Journal of Advanced Research in Applied Sciences and Engineering Technology 51, Issue 2 (2025) 138-152

SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage: https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Current Research Trends of Scratch Block based Programming for K-12: A Systematic Review

Badruliman Batni[1], Syahrul Nizam Junaini[1,*], Jonathan Sidi[1], Wan Azani Mustafa[2], Zamhar Iswandono Awang Ismail[3]

[1] Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, 94300 Kota Samarahan, Sarawak, Malaysia
[2] School of Engineering Technology, Kampus Sg. Chuchuh, Universiti Malaysia Perlis, 02100 Padang Besar, Perlis, Malaysia
[3] Faculty of Computer Information Science, Higher Colleges of Technology, Abu Dhabi, United Arab Emirates

ARTICLE INFO

Article history:
Received 5 December 2023
Received in revised form 5 July 2024
Accepted 21 August 2024
Available online 19 September 2024

Keywords:
Teaching; education; learning; scratch; K-12

ABSTRACT

In the evolving landscape of K-12 education, the introduction of programming skills through block-based environments such as Scratch has become increasingly common. This approach meets the growing need for computer literacy among students, but its effectiveness in improving learning outcomes remains controversial. The aim of this paper is to systematically review and summarize current research on the use of Scratch as a teaching and learning tool in K-12 education. Using the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) methodology, this paper analyses 27 relevant articles to assess the impact and use of Scratch in educational settings. The review aims to identify the current trends, methods and foci in Scratch-related educational research. Our results show that 17 studies focus on the use of Scratch in teaching CS and ICT subjects, while 10 examine its application in other academic disciplines. The review shows generally positive results of Scratch programming in an educational context. However, it also highlights the need for more comprehensive empirical research. This includes conducting studies with larger and more diverse student samples over longer periods of time to gain a deeper understanding of how Scratch programming can effectively improve learning outcomes in K-12 education.

1. Introduction

In the current digital era, the need for computer literacy in K-12 education has led to innovative approaches to programming instruction, among which block-based programming environments have emerged as a central element. The importance of block-based programming, enabled by tools like of Scratch, Snap! and Blockly illustrates goes beyond mere programming knowledge; it represents a fundamental shift in the way young learners are introduced to computational thinking and problem solving [1-3]. This change is driven by the need to adapt educational practices to the evolving technology landscape and ensure that students are not only technology consumers but also capable

* *Corresponding author.*
E-mail address: syahruln@unimas.my (Syahrul Nizam Junaini)

https://doi.org/10.37934/araset.51.2.138152

creators [4]. The importance of studying block-based programming lies in its potential to provide an accessible and engaging entry point into the world of programming, especially for younger students who may find traditional text-based programming intimidating [5]. By focusing on this area, educators and researchers can gain insights into how these tools can be optimized for educational purposes to bridge the gap between theoretical knowledge and practical application in a digital world. Therefore, understanding the effectiveness, challenges, and pedagogical implications of block-based programming is crucial for designing future educational strategies.

Weintrop [6] discovered that block-based tools serve as an effective gateway for learners to transition into text-based programming languages. Such environments, using Scratch as an example, are particularly beneficial for beginners who may not be familiar with programming syntax. Users can visually explore commands and understand their functionality through an intuitive drag-and-drop interface. This method is particularly beneficial for people who find traditional typing challenging. In block-based programming, each instruction is represented as a visually clear image, allowing users to use natural language to interpret and explain how different instructions work. Despite these advantages, Weintrop's study also found some concerns among students about applying language learning in block-based programs, as they felt that these platforms may have overly complex rules and limited benefits. However, the increasing use of block-based programming beyond conventional computer science is challenging this view.

Focusing specifically on Scratch provides several additional benefits. As a block-based programming language, Scratch fosters an engaged online community where children can creatively create interactive media, including stories, games and animations. This aspect of Scratch encourages collaborative learning and peer feedback, improving the learning experience [1]. Furthermore, the widespread adoption of Scratch, more than any other platform of its kind [1,7], testifies to Scratch's effectiveness and attractiveness. The platform also promotes computational thinking in a playful and context-rich environment, making complex concepts more accessible and entertaining for young learners. Additionally, Scratch supports diverse learning styles and encourages experimentation and creativity, allowing students to learn at their own pace and in their own way [7]. This inclusivity and adaptability make Scratch a valuable tool in modern education, with benefits extending far beyond a basic introduction to programming.

Furthermore, Scratch is theoretically based on the CT framework by Brennan and Resnick [7]. This framework led to Scratch 2.0. Scratch also has certain benefits for young learners. With Scratch, kids can start learning computer programming earlier while working on personally meaningful projects like animated stories and games [1]. Scratch is a programming language that many children start using at an early age, and most users are between the ages of eight and eighteen. (https://scratch.mit.edu/statistics). The Scratch for Education tool has been analysed previously, but there is still a lack of understanding of its implications and contribution to educational practice as a whole.

There is a tendency for existing studies to focus on certain criteria without taking into account teaching methodology. It is important to note, however, that very few systematic literature reviews analyse Scratch's use as a teaching or learning tool or view Scratch as a combination of teaching or learning practices and methods. For example, Zhang and Nouri [8] provide a systematic review that provides evidence of computational thinking through programming through Scratch. While studies by Moreno-León and Robles [9] summarize recent research using Scratch in subjects other than computers and communication, as well as studies examining what students learn while coding. In this context, it is necessary to know the current state of research, trends and the best educational practice around programming with Scratch.

For this reason, the scope of this review is limited to works that address block-based Scratch programming at the elementary and secondary school (K-12) level. Programming in Scratch from elementary school through secondary school was chosen because interest in this programming environment often begins early and lasts. This work aims to identify: a) current distribution of studies on learning and teaching Scratch as a tool; b) the most popular research methods used in current research topics with Scratch in K-12 education; c) the popular focuses or topics related to programming with Scratch other than ICT; d) programming with Scratch used in K-12 lessons outside of the ICT subject. The basis for selecting the articles for this systematic literature search was the PRISMA statement (Preferred Reporting Items for Systematic Reviews and Meta-analyses) [10].

