Science Education International
36(4)_2025,480-489
https://doi.org/10.33828/sei.v36.i4.11

ORIGINAL ARTICLE

Computational Thinking and Deep Learning on Science Education Framework: A Systematic Review

Zainur Rasyid Ridlo[1]*, Silvi Putri Ayu Ningsih[1], Azza Liarista Anggraini[2]

[1]Department of Science Education, Faculty of Education, University of Jember, Jember, Indonesia, [2]Department of Education, Faculty of Education, Specialising in Digital Learning, Monash University, Australia

***Corresponding Author:** zainur.fkip@unej.ac.id

ABSTRACT

This research investigates the implementation and challenges of incorporating computational thinking (CT) into science education. This research aims to develop a comprehensive framework to improve computational thinking skills in science education through the implementation of a deep learning curriculum, also integrated with computer programming, which leads to enhancing analytical and problem-solving skills for students to tackle real-world problems. This research used a systematic literature review with the PRISMA technique. Data sources used in this research are indexed by the Crossref database and then analyzed using the VOSviewer program. By examining 133 articles included in total, the research identifies key factors that affect effective teaching in science classrooms to enhance computational thinking. Through a series of case studies and empirical analysis, the study highlights obstacles faced in this educational approach. Findings suggest that while deep learning and computer programming integrated into science classrooms can influence the improvement of computational thinking skills and students' understanding of scientific concepts and their application, challenges such as limited resources are addressed. The proposed framework offers practical strategies for policymakers and educators, especially science educators, in designing learning to overcome these challenges, aiming to prepare students for a technology-driven future better.

KEY WORDS: Computational Thinking, Computer Programming, Deep Learning, Learning Framework, Science Education

INTRODUCTION

The global educational landscape has been a notable shift over the late decade; these made researchers and policymakers alike direct attention towards the development of education. One of the key areas of focus has been the equalization of digitalization, which has significantly influenced the overall structure of education worldwide (Qureshi et al., 2021). The rapid advancement of digitalization has affected numerous factors, particularly the digitalization of technology in education (Bond et al., 2020). The digitalization of the late years has been driven by production systems that utilize computer technology and incorporate new technologies, including Artificial Intelligence, robotics, augmented reality, cloud technology, and deep learning, which are being used for huge production (Taj and Jhanjhi, 2022; Ronchi, 2019).

The impact of digitalization in the educational field is obviously unclear, especially in designing teaching and learning processes to encourage students' skills. Similarly, new technologies' role in determining skill demands and development of curriculum in the education system is not yet fully understood (Gonzalez-perez and Ramírez-Montoya, 2022; Ellahi et al., 2019). Nevertheless, both teachers and students are required to develop the skills necessary to be in line with the technological complexities of the 21st century, which have been termed "21st-century skills" (Teo et al., 2021). The fundamental objective of contemporary educational systems has undergone a significant transformation, with a new emphasis placed on the acquisition and utilisation of technological competencies. An individual will be deemed proficient in the use of new technology if they have acquired the requisite skills at the relevant time. The future of learning will be shaped by digital technologies that are accessible, cost-effective, and efficient (Khan and Qureshi, 2020). To support this shift, educational institutions and learning processes must prioritize comprehensive improvement, exceptional service systems, determination, specialized proficiency, and the ability to lead data and information resources. In addition, they must embrace constant transformation and development, align with excellence, refine their approach, and instill confidence in their students (Eglash et al., 2019).

The 21st century landscape is developed rapidly, and technology integration into education is more crucial than ever. The 21st century is characterized by the transformations of digital technologies, data analysis, and automation, equipping students with essential skills becomes imperative. Among these skills, computational thinking stands out as a foundational competency that fosters digitalization and prepares students for future challenges. Development of 21st century skills can be achieved through computational thinking implementation (Lye and Koh, 2014; Agbo et al., 2019). Seymour Papert declared the term namely "computational thinking" in 1980 and explained that computers might recognize and change the pattern of knowledge and enhance thinking capability (Papert, 1980). Papert further emphasized that all students need to think such as computers to build their learning and elaborate their thinking capability (Papert, 1996). Then, Jeannette Wing (2006) states her idea about Computational Thinking in the article as "*a universally applicable attitude and skill set everyone.would be eager to learn and use*" (Wing, 2006). Wing states that all children should be taught about Computational Thinking Skills, while teaching reading, writing, and arithmetic in their cognitive development. Wing further declared that computational thinking includes how to designing system to tackle the problem and find the solution and understanding how behaviour of human, by abstracting the basic concepts to computer science Wing stated that this Computational Thinking Skill is not programming but as abstract thinking like a computer "*is more than just being able to program a computer. It requires thinking at multiple levels of abstraction*".

Despite the clear advantages of integrating computational thinking and digital tools into science education, several challenges remain. Variability in students' familiarity with technology can hinder engagement, and educators did not receive enough training about computer and software programming. In this way, educators need to develop teaching competence to design, implement, and evaluate (So et al., 2024). To address these challenges, schools can provide training for educators and create supportive learning environments that gradually introduce students to computational thinking concepts. This study aims to make a research framework an effective ideal media to address challenges and connect all keys to enhance students' computational thinking in science education. Align with this, this study addresses the following research questions:

- RQ1: To what extent are keywords correlations useful for improving computational thinking?
- RQ2: What is an effective tool to address an effective framework?

LITERATURE REVIEW

Computational Thinking Skills

Computational thinking is a cognitive process that involves logical reasoning to understand problems and design effective procedures or systems. Through computational thinking, students are able to analyze situations, identify key issues, and develop appropriate solutions (Ridlo et al., 2022). In the 21st century, such problem-solving skills are essential for navigating rapid technological advancements and preparing students to adapt to the growing influence of artificial intelligence (AI) (Sun et al., 2021). As global developments continue to be driven by innovations in information technology, the education sector is challenged to design curricula that cultivate competencies aligned with these changes. Among these competencies, computational thinking plays a critical role in supporting students' ability to engage with technological and informational challenges (Angraini et al., 2023). This skill is typically developed through computer programming and algorithmic exercises and fosters important qualities such as abstract thinking, pattern recognition, logical reasoning, and effective problem-solving (Angeli and Giannakos, 2020). To enhance students' computational thinking, it is crucial to embed this skill as a core component across educational disciplines, especially in the science education field. Furthermore, computational thinking should be applied in various individual and collaborative learning activities, enabling students to solve problems both independently and in group settings (Li et al., 2020).

