárez, Emrah Pektas, and Lian Duan. Developing pedagogical practices that support disciplinary practices when integrating computer science into elementary school curriculum. 2020.

[40] Alison E Leonard, Nikeetha Dsouza, Sabarish V Babu, Shaundra B Daily, Sophie Jörg, Cynthia Waddell, Dhaval Parmar, Kara Gundersen, Jordan Gestring, and Kevin Boggs. Embodying and programming a constellation of multimodal literacy practices: Computational thinking, creative movement, biology, & virtual environment interactions. *Journal of Language and Literacy Education*, 11(2):64–93, 2015.

[41] Stephanie DC Playton, Rebecca Hite, and Paula K Leach. A case study of a researcher-practitioner partnership in teaching stem+ c to rural elementary students. *The Electronic Journal for Research in Science & Mathematics Education*, 25(2):30–46, 2021.

[42] Veronica J Lin. Computational thinking and technology toys. 2015.

[43] Elena Jurado, David Fonseca, Jorge Coderch, and Xavi Canaleta. Social steam learning at an early age with robotic platforms: A case study in four schools in spain. *Sensors*, 20(13):3698, 2020.

[44] Maya Israel, Jamie N Pearson, Tanya Tapia, Quentin M Wherfel, and George Reese. Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, 82:263–279, 2015.

[45] Stephanie Hladik. Development of a cdio framework for teacher training in computational thinking. Master's thesis, Graduate Studies, 2017.

[46] Ammar H Safar, Ali A Al-Jafar, and Zainab H Al-Yousefi. The effectiveness of using augmented reality apps in teaching the english alphabet to kindergarten children: A case study in the state of kuwait. *EURASIA Journal of Mathematics, Science and Technology Education*, 13(2):417–440, 2016.