While the use of block-based programming tools such as Scratch in K-12 education has shown promising results, there remains a significant research gap in fully understanding their long-term impact on education. In particular, there is a need for systematic studies that examine how these tools contribute to the development of computational thinking, the transition from block-based to text-based programming, and their effectiveness in non-ICT subjects. This study aims to address these gaps by providing a comprehensive analysis of the current state of research on the use of Scratch in K-12 education. It aims to contribute to the field by providing insights into the most effective practices and methodologies in using Scratch as a teaching tool and understanding its broader impact on student learning trajectories in computer science and related disciplines.

2. Methodology

2.1 Research Question

The methodology used in the systematic literature review is presented here. The systematic literature review followed the guidelines proposed in the PRISMA statement (Preferred Reporting Items for Systematic Reviews and Meta-analysis) [10]. The PRISMA statement consists of a 27-point evidence-based checklist and a four-step flowchart that can be used to critically appraise published systematic reviews. PRISMA is not intended as a quality assessment tool. Instead, it is used to ensure that systematic literature reviews are documented in a consistent and understandable manner. The main purpose of the search was to create a map of the existing research and literature on Scratch in education. The search was then narrowed down to the use of Scratch in K-12 classrooms. This systematic literature review provides information on educational practices in block-based programming. The research questions (RQ) are:

RQ 1: What is the current distribution of studies on learning and teaching Scratch as a tool in K-12 classrooms?

RQ 2: What are the most popular research methods used in current research topics with Scratch in K-12 education?

RQ3: What are the popular focuses or topics related to programming with Scratch other than ICT in the K-12 education?

RQ4: How was programming with Scratch used in K-12 lessons outside of the ICT subject?

2.2 Conducted Search

Planning is essential for a consistent systematic literature review. The search string for digital libraries was constructed according to the definition of RQs. The protocol defines the following guidelines:

(a) Identify the main keywords of the research questions;
(b) Keyword related words and synonyms;
(c) Run tests on databases and review the results. Repeat keyword research by selecting related words and synonyms;
(d) Specify the keywords, related words, and synonyms used in the search string;
(e) Use OR Boolean to join synonyms and AND Boolean to join keywords;
(f) Run tests to evaluate the results. Rebuild the search string if necessary.

The research questions and study objective identified the main keywords. The main keywords were teaching, learning, education and Scratch. Based on the above guideline, the search string is: (learning OR teaching OR education AND scratch AND school OR K-12).

2.3 Screening of Papers for Inclusion and Exclusion

The search was conducted in Web of Science (Clarivate) and Scopus. These libraries were selected due to their better results during the pilot test and most publications related to computers and education. Originally, 734 studies were found in Scopus and 572 in Web of Science (Clarivate) without any parameter being selected. Studies were then selected based on inclusion and exclusion criteria. Inclusion criteria focused on articles directly addressing the use of Scratch as a tool in the K-12 classroom and published in academic journals from 2018 to 2022. Non-English language articles were excluded, as were studies that did not specifically focus on Scratch or were outside of the K-12 context. Conference papers, editorials and grey literature were also excluded to ensure the academic rigor of the review. The inclusion and exclusion criteria are listed in Table 1.

Table 1
Inclusion and exclusion criteria

Inclusion	Exclusion criteria
Articles that focus on the use of Scratch in teaching or learning in compulsory school (1st to 12th grade)	Articles that do not focus on the of Scratch in teaching or learning in compulsory school
Articles published in referenced or peer-reviewed articles and documented in English only	Articles in book chapter format, conferences and gray literature (opinion papers, technical reports, blogs, presentations, etc.);
Articles written in English language	Articles not written in English language
Empirical studies articles	Theoretical works such as frameworks and reviews articles
Articles released between 2018 and 2022	Articles published before 2018

A total of 989 documents were rejected, including conference papers, proceedings, dissertations, book chapters, editorials, magazine zines, reports, lecture notes and errata reviews published before 2018. After filtering out non-English studies and eliminating duplicate entries, the study sample was reduced to 878 articles. The selection was then made in two steps. First, titles and abstracts were read. Studies not related to Scratch in K-12 education practices were excluded. After filtering and eliminating duplicate entries, the study sample was reduced to 103 articles. In the second step, the introduction, methodology and conclusion of the studies selected in the first step were read and the inclusion and exclusion criteria were applied. All research papers were thoroughly reviewed at this point, resulting in the rejection of 76 papers. This systematic literature review included 27 articles that directly addressed studies addressing the use of Scratch as a tool in the classroom. Summary of the PRISMA process in Figure 1.

3. Results and Discussion

3.1 What is the Current Distribution of Studies on Learning and Teaching Scratch as a Tool in K-12 Classrooms?

3.1.1 Distribution by database

As shown in Figure 1, the number of works found in each database is distributed according to the database. The search string generated 1306 items by selection level 1. The abstracts of 103 studies were considered relevant. After applying the exclusion criteria, 27 studies were selected. Non-empirical studies and teacher-related studies were the main reasons for exclusion.