Computer Programming and Deep Learning Curriculum in Science Education

Computer programming has an important role in science education due to its capability to ease students' understand of science concepts, deepen and practically (Kovac et al., 2025; Hsieh and Lin, 2020). In science learning, students are often faced with complex natural phenomena. To help students solve problems alongside the demands of 21st century, computer programming offers solutions such as creating simulations, visualisations, and models of these phenomena (Vallejo et al., 2022). Through programming, students also learn how smart technologies such as machine learning and artificial intelligence (AI) work in real contexts, for example, to predict micro-weather in coffee farms or analyse environmental data. Thus, computer programming is not only a learning tool but also a means to foster higher-order thinking skills, collaboration, and digital and technological literacy, all of which are essential in a modern science curriculum. Furthermore, computer programming trains students to think computationally by its step-by-step running system, and students also gained the capability of abstraction, generalisation, decomposition, algorithm, and debugging as core of computational thinking skills while integrated with computer programming (Ridlo et al., 2022). These skills are highly relevant to 21st-century learning and support the deep learning curriculum that emphasises in-depth understanding of concepts, problem solving, and integration of technology and science (Yulianti et al, 2020). Implementation of a deep learning curriculum is seen as a transformative step in education, requiring collaboration between government, educational institutions, and industry to overcome challenges and capitalize on opportunities. The curriculum should encompass theory, practical skills, ethics, and real-world applications to prepare a workforce ready for the digital age (Wijaya, 2025).

METHODS

This article was conducted by the Systematic Literature Review (SLR) method with the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) technique. This method is a structured, clear, and comprehensive analysis of the literature through a process of identifying, evaluating, and collecting data from research conducted by other researchers. In a systematic literature review, researchers identify articles and conduct systematic analyses following steps that are defined in fair research procedures. The systematic literature review method allows researchers to systematically analyze and identify articles, following specific procedures at each stage of analysis or review. According to Moher et al. (2009) the PRISMA technique involves four steps. These are identification, screening, eligibility, and including. The steps are presented in Figure 1.

The data collection is focused on "computational thinking, deep learning and science education" subject and the research article have been published. To identify relevant publications, a structured search query was applied (Alharbi and Stevenson, 2020). The following search terms were used: "computational thinking," "deep learning," "science education." These terms were combined using Boolean operators: ("Computational thinking" AND "deep learning"), also ("science education" AND ("deep learning" OR "computational thinking")).

The search was limited to articles published between 2020 and 2025, with a maximum cap of 1,000 publications as set in Publish or Perish. The final dataset consists of articles indexed in Crossref within the specified period. Crossref database selected due to the largest scope of research articles, open foundational infrastructure, and gave an important foundation of service, namely DOI (Pentz, 2022).

Included articles with the PRISMA technique are the number of final article data obtained is (N = 133). The articles obtained certainly have eligibility and are not included in this study. This refers to the inclusion and exclusion criteria in selecting articles following the research focus. The article criteria are shown in Table 1.

The information collected from the research articles is saved in two formats there is (*.ris) for VOSviewer visualization analysis and (*.csv) for Microsoft Excel analysis. Following data collection, article data were screened to analyze whether or not certain components (such as year, title, subject area, and abstract) were completed. Moreover, data saved in (*.csv) format will be analyzed further, and after all the data are clear to discuss further.

The retrieved data were exported in two formats: (*.ris) for bibliometric visualization using VOSviewer and (*.csv) for descriptive analysis using Microsoft Excel. Before analysis, the dataset was screened to ensure completeness of essential components such as publication year, title, subject area, and abstract (Kirby, 2023).

Following data cleaning, bibliometric mapping was conducted using VOSviewer. Several types of bibliometric analyses were performed, including: Country contribution and collaboration analysis to map global research productivity in the field,

Table 1: Exclusion criteria

Criteria	Include	Exclude
Article focus	Computational thinking, deep learning, science education	Interventions other than the research focus
Year of publication	Last 5 years (2020-2025)	Before 2020
Methods	Qualitative, Quantitative, Research and Development (R&D)	-
Language	Indonesian and English	Other than Indonesian and English