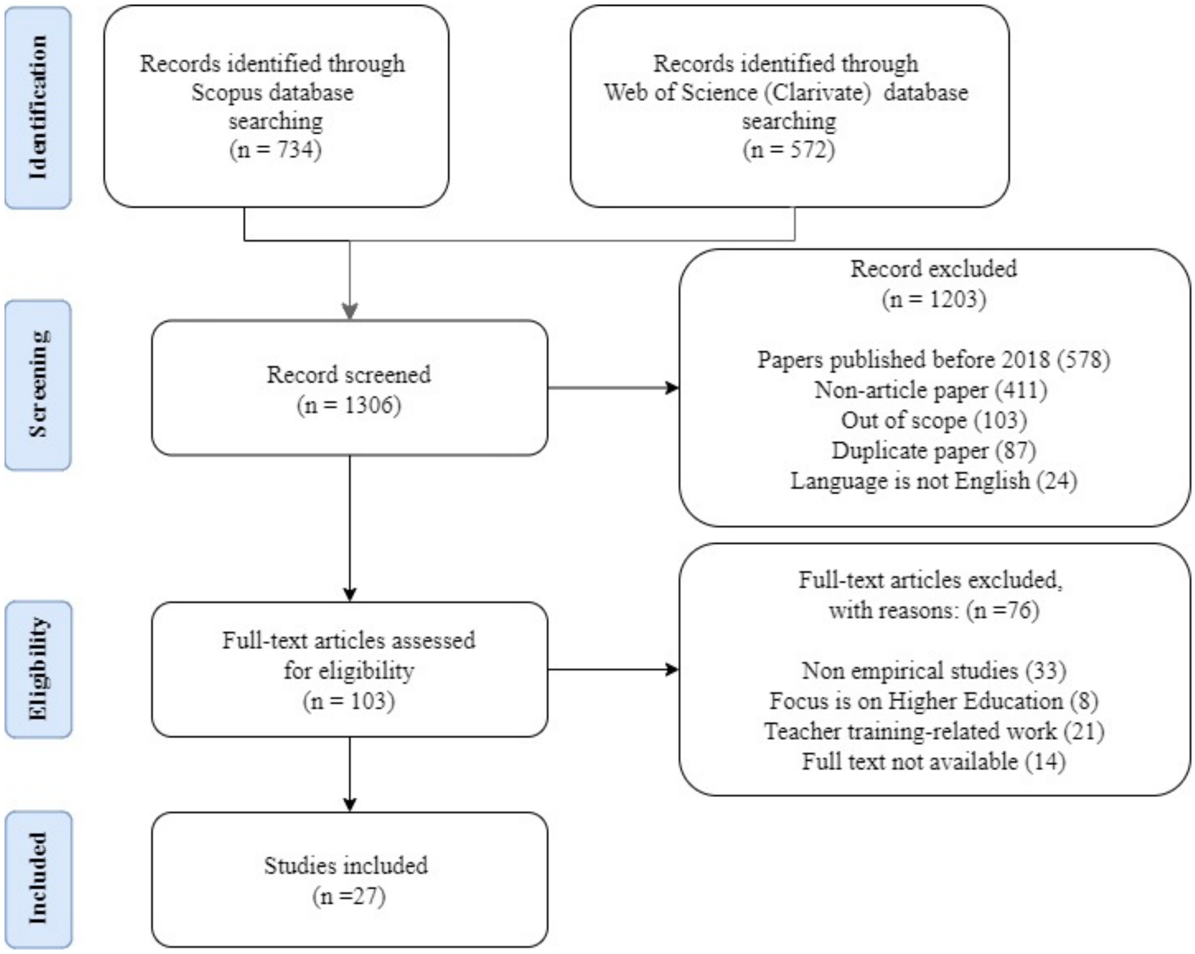

Fig. 1. Study selection chart [10]

3.1.2 Distribution by publication year

Figure 2 shows the number of studies published by year of publication. As can be seen from the graph, the number of publications peaks in the year 2020 with 33.3% of papers, followed by the year 2021 with 29.6% of papers. The distributions for the years 2022, 2019, and 2018 were 0.07%, 18.5%, and 14.8%, respectively. Based on these numbers, it is evident that the scientific community is still interested in the use of Scratch as a tool for teaching.

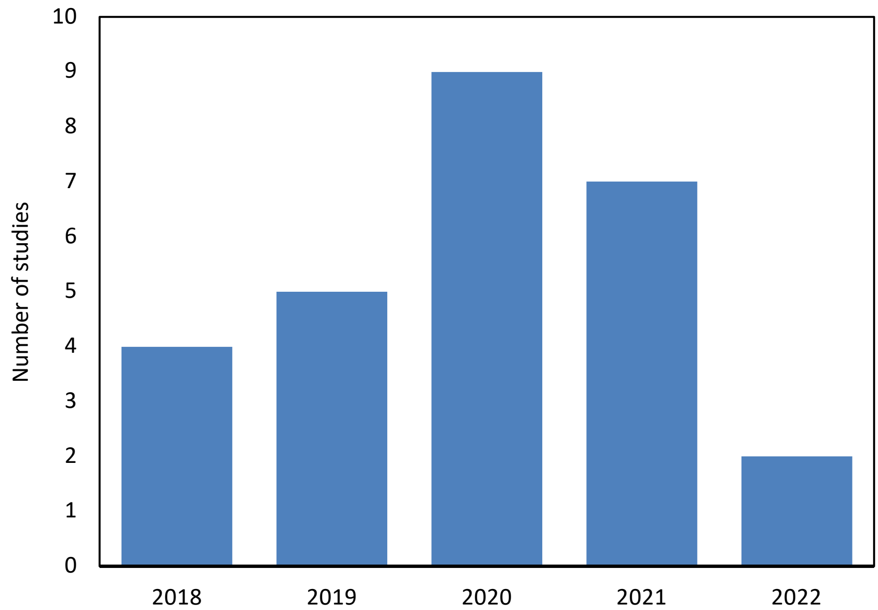

Fig. 2. The absolute number of studies by year of publication

3.1.3 Distribution by country

Turkey, Serbia and China are the countries with the most studies with four each. Spain has three studies, while the England and Greece each published two studies. Taiwan, Sweden, Korea, New Zealand, Scotland, Finland, Croatia and Brazil each have only one published study. This can be attributed to the fact that Turkey, Serbia and China are leading research in this field due to their recent serious actions in computer education, especially in the field of programming. The distribution of countries where research has been conducted in this area is shown in Figure 3. Note that the number of studies is higher than the total number of studies selected because some studies have authors from different countries.

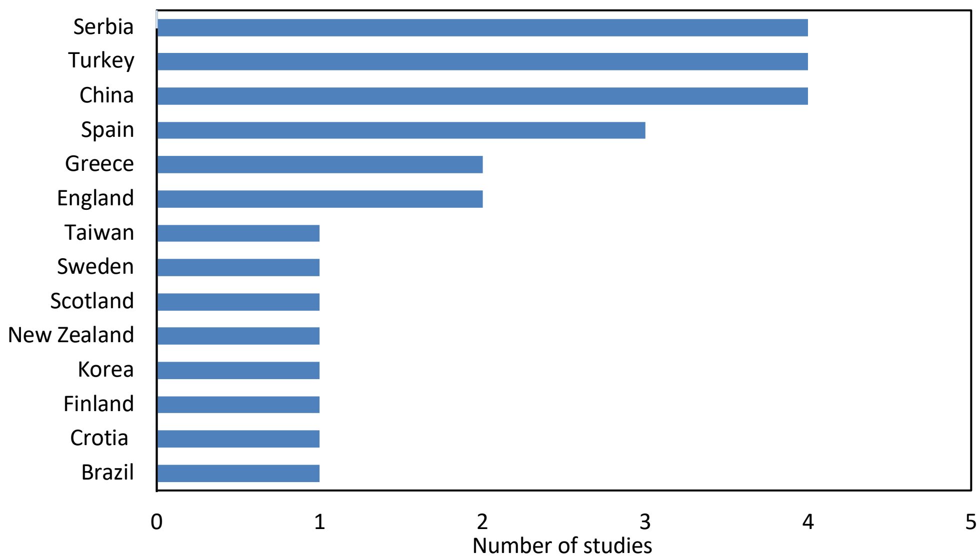

Fig. 3. Countries distribution

3.1.4 Distribution by sample size

As shown in Figure 4, sample sizes range from less than 20 to more than 100 participants. The Erümit [11] study has the largest sample size with 423 participants. This study used ANOVA and t-Test for quantitative analysis. There were only 5 participants in studies by Bowden [12] that used thematic analysis, a qualitative analysis method, to examine testimonies, case studies, interviews, and other text or image data. Unfortunately, these studies did not report the age of the users and were therefore excluded.

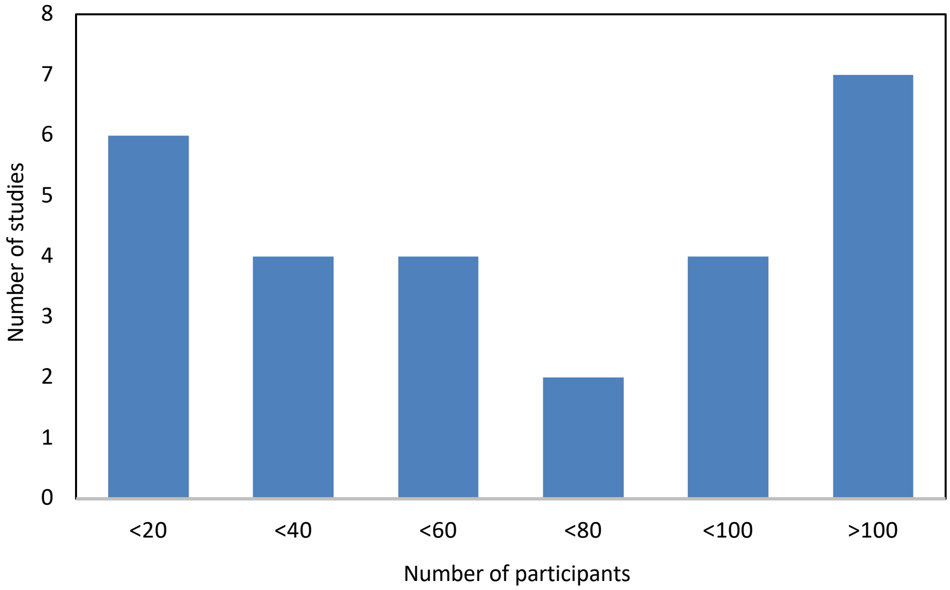

Fig. 4. Sample size of studies included

3.2 What are the Most Popular Research Methods Used in Current Research Topics with Scratch in K-12 Education?

3.2.1 Research method

A mixed-method research design with eleven studies proved to be the most common research method. This combines quantitative and qualitative research in a single project to provide an in-depth understanding of a research problem, accounting for 40.7% of all studies reviewed. Ten studies used a qualitative research design. This approach emphasized the use of unstructured rather than numeric information to gain a deep understanding of how Scratch is used in K-12 education. Finally, six studies adopted a quantitative research design focused on explaining and interpreting Scratch usage in K-12 formation. Table 2 shows the type of research method used.

Table 2
Article distribution based on the methodology

Research Method	Number of Studies
Mixed Method	11
Qualitative	10
Quantitative	6

3.2.2 Data collection method

Interview and survey were the main data collection methods, with 54.9% of studies investigating the use of Scratch. Interviews are two-way conversations between the interviewer and a participant to determine their perception. During the survey with open or closed questions to collect information. The results also showed that nine of the studies reviewed used tests as a tool, eight used Scratch code, while six studies used observation. These numbers are presented in Table 3.

Table 3
Distribution of Data Collection Methods

Method	Frequency
Scratch artefacts/Code	8
Observation	6
Pre/post test	9
Interview	14
Survey/Questionnaire	14

3.2.3 Types of statistical analysis methods

The statistical analysis included both descriptive and inferential methods. The t-test method was the most widely used at 6 times. It is often used to determine whether a process or treatment is having an impact on the population of interest or whether two groups differ. Figure 5 illustrates the statistical analysis methods used in the studies.