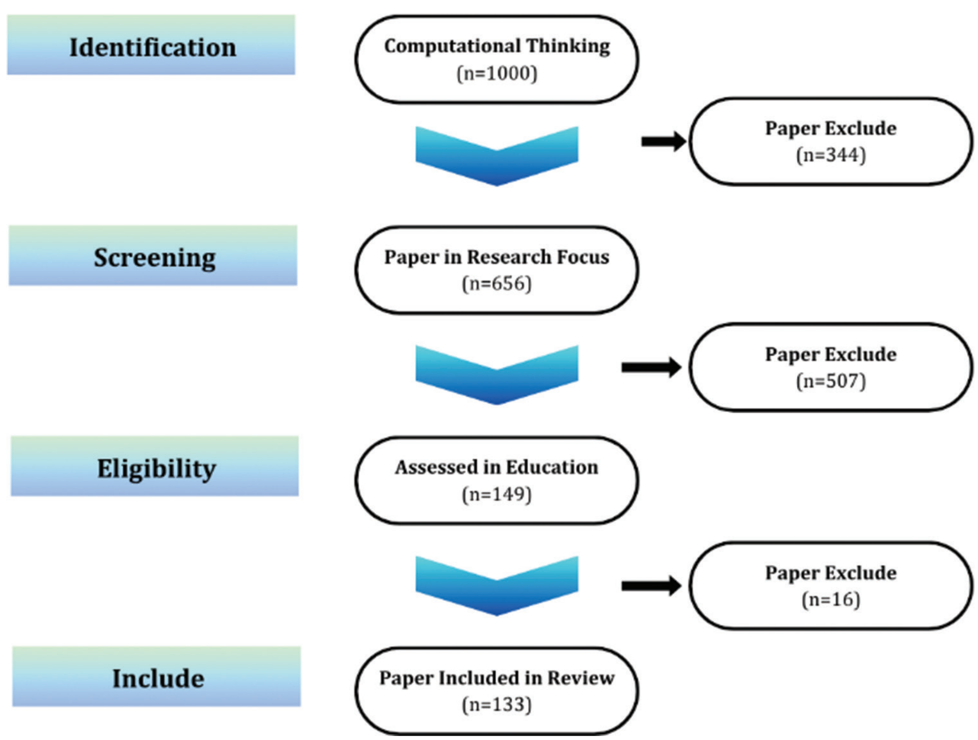

Figure 1: Research flowchart

Keyword co-occurrence analysis to identify major research themes and conceptual structures, Co-authorship analysis to examine patterns of research collaboration among authors. These visualizations provided a comprehensive overview of research trends, influential authors, and thematic developments related to computational thinking and deep learning in science education.

RESULTS

As a research method, this study employed bibliometric analysis. This method is used to collect articles as literature study data. Hereby, the results of the country distribution of articles retrieved. Moreover, another detail presented in several ways, there are: Document types through a range of year distribution, Top 10 rank article through cites per year (CPY) and cites per author (CPA), and VOSViewer analysis. Based on the country distribution, there are 133 articles represented as shown in Figure 2.

On the other hand, a total of 133 publications were published between 2020 and 2025 related to the keywords used. Of these articles, the data in Figure 2 show the distribution of document types, which includes book, journal articles, and proceeding articles such as thesis and dissertation. Figure 3 shows that journal articles are the most documents selected during the last five years, it is represented that researchers had concerns about publication in keywords related field.

After the data are obtained from publish or perish and downloaded in the format (*.csv), the data are then analyzed by sorting them based on the highest CPY and CPA. Here are

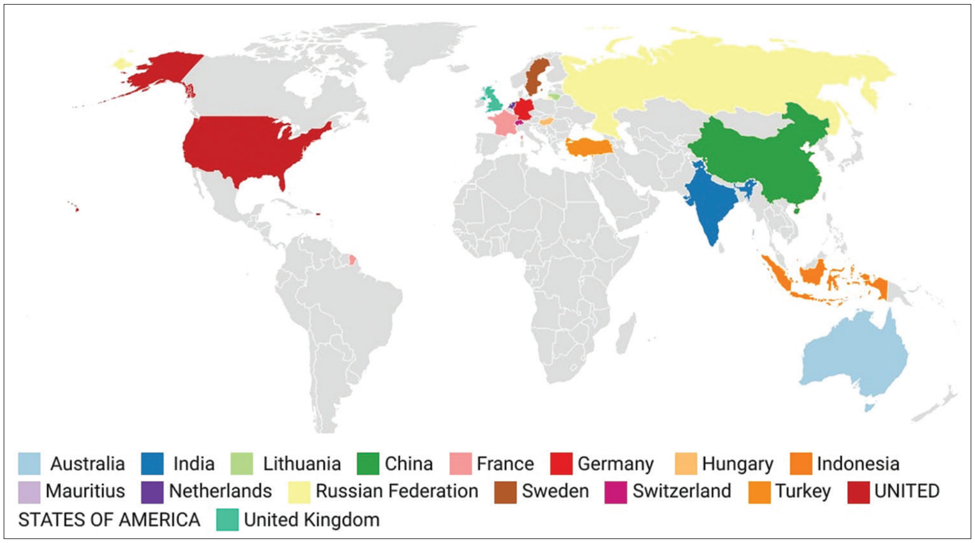

Figure 2: Country distribution of articles publishers

Table 2: Top 10 rank of articles the most cited

Title	Year	Source	CPY	CPA
Design and evaluation of a deep learning recommendation based augmented reality system for teaching programming and computational thinking	2020	IEEE Access	31.00	62
Computational thinking in early childhood education: The impact of programming a tangible robot on developing debugging knowledge	2023	Early Childhood Research Quarterly	6.00	3
Performance Comparison of TPU, GPU, CPU on Google Colaboratory Over Distributed Deep Learning	2021	2021 IEEE 14th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)	5.00	5
ScratchThAI: A conversation-based learning support framework for computational thinking development	2022	Education and Information Technologies	5.00	3
Computational Thinking Interventions in Higher Education: A Scoping Literature Review of Interventions Used to Teach Computational Thinking	2020	Koli Calling '20: Proceedings of the 20th Koli Calling International Conference on Computing Education Research	4.75	10
A framework for measuring abstraction as a sub-skill of computational thinking in block-based programming environments	2022	Education and Information Technologies	4.50	5
Thinking about computational thinking	2020	Journal of Digital Learning in Teacher Education	4.50	6
Communicating about computational thinking: understanding affordances of portfolios for assessing high school students' computational thinking and participation practices	2021	Computer Science Education	4.00	2
Computational thinking training and deep learning evaluation model construction based on scratch modular programming course	2023	Computational Intelligence and Neuroscience	4.00	2
Exploring Potentials and Challenges to Develop 21st Century Skills and Computational Thinking in K-12 Maker Education	2020	Frontiers in Education	4.00	4

CPY: Cites per year, CPA: Cites per author the top 10 articles that have the highest CPY, presented in Table 2. Further, each content will be analyzed and discuss in the next section.

Second format document from publish or perish and downloaded in the format (*.ris) the data is then represented with VOSviewer software to determine the relationship between the keywords used, namely “computational thinking-deep learning-science education.” VOSviewer provides three different forms of visualization namely network (showing the closeness of a relationship from each publication), overlay (showing the range of years the article was published), and density visualization (showing how often the topic is discussed with a cloud view) which is used to identify publications in the form of co-authorship analysis. Figure 4 shows network of keywords and its correlation each other (on the left), we also analyze based on density visualization to find out which keywords has a thick connection to each other. The density view is useful to get an important areas overview of a map quickly; the density view is presented on Figure 4 on the right.