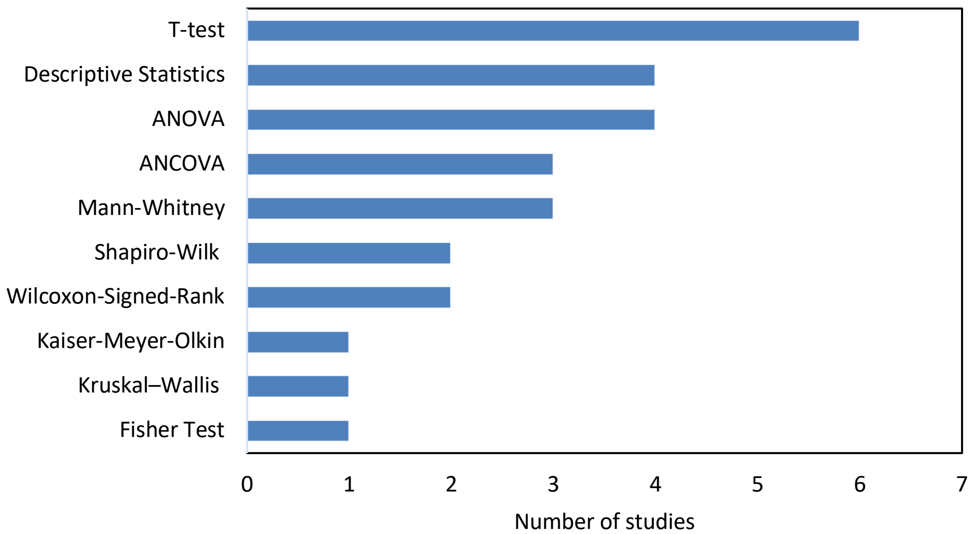

Fig. 5. Types of statistical analysis methods

3.3 What are the Popular Focuses or Topics Related to Programming with Scratch Other than ICT and CS in the K-12 Education?

Several studies in the academic literature describe the development of different types of applications that use Scratch to address different topics. Budinski *et al.,* [13] describe how students' program basic coding in Scratch using mathematical concepts from fractals and ancient paper folding skills from origami. Kabak and Korucu [14] demonstrate how students develop a game with Scratch in the context of English class through coding. After an experience introducing teachers to coding, it is possible to find articles that show how coding can benefit programmers from their subjects, such as art and music. From the summary of subjects, 17 studies use Scratch to teach CS and ICT subjects. Another focus is on subjects other than ICT and CS, where the pedagogical impact of using this resource in some way is measured and the usefulness of programming in K-12 is demonstrated. In the 27 papers selected for revision, ten use programmers to learn subjects other than ICT.

There is information about the articles, the students' grades, the subject or field in which programming was applied, and the environment. From the study, the most popular are articles addressing the impact of programming in STEM subjects. This is reflected in mathematics with five studies and sciences with one study. Programming with Scratch is also used in STEAM in 2 studies. STEAM combines STEM with ART. In addition, Scratch programming is also used in the introduction of artificial intelligence. Finally, there is a study that introduces Scratch to the topic but also to STEAM English as a second language context. The age of the study participants also varied greatly, ranging from 8 to 17 years. The majority of the studies, 9 out of 10, were conducted in schools and were therefore integrated programs in the curriculum, and only one study was developed as an extracurricular activity. Table 4 summarizes the topic of each paper.

Table 4
Subjects learned through coding with Scratch

Paper	Age	Subject	Environment
Chiang and Qin [15]	7th grade	Mathematics	School
Budinski *et al.,* [13]	17 years old	Mathematics	School
Calder [16]	9 - 10 years old	Mathematic	School
da Silva Pereira and Lopes [17]	12-15 years old	STEAM	School
Estevez *et al.,* [18]	16-17 years old	Artificial Intelligence (AI)	Workshop
Iskrenovic-Momcilovic [19]	3th grade	Mathematics	School
Kabak and Korucu [14]	Secondary School	English lessons	School
Ntourou *et al.,* [20]	5th grade	Science	School
Pou *et al.,* [21]	13–14 years old	STEAM	School
Rodríguez-Martínez *et al.,* [22]	6th grade	Mathematics	School

3.4 How was Programming with Scratch Used in K-12 Lessons Outside of the CS and ICT Subject?

A summary of the articles selected for this review is provided in Table 4. This table provides the following information for each article: the reference of the article, a general description of the study, the duration of the study, and the characteristics of the sample, including its age and size. As a result of the analysis of the selected papers, as shown in Table 5, only two studies have a sample size greater than 100, and the number of participants is less than 100 in all cases.

Table 5
Coding with Scratch to improve other subjects

Paper	Subject	Description	Duration	Sample	Prove Result
Chiang and Qin [15]	Mathematics	Assess the impacts of game-based construction learning, using Scratch, on students' multi-step equation-solving performance	10 weeks	89 seventh graders	Positive effects on equation-solving performance and attitudes toward learning mathematics through technology.
Budinski *et al.,* [13]	Mathematics	Mathematical and coding lessons based	15 weeks	15 Students 17 years old	The combination of origami and Scratch for creating fractals assisted them in