In this study is focused on computational thinking and deep learning curriculum in science education, the scope of connectivity is minimized only just on articles selected to determine the correlation between computational thinking in science education. Figure 5 shows the closest connection between computational thinking in educational scope. Furthermore, Figure 6 shows the correlation with deep learning in science education is analyzed.

Articles selected also analyzed based on the co-authorship, it is present how each article was cited by each author. Figure 7 shows the co-authorship mapping of articles selected. Each dot/name represents an author. The size of the dot/name indicates

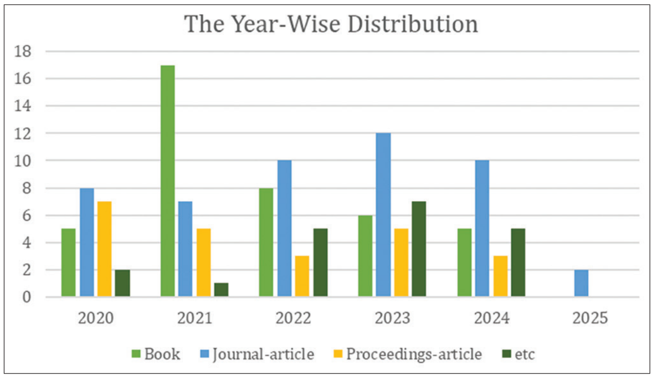

Figure 3: Documents types’ distribution during the past 5 years

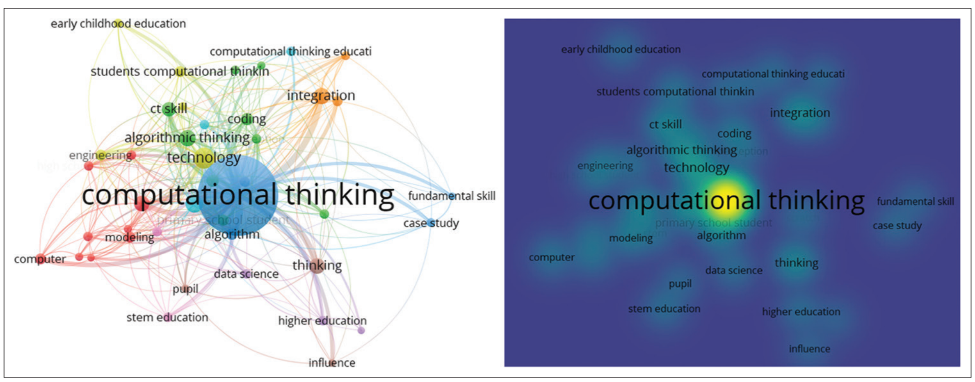

Figure 4: Representation of the connection of keywords (left) and density visualization of research area (right)

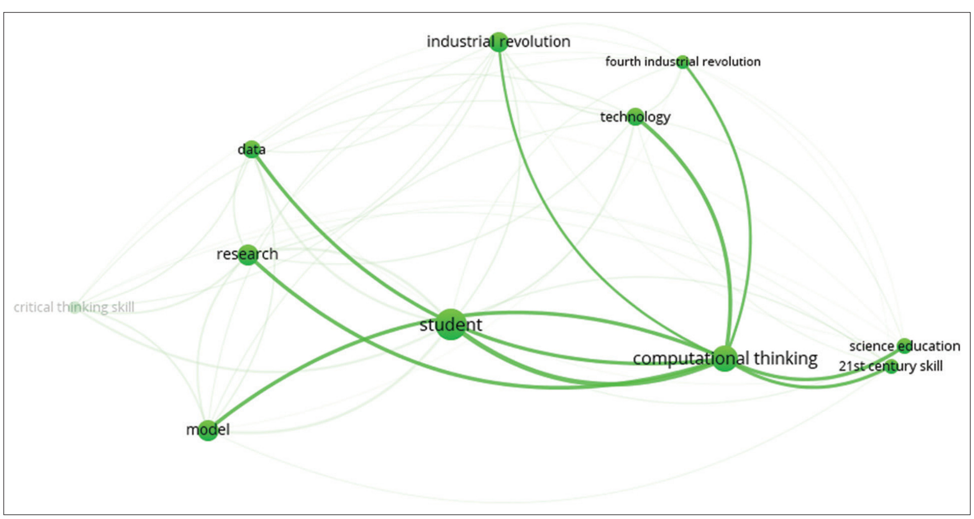

Figure 5: Computational thinking in science education

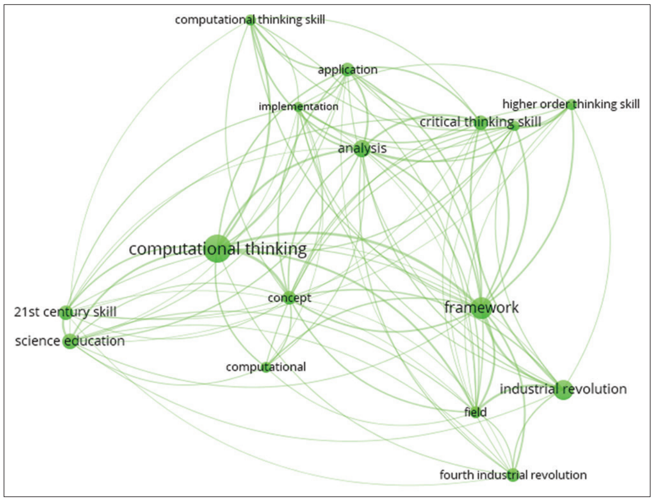

Figure 6: Computational thinking and deep learning in science education

how frequently the author is involved in co-publications, it means the larger the size, the more frequent the collaboration. Colors represent clusters or collaborative groups, where authors within the same color are closely connected or had the work together excessively. Lines connecting the dots indicate co-authorship or collaborative relationships between authors.

The co-authorship visualization generated by VOSviewer reveals a collaborative structure divided into several main clusters. *Chen, Yi* occupies the most central position with an extensive network of collaborations, followed by other prominent figures such as *Kong Sie-Cheung*, *O'Connor, Adrian*, and *David* and *Weintrop*. On the other hand, some authors appear isolated or have limited connections, indicating emerging or localized collaborations. Overall, the structure suggests the existence of strong collaboration centers and

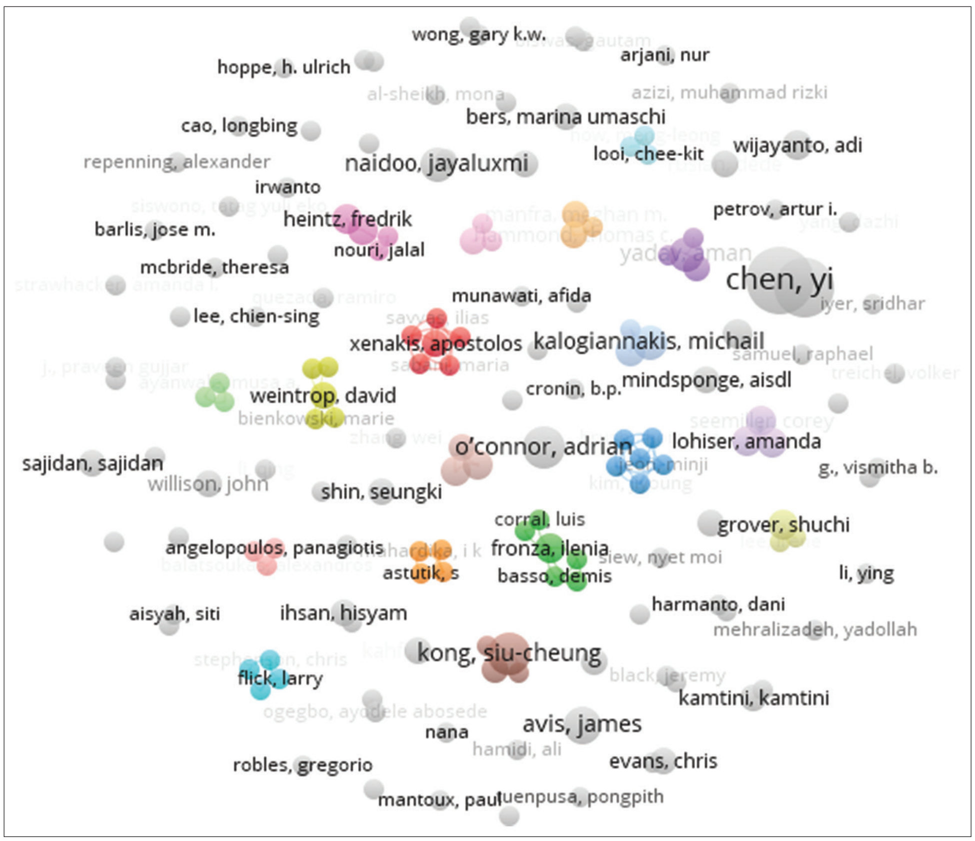

Figure 7: Co-authorship mapping visualization

Table 3: Indicators based on the articles review

Author	Year	Country	CT tools	Indicators
M C Hsieh	2022	Taiwan	Robotics	Abstraction, Algorithmic thinking, Decomposition, Pattern recognition
Yang Wang	2023	China	Robotics	Creativity, Algorithmic thinking, Collaboration skills, Critical thinking, Problem-solving ability
Z R Ridlo	2022	Indonesia	Computer Programming	Abstraction, generalization, decomposition, algorithm, debugging
Ge Yaoping	2022	China	Project Based with computer programming	Decomposition, Abstraction, Algorithm, Generalization, Evaluation, Collaborative ability
Li Shan	2023	China	Project Based	Creativity, Algorithmic Thinking, Collaboration and Communication, Critical Thinking, Problem Solving

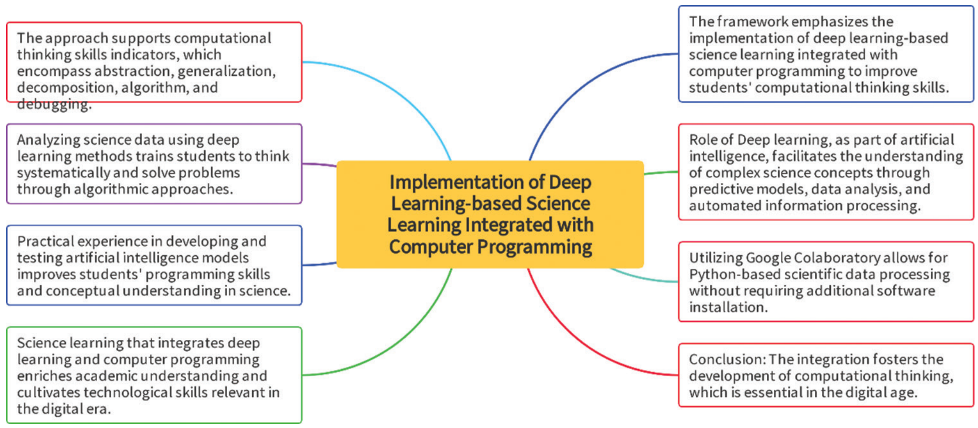

Figure 8: Research framework

Table 4: Indicators of computational thinking in computer programming integration

Indicator	Definition
Abstraction	The process of finding relevant things to solve a problem.
Generalization	The process of recognizing patterns to be able to formulate other problems
Decomposition	The process of designing the stages of a solution to solve a problem.
Algorithm	The process of designing step-by-step solutions to solve the problem.
Debugging	The process of finding errors in existing programs and fixing them

the potential for further expansion of scholarly networks in the future.

DISCUSSION AND CONCLUSION

RQ1: To What Extent Are Keywords Correlations to Improve Computational Thinking?