		on creative origami activities			developing mathematical and coding concepts.
Calder [16]	Mathematics	Using Scratch to facilitate mathematical thinking	2 weeks	14 students 9 - 10 years old	By using Scratch in the development of the digital learning object, mathematical thinking was encouraged and a better understanding of concepts was encouraged.
da Silva Pereira and Lopes [17]	STEAM	STEAM practices through the creation of electronic games using Scratch software	4 months	7 students 12-15 years old	Students made use of playfulness, creativity and collaboration to transform realities - proving to be an efficient platform for STEAM-related learning.
Estevez *et al.,* [18]	Artificial Intelligence	Introduction to Artificial Intelligence for High-School Students Using Scratch	2 weeks	37 students 16-17 years old	Achieve a positive change in their attitudes toward AI so as to become more realistic and aware of the potential benefits and pitfalls that AI may offer.
Iskrenovic-Momcilovic [19]	Mathematics	Improving geometry teaching with Scratch		106 third graders	Scratch provides an environment that has enabled mathematics to become more interesting and meaningful for students.
Kabak and Korucu [14]	English	Effect of students' developing their own digital games on their academic achievement and attitudes towards for English lessons	8 weeks	34 Secondary School students	The innovative method used in this study is a beneficial application that positively affect the academic achievement and attitudes of the students.
Ntourou *et al.,* [20]	Science	Impact of Arduino and Visual Programming in self-efficacy, motivation, computational thinking and students' perceptions on electricity	4 weeks	33 fifth graders	The effects on motivation and self-efficacy could not be demonstrated by data processing, while the effects on conceptual understanding electricity and CT were clear.
Pou *et al.,* [21]	STEAM	Computational thinking and educational robotics integrated into project-based learning	2 years	106 students 13–14 years old	Increase the performance and motivation of students and improve their skills gained and acquisition of knowledge of CT and the use of ER platforms.
Rodríguez-Martínez *et al.,* [22]	Mathematics	Computational thinking and mathematics using Scratch	8 weeks	47 sixth graders	Scratch can be used to develop both students' mathematical ideas and computational thinking.

The overview shows that programming with Scratch can improve learning in other subjects that are not related to ICT. Research by Chiang and Qin [15] shows that participants' attitudes toward learning mathematics and performance in solving equations were positively impacted. In Budinski *et al.,* [13]'s study, combining origami and Scratch to create fractals in math, helped students learn programming and math concepts. Scratch has made mathematics more interesting and meaningful for students [19]. In addition, Rodríguez-Martínez *et al.,* [22] affirm that Scratch can be used to develop mathematical ideas and computational thinking in students. Regarding science, based on a study by Ntourou *et al.,* [20], while the effect of motivation and self-efficacy of using elementary and science classes during data processing cannot be shown, it has a positive impact on science understanding Concepts and CT. However, in a study by Pou *et al.,* [21], it was found that using Scratch increased student performance and motivation, and enhanced their skills acquired in STEAM robot activities. This may be due to the relatively short time in the Ntourou *et al.,* [20] study, which lasted only 4 weeks compared to the Pou *et al.,* [21] study.

Scratch has also proven to be an effective learning and teaching platform through continued use. Through playful, creative and cooperative action, the students turned their ideas into reality in da Silva Pereira and Lopes [17] study on STEAM practices with Scratch over a period of four months. According to Kabak and Korucu [14] in the English classroom, developing digital games with Scratch positively affects students' academic performance and attitudes. Artificial intelligence was also introduced to Scratch. Using Scratch makes students more realistic and aware of the potential benefits and pitfalls of AI [18].

3.5 Scratch Programming in CS and ICT

The integration of Scratch programming into ICT and CS education is increasingly recognized for its potential to significantly improve computational thinking skills and overall academic performance in these areas. Jiang and Li [23] were instrumental in highlighting Scratch's role in creating interdisciplinary connections between fields such as mathematics and robotics. This integration not only provides students with hands-on programming experiences that effectively combine theoretical knowledge with practical, real-world applications, but also enriches their entire educational journey. However, the inclusion of Scratch in the curriculum brings with it complexities, especially in terms of ensuring that advanced students are sufficiently challenged. Fagerlund *et al.,* [24] emphasize the importance of a curriculum that is both deep and challenging, particularly for these students. Although Scratch's interactive and engaging nature makes it easily accessible and enjoyable for younger learners, caution against the danger of oversimplifying complex programming concepts and emphasize the need for careful consideration in curriculum design to ensure a robust learning experience [25].

In addition to its programming benefits, Scratch is also known for its ability to promote other important skills such as collaboration and creativity. As Allsop [26] points out, these skills are critical in modern education and are consistent with technology friendliness. However, he also draws attention to the challenges of assessing these skills within traditional educational settings, where they are often underrepresented. The effectiveness of Scratch varies across different learning environments, which, according to Çakiroğlu *et al.,* [27], calls for flexible teaching strategies. Game-based constructivist learning methods, while promising, require adaptation to different educational contexts to be effective, as argued by Hainey *et al.,* [28].

A significant research gap identified by Mladenović *et al.,* [29] argues that programming with Scratch helps students understand abstract concepts such as loops by providing concrete experiences without having to navigate complex syntax. Future courses need to examine how students move from block-based to text-based programming languages and assess their ability to transfer the skills learned. This gap is critical in understanding the long-term impact on students' ability to grasp complex programming languages. Bowden [12] looks at the broader educational implications of Scratch and highlights its critical role in fostering digital literacy and creativity skills that will become increasingly essential in a technology-driven future. However, this also highlights the need for a thorough examination of how digital tools such as Scratch influence students' perception and understanding of technology and highlights the importance of a nuanced approach when integrating these tools into educational contexts.

The application of robot-based learning practices in conjunction with the 6E educational model, as demonstrated by Hsiao *et al.,* [30], has shown significant improvements in motivation and performance among primary school students. This approach highlights the importance of integrating life experiences with learning content and characterizations as well as interdisciplinary knowledge and practical tasks to create a more holistic and engaging learning environment. While these results suggest a strong connection between innovative educational practices and improved learning outcomes, they also highlight the need for further research in diverse demographic groups to determine their broader applicability and effectiveness.

Furthermore, the different impacts of teaching methods that incorporate animation and game-based learning on computational thinking skills highlight the need for a differentiated and balanced approach to technical education [31]. Chun *et al.,* [32] have raised concerns that students may develop an over-reliance on visual cues in Scratch, which could hinder the development of abstract thinking skills necessary for understanding more advanced programming concepts.

In addition, collaborative learning strategies such as pair programming in Scratch, studied by Iskrenovic-Momcilovic [33], offer valuable interactive and social learning opportunities. However, they also present challenges in assessing individual learning in group settings. Wei *et al.,* [34] pointed out the benefits of partial pair programming in strengthening computational thinking skills. However, this result suggests that further research is needed on the manifestation of individual learning styles in collaborative contexts. Furthermore, Durak [31] observed different effects of different programming tools on student engagement. This research suggests that programming instruction using Scratch tends to be more effective in improving students' engagement and reflective thinking skills in problem solving compared to Alice. This highlights the importance of selecting the right tools in line with educational goals and students' specific needs.