Computational thinking and computer programming on science education

Computational thinking skills are 21st century skills for solving problems (Saldo and Walag, 2020; Lockwood and Mooney, 2018). Computational thinking skill is a framework for thinking to tackle difficult problems into simpler and creates solvable problems, Computational thinking is not only owned by computers, but also must be owned by humans in their daily lives to process an existing problem (Wing, 2006). Since Wing brought the term Computational thinking to the educational research field, there has been a huge transformation in research examining the concept of Computational thinking skills (Lyon and Magana, 2020). Today, almost everyone, regardless of age, is expected to have computational thinking skills along with rapid transformation technology (Kalelioglu et al., 2016) due to benefits to adapt in artificial intelligence development era for students in the future (Sun et al., 2021). Initially, computational thinking was focused on and specialized skills possessed by computer scientists, engineers, mathematicians, and those from similar disciplines. Moreover, besides the computational thinking, programming ability is also a very utilized skill that can benefit students in a variety of disciplines respectively (Saldo and Walag, 2020; Lockwood and Mooney, 2018; Lyon and Magana, 2020, Ridlo et al., 2024). Nowadays, alongside with demands of digital age and digitalization had been evolved, computational thinking skills are utilized in a variety of disciplines, including statistics, biology, physics, and economics as well (Angraini et al, 2023; Ridlo et al., 2022), which means that the application of computational thinking skills should extend beyond computer science into other major subject especially science education. Hence, it is essential to integrate computational thinking skills into the primary and secondary school curriculum as an effort to prepare students who are able to compete in a global and digital age (Morze et al., 2022).

However, there are challenges while implementing and improving computational thinking in classroom would be address. First, for students need to cover the capability to solve complex problems and use strategies across domain-specific and domain-general contexts is regarded as important in science and computational thinking itself (Lai and Ellefson 2022). On the other hands, for teachers who felt difficult in applying new teaching methods and integrated technologies such as computer programming. Teachers need to acquire certain content knowledge in the classroom. Teachers also need to design challenging tasks and provide pedagogical support to help students solve computational problems that are connected to authentic contexts (So et al., 2024). To bridge the problems or obstacles faced by students and teachers, it means that they need stages or metrics as a step towards achieving these skills that can be used to lesson plan and/or Students' task, it can begin from addressing the indicators of computational thinking, then be utilized by teachers to develop their learning plans. Since computational thinking was declared by Seymour Papert and then further researched by Jeanette Wing, until research to date, various kinds of indicators have been used that are tailored to research in certain focus areas, involved robotics, block-based, and computer programming nowadays. According to research by Zhang et al., (2024) based on a review of 31 experiments, several indicators are shown in Table 3.

From these studies, it can be concluded that the indicators from several studies had problem-solving ability, which means that the final activities require students to solve problems. Furthermore, this research focuses on addressing challenges in improving computational thinking through the integration of computer programming, it requires indicators that are able to tackle errors in coding, as mentioned in research by Ge (2022), and debugging in research by Ridlo (2022). This is also reinforced by Li (2020) and Yadav et al (2018), who declared that debugging is an essential indicator of students' computational thinking by fixing the errors of the code vividly. A complete description of the indicators of computational thinking by Ridlo et al (2022) can be seen in Table 4.

The implementation of Computational thinking in education not only supports the development of individual skills but also contributes to students' ability to adapt to an ever-changing work environment in the future with evolving technology (Kalelioglu et al., 2018). Moreover, other challenges should be addressed, such as choosing the right tools that can be utilized as learning media to improve computational thinking. According to (Ridlo et al., 2022) to enhance computational thinking skills, it is important to have a programming basis and behavior of utilizing computer programs for students in solving problems. Since it was first introduced by Seymour Papert (Papert, 1980; Papert, 1996), the programming used is Logo, then computer programming has developed rapidly until now, integrated with artificial intelligence. The use of computer programming as a learning medium can have a massive impact on students' knowledge and problem-solving, because students can directly perform minds-on activities to solve problems with computational concepts (Li, 2020; Yadav et al., 2018). Nowadays, to make an effective framework it needs to integrate software or computer programming that beginner friendly, both should be easy to use pose for teachers and students, and one of the latest programming is Google Colaboratory, released by Google (Vallejo et al., 2022).

Deep learning and computational thinking on science education

Science education, in its learning, has a numeric aspect, especially in materials that involve numerical aspects, which can be a challenge for students. One clear example of this difficulty is when students are asked to solve problems related to a physics major, for example, to determine the acceleration of the motion of objects. The concept of acceleration involves an understanding of initial velocity, final velocity, time, as well as the application of the appropriate formula. Many students have difficulty in understanding these concepts and applying them in problem solving due to limited understanding of mathematical concepts, a lack of visualization in learning, and lack of varied exercises. Along with the development of technology, deep learning is one of the approaches that can help students overcome obstacles in numerical learning (Weintrop et al., 2016; Lin and Chen, 2020). Deep learning, which is also part of artificial intelligence, can analyze patterns in problem solving, provide more interactive solutions, and provide adaptive feedback to student errors (Chen and Lai, 2024). With a deep learning-based learning system, students can understand the concept of acceleration through interactive visual simulations, repetition of problems tailored to their level of understanding, and automatic analysis of the mistakes they made (Chen and Lai, 2024; Wang, 2024).

The application of deep learning in science learning allows students to not only memorize formulas but also understand the concepts behind them more deeply (Wang, 2024). In addition, the deep learning model can provide problem-solving recommendations based on the patterns of errors that students often make, so that they can learn in a more effective way. Hence, deep learning has the potential to be an innovative solution in overcoming the difficulties of science learning, especially in numerical materials such as the acceleration of object motion.

Deep learning is a branch of machine learning that aims to process complex data and find patterns that are not easily recognized by people (Wang, 2024). In the field of science education, deep learning can provide various benefits that support the strengthening of students' computational thinking at every implementation of the computational thinking indicators themselves (abstraction, generalization, decomposition, algorithm, and debugging) (Lmati and Khoulimi, 2023; Ridlo et al., 2022). Deep learning can be used in debugging by automatically analyzing error patterns in the code. Deep learning models can detect bugs, predict possible fixes, and provide optimal suggestions (Chen and Lai, 2024). With this approach, the debugging process becomes more efficient, reducing fault-finding time and improving software quality.

With the integration of deep learning in science learning, students not only understand scientific concepts deeper but are also involved in computational thinking that is essential for the digital era. Therefore, this approach needs to be continuously developed to prepare a generation that is ready to face future technological challenges. Computational thinking is an essential skill in science learning, as it involves systematic and logical problem-solving. Further, it can help many students face various difficulties in developing Computational thinking. Therefore, the application of deep learning in science learning can be a solution to overcome these barriers and obstacles.

RQ2: What is an Effective Tool to Address an Effective Framework?

Research framework to improve computational thinking in science education

This research framework focuses on improving computational thinking; it will be implemented through the implementation of deep learning-based science learning integrated with computer programming, such as Google Colaboratory. In today's digital era, learning approaches that combine natural science with artificial intelligence technology can provide a more interactive, exploratory, and data-driven learning experience. Deep learning allows learners to understand complex concepts in science through predictive models, data analysis, and automated information processing (Lin and Chen, 2020; Chen and Lai, 2024; Wang, 2024). In addition, the implementation of deep learning curriculum into the educational scope, especially in science, allows students to think deeply and analyze each learning activity systematically and logically (Wijaya, 2025).

Integration with Google Collaboratory provides the advantage of Python-based scientific data processing without the need for additional software installation, so students can directly perform simulations, visualizations, and programming-based experiments (Vallejo et al., 2022; Werth et al., 2022). This approach is in line with the computational thinking skills indicators, which include abstraction, generalization, decomposition, algorithm, and debugging (Ridlo et al., 2022). By analyzing science data using deep learning methods, students are trained to think systematically and solve problems with an algorithmic approach. In addition, practical experience in developing and testing artificial intelligence models can improve programming skills as well as conceptual understanding in science. Thus, science learning that integrates deep learning and computer programming not only enriches academic understanding, but also cultivates students with technological skills that are relevant in the digital era, namely Computational thinking. This framework is mapped in Figure 8.

Cultivating computational thinking and deep learning through Google Colab integration in 21st century learning is particularly crucial in science education and science learning, as the tools not only provide a platform to support interactive experiments and minds-on data analysis. Moreover, this way also equips students to learn how to design algorithms, tackle complex problems, and apply programming in a scientific context, improving their understanding of scientific concepts and analytical skills needed for scientific research and inquiry. This research found out the effective way to enhance computational thinking in students. Further research can implement this framework into wider range, either in science education or beyond to enhance students' computational thinking skills.

REFERENCES

Agbo, F.J., Oyelere, S.S., Suhonen, J., & Adewumi, S. (2019). A systematic review of computational thinking approach for programming education in higher education institutions. In: *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, pp. 1-10.

Alharbi, A., & Stevenson, M. (2020). Refining Boolean queries to identify relevant studies for systematic review updates. *Journal of the American Medical Informatics Association*, 27(11), 1658-1666.

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, 105, 106185.

Angraini, L.M., Yolanda, F., & Muhammad, I. (2023). Augmented reality: The improvement of computational thinking based on students' initial mathematical ability. International Journal of Instruction, 16(3), 1033-1054.

Bond, M., Buntins, K., Bedenlier, S., Zawacki-Richter, O., & Kerres, M. (2020). Mapping research in student engagement and educational technology in higher education: A systematic evidence map. *International Journal of Educational Technology in Higher Education*, 17(1), 2.

Chen, R., & Lai, Y. (2024). An empirical study on the cultivation of college students' computational thinking in the context of deep learning. *Creative Education*, 15(10), 2212-2223.

Eglash, R., Lachney, M., Babbitt, W., Bennett, A., Reinhardt, M., & Davis, J. (2019). Decolonizing education with Anishinaabe arcs: Generative STEM as a path to indigenous futurity. *Educational Technology Research and Development*, 68(3), 1569-1593.

Ellahi, R.M., Khan, M.U.A., & Shah, A. (2019). Redesigning curriculum in line with Industry 4.0. *Procedia Computer Science*, 151, 699-708.

Ge, Y. (2021). *Research on the Construction and Application of Project-Based Teaching Model for Information Technology Curriculum Based on Computational Thinking [Master's Thesis, Gannan Normal University]*.

González-Pérez, L.I., & Ramírez-Montoya, M.S. (2022). Components of education 4.0 in 21st century skills frameworks: Systematic review. *Sustainability*, 14(3), 1493.

Hsieh, H.L., & Lin, Y.T. (2020). Teaching Programming to Science Majors by Modelling. In: *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pp. 1367-1367.

Hsieh, M.C., Pan, H.C., Hsieh, S.W., Hsu, M.J., & Chou, S.W. (2022). Teaching the concept of computational thinking: A STEM-based program with tangible robots on project-based learning courses. *Frontiers in Psychology*, 12, 828568.

Kalelioglu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, 4(3), 583.

Khan, N., & Qureshi, M.I. (2020). A systematic literature review on online medical services in Malaysia. *International Journal of Online and Biomedical Engineering (iJOE)*, 16(6), 107-118.

Kirby, A. (2023). Exploratory bibliometrics: Using VOS viewer as a preliminary research tool. *Publications*, 11(1), 10.

Kovač, V.B., Nome, D.Ø., Jensen, A.R., & Skreland, L.L. (2025). The why, what and how of deep learning: Critical analysis and additional concerns. *Education Inquiry*, 16(2), 237-253.

Lai, R.P., & Ellefson, M.R. (2022). How multidimensional is computational thinking competency? A bi-factor model of the computational thinking challenge. *Journal of Educational Computing Research*, 61(2), 259-282.