These diverse perspectives underscore the importance of critically evaluating the use of Scratch in ICT and CS education. It's not just about integrating a new tool but about understanding how it aligns with educational objectives, the needs of diverse learners, and the evolving demands of a technology-centric educational landscape. As such, educators and curriculum developers must navigate these complexities with a keen understanding of both the potential and the limitations of tools like Scratch.

4. Conclusions

The comprehensive analysis of Scratch programming in K-12 education highlights its critical role in the contemporary educational landscape. Integrating Scratch into school curricula is a strategic response to the increasing demand for digital literacy in a technology-driven world. As a block-based programming tool, Scratch simplifies the coding process, making it accessible and engaging, especially for young learners and beginners. This approach goes a long way toward introducing basic programming concepts while minimizing syntax errors, promoting a basic understanding of coding from an early age.

However, the transition from Scratch's block-based environment to more advanced text-based programming languages represents a significant research gap. This transition is critical for developing deeper computational skills and understanding complex programming concepts. The research highlights the need for further investigation into how students are coping with this change and the long-term impact it has on their programming skills. Beyond programming skills, Scratch's impact extends to fostering creativity, problem-solving, and collaboration among students. These skills are critical in modern education, but present challenges when assessed within traditional educational frameworks. The studies reviewed indicate a need for adaptive assessment methods that can effectively capture these diverse skills.

Scratch's versatility is also evident in its application in various subjects beyond ICT, such as mathematics, science and languages. This interdisciplinary application highlights Scratch's ability to increase understanding and engagement across a wide range of topics, making learning more interactive and enjoyable. The research methods used in studying Scratch in K-12 education are varied and include a mix of quantitative and qualitative approaches. This suggests a comprehensive effort to understand the pedagogical implications of Scratch from multiple perspectives. However, there is a concern that students may rely too heavily on visual cues in Scratch, which could hinder the development of abstract thinking skills required for advanced programming.

In summary, Scratch programming represents a significant advancement in K-12 education, providing an innovative and accessible platform for introducing programming concepts. Its integration into education systems promises to provide students with essential digital and computational skills. However, it also highlights the need for ongoing research, particularly in understanding the transition to advanced programming and assessing broader skills promoted by Scratch. Educators and curriculum developers must manage this complexity and ensure that the use of Scratch aligns with educational goals and meets the diverse needs of learners. Therefore, Scratch is a key tool in the field of educational technology and requires a balanced and reflective approach to its implementation.

Acknowledgement
This research was supported by the Faculty of Computer Science and Information Technology, University Malaysia Sarawak.

References

[1] Maloney, John, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond. "The scratch programming language and environment." *ACM Transactions on Computing Education (TOCE)* 10, no. 4 (2010): 1-15. https://doi.org/10.1145/1868358.1868363
[2] Harvey, Brian, and Jens Mönig. "Bringing “no ceiling” to scratch: Can one language serve kids and computer scientists." *Constructionism 2010*, Paris. 2010.
[3] Fraser, Neil. "Ten things we've learned from Blockly." In *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*, pp. 49-50. IEEE, 2015. https://doi.org/10.1109/BLOCKS.2015.7369000
[4] Goode, Joanna, Gail Chapman, and Jane Margolis. "Beyond curriculum: The exploring computer science program." *ACM Inroads* 3, no. 2 (2012): 47-53. https://doi.org/10.1145/2189835.2189851
[5] Astrachan, Owen, and Amy Briggs. "The CS principles project." *ACM Inroads* 3, no. 2 (2012): 38-42. https://doi.org/10.1145/2189835.2189849
[6] Weintrop, David. "Block-based programming in computer science education." *Communications of the ACM* 62, no. 8 (2019): 22-25. https://doi.org/10.1145/3341221
[7] Brennan, Karen, and Mitchel Resnick. "New frameworks for studying and assessing the development of computational thinking." In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, Vancouver, Canada, vol. 1, p. 25. 2012.
[8] Zhang, LeChen, and Jalal Nouri. "A systematic review of learning computational thinking through Scratch in K-9." *Computers & Education* 141 (2019): 103607. https://doi.org/10.1016/j.compedu.2019.103607
[9] Moreno-León, Jesús, and Gregorio Robles. "Code to learn with Scratch? A systematic literature review." In *2016 IEEE Global Engineering Education Conference (EDUCON)*, pp. 150-156. IEEE, 2016. https://doi.org/10.1109/EDUCON.2016.7474546
[10] Moher, David, Alessandro Liberati, Jennifer Tetzlaff, Douglas G. Altman, and T. PRISMA Group*. "Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement." *Annals of Internal Medicine*