Li, S. (2022). *Design and Practice of Project-Based Teaching for First-Year Programming Courses Aimed at Cultivating Computational Thinking [Master's Thesis, Ningxia University*].

Li, Y., Schoenfeld, A.H., diSessa, A.A., Graesser, A.C., Benson, L.C., English, L.D., & Duschl, R.A. (2020). Computational thinking is more about thinking than computing. *Journal for STEM Education Research*, 3(1), 1-18.

Lin, P.H., & Chen, S.Y. (2020). Design and evaluation of a deep learning recommendation based augmented reality system for teaching programming and computational thinking. *IEEE Access*, 8, 45689-45699.

Lmati, I., & Khoulimi, H. (2023). Towards improving mathematics learning using the deep learning models. In: *Proceedings of the 2023 IEEE International Conference on Advanced Data-Driven Analysis and Intelligent Systems (ADACIS)*, pp. 1-5.

Lockwood, J., & Mooney, A. (2018). Computational thinking in secondary education: Where does it fit? A systematic literary review. *International Journal of Computer Science Education in Schools*, 2(1), 41-60.

Lye, S.Y., & Koh, J.H.L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61.

Lyon, J.A., & Magana, A.J. (2020). Computational thinking in higher education: A review of the literature. *Computer Applications in Engineering Education*, 28(5), 1174-1189.

Moher, D., Liberati, A., Tetzlaff, J., & Altman, D.G. (2009). Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *PLoS Medicine*, 6(7), e1000097.

Morze, N.V., Barna, O.V., & Boiko, M.A. (2022). The importance of computational thinking training for primary school teachers. *Educational Dimensions*, 6, 22-39.

Papert, S. (1980). Computers for children. *In: Mindstorms: Children, computers, and powerful ideas*. United States: Basic Books, Inc., pp. 3-18.

Papert, S. (1996). An exploration in the space of mathematics educations.

International Journal of Computers for Mathematical Learning, 1(1), 95-123.

Pentz, E. (2022). Role of Crossref in journal publishing over the next decade. *Science Editing*, 9(1), 53-57.

Qureshi, M.I., Khan, N., Raza, H., Imran, A., & Ismail, F. (2021). Digital technologies in Education 4.0: Does it enhance the effectiveness of learning? A systematic literature review. *International Journal of Interactive Mobile Technologies (iJIM)*, 15(4), 31.

Ridlo, Z.R., Dafik, Waluyo, J., & Yushardi. (2024). Exploring the impact of makerspace-based learning materials on students' computational thinking skills: Using machine learning to address challenges in smart coffee agroforestry. *Edelweiss Applied Science and Technology*, 8(5), 52-73.

Ridlo, Z.R., Supeno, S., Wahyuni, S., Wicaksono, I., & Ulfa, E.M. (2022). Analysis of implementation project-based learning model of teaching integrated with computer programming in improving computational thinking skills in a classical mechanics course. *Jurnal Penelitian Pendidikan IPA*, 8(4), 2029-2035.

Ronchi, A.M. (2019). e-Learning: How teaching and training methods changed in the last 20 years. In: *e-Services*. Berlin: Springer, pp. 69-113.

Saldo, I.J.P., & Walag, A.M.P. (2020). Utilizing problem-based and project-based learning in developing students' communication and collaboration skills in physics. *American Journal of Educational Research*, 8(5), 232-237.

So, H.J., Kim, D., & Ryoo, D. (2024). Software education in South Korea for cultivating computational thinking: Opportunities and challenges. In: *Computational Thinking Curricula in K–12*. Cambridge: The MIT Press, pp. 165-186.

Sun, L., Hu, L., & Zhou, D. (2021). Improving 7th-graders' computational thinking skills through unplugged programming activities: A study on the influence of multiple factors. *Thinking Skills and Creativity*, 42, 100926.

Taj, I., & Jhanjhi, N. (2022). Towards industrial revolution 5.0 and explainable artificial intelligence: Challenges and opportunities. *International Journal of Computing and Digital Systems*, 12(1), 295-320.

Teo, T., Unwin, S., Scherer, R., & Gardiner, V. (2021). Initial teacher training for twenty-first century skills in the Fourth Industrial Revolution (IR 4.0): A scoping review. *Computers and Education*, 170, 104223.

Vallejo, W., Díaz-Uribe, C., & Fajardo, C. (2022). Google colab and virtual simulations: Practical e-learning tools to support the teaching of thermodynamics and to introduce coding to students. *ACS Omega*, 7(8), 7421-7429.

Wang, F.H. (2024). A feasible study of a deep learning model supporting human-machine collaborative learning of object-oriented programming. *IEEE Transactions on Learning Technologies*, 17, 413-427.

Wang, Y. (2023). The role of computer supported project-based learning in students' computational thinking and engagement in robotics courses. *Thinking Skills and Creativity*, 48, 101269.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2015). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127-147.

Werth, A., Oliver, K.A., West, C.G., & Lewandowski, H.J. (2022). Engagement in collaboration and teamwork using Google Colaboratory. In: *2022 Physics Education Research Conference Proceedings*, pp. 481-487.

Wijaya, M. (2025). Kurikulum Deep Learning di Indonesia; Sebuah Harapan Baru. *Jurnal Ilmiah Pendidikan Scholastic*, 9(1), 10-15.

Wing, J.M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

Yadav, A., Krist, C., Good, J., & Caeli, E.N. (2018). Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. *Computer Science Education*, 28(4), 371-400.

Yulianti, D., Wiyanto, W., Rusilowati, A., & Nugroho, S. (2020). The Development of High School Physics Teaching Material Based on STEM to Facilitate the Development of 21th Century Learning Skills. In: *Proceedings of the Proceedings of the 7th Mathematics, Science, and Computer Science Education International Seminar, MSCEIS 2019*, Bandung, West Java, Indonesia.

Zhang, W., Guan, Y., & Hu, Z. (2024). The efficacy of project-based learning in enhancing computational thinking among students: A meta-analysis of 31 experiments and quasi-experiments. *Education and Information Technologies*, 29(11), 14513-14545.