151, no. 4 (2009): 264-269. https://doi.org/10.7326/0003-4819-151-4-200908180-00135
[11] Erümit, Ali Kürşat. "Effects of different teaching approaches on programming skills." *Education and Information Technologies* 25, no. 2 (2020): 1013-1037. https://doi.org/10.1007/s10639-019-10010-8
[12] Bowden, Helen Melander. "Problem-solving in collaborative game design practices: epistemic stance, affect, and engagement." *Learning, Media and Technology* 44, no. 2 (2019): 124-143. https://doi.org/10.1080/17439884.2018.1563106
[13] Budinski, Natalija, Zsolt Lavicza, Kristof Fenyvesi, and Miroslav Novta. "Mathematical and Coding Lessons based on creative origami activities." *Open Education Studies* 1, no. 1 (2019): 220-227. https://doi.org/10.1515/edu-2019-0016
[14] Kabak, Kadir, and Agah Tuğrul Korucu. "The effect of students' developing their own digital games on their academic achievement and attitudes towards for English lessons." *Participatory Educational Research* 8, no. 2 (2021): 74-93. https://doi.org/10.17275/per.21.30.8.2
[15] Chiang, Feng-kuang, and Lian Qin. "A Pilot study to assess the impacts of game-based construction learning, using scratch, on students' multi-step equation-solving performance." *Interactive Learning Environments* 26, no. 6 (2018): 803-814. https://doi.org/10.1080/10494820.2017.1412990
[16] Calder, Nigel. "Using Scratch to facilitate mathematical thinking." *Waikato Journal of Education* 23, no. 2 (2018): 43-58. https://doi.org/10.15663/wje.v23i2.654
[17] da Silva Pereira, Elenise, and Leticia Azambuja Lopes. "Electronic game creation through scratch software: creative and collaborative learning fostering STEAM practices." *Acta Scientiae* 22, no. 3 (2020): 28-46. https://doi.org/10.17648/acta.scientiae.5535
[18] Estevez, Julian, Gorka Garate, and Manuel Graña. "Gentle introduction to artificial intelligence for high-school students using scratch." *IEEE Access* 7 (2019): 179027-179036. https://doi.org/10.1109/ACCESS.2019.2956136
[19] Iskrenovic-Momcilovic, Olivera. "Improving geometry teaching with scratch." *International Electronic Journal of Mathematics Education* 15, no. 2 (2020): em0582. https://doi.org/10.29333/iejme/7807
[20] Ntourou, Vassiliki, Michail Kalogiannakis, and Sarantos Psycharis. "A study of the impact of Arduino and Visual Programming In self-efficacy, motivation, computational thinking and 5th grade students' perceptions on Electricity." *Eurasia Journal of Mathematics, Science and Technology Education* 17, no. 5 (2021): em1960. https://doi.org/10.29333/ejmste/10842
[21] Pou, Albert Valls, Xavi Canaleta, and David Fonseca. "Computational thinking and educational robotics integrated into project-based learning." *Sensors* 22, no. 10 (2022): 3746. https://doi.org/10.3390/s22103746
[22] Rodríguez-Martínez, José Antonio, José Antonio González-Calero, and José Manuel Sáez-López. "Computational thinking and mathematics using Scratch: an experiment with sixth-grade students." *Interactive Learning Environments* 28, no. 3 (2020): 316-327. https://doi.org/10.1080/10494820.2019.1612448
[23] Jiang, Bo, and Zhixuan Li. "Effect of Scratch on computational thinking skills of Chinese primary school students." *Journal of Computers in Education* 8, no. 4 (2021): 505-525. https://doi.org/10.1007/s40692-021-00190-z
[24] Fagerlund, Janne, Päivi Häkkinen, Mikko Vesisenaho, and Jouni Viiri. "Computational thinking in programming with Scratch in primary schools: A systematic review." *Computer Applications in Engineering Education* 29, no. 1 (2021): 12-28. https://doi.org/10.1002/cae.22255
[25] P. Rose, Simon, M. P. Jacob Habgood, and Tim Jay. "Designing a programming game to improve children's procedural abstraction skills in scratch." *Journal of Educational Computing Research* 58, no. 7 (2020): 1372-1411. https://doi.org/10.1177/0735633120932871
[26] Allsop, Yasemin. "Assessing computational thinking process using a multiple evaluation approach." *International Journal of Child-Computer Interaction* 19 (2019): 30-55. https://doi.org/10.1016/j.ijcci.2018.10.004
[27] Çakiroğlu, Ünal, S. Sude Suiçmez, Yılmaz B. Kurtoğlu, Ayhan Sari, Suheda Yildiz, and Mücahit Öztürk. "Exploring perceived cognitive load in learning programming via Scratch." *Research in Learning Technology* 26 (2018). https://doi.org/10.25304/rlt.v26.1888
[28] Hainey, Thomas, Gavin Baxter, and Amanda Ford. "An evaluation of the introduction of games-based construction learning in upper primary education using a developed game codification scheme for scratch." *Journal of Applied Research in Higher Education* 12, no. 3 (2020): 377-402. https://doi.org/10.1108/JARHE-02-2018-0031
[29] Mladenović, Monika, Saša Mladenović, and Žana Žanko. "Impact of used programming language for K-12 students' understanding of the loop concept." *International Journal of Technology Enhanced Learning* 12, no. 1 (2020): 79-98. https://doi.org/10.1504/IJTEL.2020.103817
[30] Hsiao, Hsien-Sheng, Yi-Wei Lin, Kuen-Yi Lin, Chien-Yu Lin, Jheng-Han Chen, and Jyun-Chen Chen. "Using robot-based practices to develop an activity that incorporated the 6E model to improve elementary school students' learning performances." *Interactive Learning Environments* 30, no. 1 (2022): 85-99. https://doi.org/10.1080/10494820.2019.1636090
[31] Durak, Hatice Yildiz. "The effects of using different tools in programming teaching of secondary school students on

engagement, computational thinking and reflective thinking skills for problem solving." *Technology, Knowledge and Learning* 25, no. 1 (2020): 179-195. https://doi.org/10.1007/s10758-018-9391-y
[32] Chun, Seok-Ju, Yunju Jo, and Seungmee Lee. "The Effect of Programming Classes with Tangible Scratch Blocks on the Programming Interest of 6 th Grade Elementary School Students." *International Journal of Information and Education Technology* 11, no. 9 (2021): 405-409. https://doi.org/10.18178/ijiet.2021.11.9.1542
[33] Iskrenovic-Momcilovic, Olivera. "Pair programming with scratch." *Education and Information Technologies* 24 (2019): 2943-2952. https://doi.org/10.1007/s10639-019-09905-3
[34] Wei, Xuefeng, Lin Lin, Nanxi Meng, Wei Tan, and Siu-Cheung Kong. "The effectiveness of partial pair programming on elementary school students' computational thinking skills and self-efficacy." *Computers & Education* 160 (2021): 104023. https://doi.org/10.1016/j.compedu.2020.104023