World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(Research Article)

Check for updates

Enhancing computational thinking in children through logical puzzles: A machine learning approach for early coding education

Nishant Gadde [1,*], Avaneesh Mohapatra [2], Karan Mody [3], Siddhardh Manukonda [4], Navnit Vijay [5] and Rayan Idris [6]

[1] *Jordan High School, Katy, Texas, United States.*
[2] *West Forsyth High School, Cumming, Georgia, United States.*
[3] *Irvington High School, Fremont, California, United States.*
[4] *Jordan High School, Katy, Texas, United States.*
[5] *Acellus High School, Cumming, Georgia, United States.*
[6] *Jordan High School, Katy, Texas, United States.*

World Journal of Advanced Research and Reviews, 2024, 24(01), 877–882

Publication history: Received on 22 August 2024; revised on 05 October 2024; accepted on 08 October 2024

Article DOI: https://doi.org/10.30574/wjarr.2024.24.1.3024

Abstract

This recent growth in early education in coding evidences the rising need for the computing-thinking-skills development in children. Though most of the programming curricula are based upon syntax and other basic concepts of coding, computational thinking encompasses broader cognitive skills, such as problem-solving, abstraction, and logical reasoning. This research therefore intends to fill a gap in research and will investigate the ways pre-coding games employ logical puzzles as an enhancement in developing computational thinking skills in children aged 7-10 years and how these affect the acquisition of programming knowledge. In all, there will be two groups assigned: one will be exposed to logical-oriented puzzle training, Group A, and the other will be a control group, Group B, which receives only regular standard lessons in programming introduction. Pre- and post-measures of computational thinking and basic programming skills will be obtained from the participants. The machine learning algorithms, such as Random Forest for prediction in terms of improvement in subject performance in computational thinking and K-Means clustering to identify patterns in their development of skills, will be implemented in their pure or integrated forms to analyze data. Logistic Regression will be executed in order to model the odds of improved performance in programming given an intervention of exposure to puzzles. The research study will, therefore, apply these algorithms in comparing the performance of the two groups and reveal the aspects of computational thinking to which the greatest influence in a positive way through puzzle activities is exerted. It is expected that children exposed to logical puzzles may show significant improvements in both computational thinking and programming skills. These findings should be useful to educators in their effort to show how logical puzzles can be integrated into the curricula of early coding education to build foundational skills and thereby contribute to a refinement of best practices in STEM education.

Keywords: Random Forest; Machine Learning**;** Children; Logical puzzles

1 Introduction

Especially in the last few years, there has been an increased interest in early coding as a means to prepare students for a future where technology is becoming an inseparable part of life. Computational thinking is one of the prerequisites for coding: a cognitive process whereby one breaks down problems, recognizes patterns, abstracts solutions, and thinks logically. This has become an important skill in developing problem-solving strategies and adaptability in an ever-changing technological world. Most introductory coding curriculums are focused on syntax and basic notions of programming, though, leaving broader cognitive skills of computational thinking underdeveloped in young learners.

* Corresponding author: Nishant Gadde

Early exposure to logical reasoning activities, such as doing puzzles, has shown a positive impact on computational thinking and is said to better prepare students for learning to code. There is limited research regarding this area of work. The majority of the studies look into how coding education generates computational thinking; only a few examine whether pre-coding activities-logical puzzles-can be seen as an effective precursor to formal programming education.

This paper, therefore, tries to fill that research gap by exploring how the introduction of logical puzzles in early education influences the development of computational thinking skills in children aged 7-10 years. More precisely, it seeks to find out whether the exposure of students to puzzles like Sudoku and pattern recognition games enhances their capabilities for comprehending and applying basic concepts of programming. It seeks, through the use of ML algorithms on data analysis, to ascertain which particular aspects of computational thinking have been most positively influenced by activities, and whether students exposed to logical puzzles would perform better in programming tasks than those unexposed.

The findings from this study will be instrumental to educators and curriculum developers in giving a new dimension to how logical puzzles can be combined with early coding education to enhance problem-solving skills, deepen the understanding of computational thinking, and further develop the long-term programming outcomes.

2 Literature Review

Computational thinking, in the digital era, becomes an important competence, with special emphasis on early programming education. Wing identifies this computational thinking as the "thought processes involved in formulating problems and expressing their solutions in such a way that a computer-human or machine-can execute them," therefore referring to problem-solving, abstraction, and pattern recognition, part of not only programming skills but also from the wider scope of competencies in STEM. These skills let people think of problems in a way that computers can solve, making CT an underpinning to computer science and everyday problem-solving. Given these advantages, however, most introductory coding courses focus narrowly on syntax and the structures of programming languages at the expense of more cognitive processes associated with CT.

There is growing evidence of the development of computational thinking through well-designed instructional intervention in early education. Teaching children programming skills has been one of the current popular approaches to fostering CT, incorporating block-based programming environments, such as Scratch, into teaching the basics of coding. These environments offer students ways of interacting with code in a visual, intuitive manner that lessens the cognitive load normally associated with traditional programming syntax. Despite these advances, computational thinking via coding in and of itself is bounded, as syntax often acts as a sort of roadblock for young learners who may not have prior exposure to activities entailing logic and problem-solving.

Research in the area of cognitive development indicates that logical puzzles and problem-solving games enhance the basic competencies that underlie computational thinking. Games like Sudoku, pattern recognition games, and other games that require logical reasoning form the basis in children for critical thinking, abstraction, and identification of patterns, which are also intrinsic elements of computational thinking. These puzzles involve learners breaking down a complex problem into manageable parts, recognizing patterns, and creating a strategy to reach a solution-processes that closely parallel the stages of computational thinking.

Studies have also shown that engaging children in activities such as logical puzzles prior to coding can help build their cognitive skills for later application to coding tasks. For example, Brenner and Barnett (2016) found that students exposed to logic puzzles prior to coding outperformed students not given such an opportunity on early programming tasks. In a similar vein, Arfé et al. (2020) established that logical puzzles can be used as scaffolding in the enhancement of problem-solving skills among young learners, therefore preparing them for instructions in coding.

While there is significant evidence supporting the application of coding activities to develop computational thinking skills, there is a lack of research that has focused explicitly on how preparation with logical puzzles can enhance such skills before explicit coding instruction. Indeed, most current studies focus on the direct influence of coding on computational thinking, not considering other possible benefits coming from previously mentioned preparatory activities. Lye and Koh 2014 also further reinforce this thinking; however, their studies look at the needs of interventions required in computational thinking. Their studies are not deep enough to follow through on the possibilities regarding how activities that may be pursued will be non-coding.

The use of ML techniques is becoming more prevalent in education research in the analysis of data, deriving patterns that may, with conventional statistical methods, be missed. Different ML algorithms, such as Random Forest and K-

Means clustering, can be employed in this research to analyze the influence of logical puzzles on computational thinking. For instance, using a model with a random forest could come up with the prediction of the improvement of computational thinking skills in students, depending on exposure to puzzles. Common patterns in the development of skills may be found by performing K-Means clustering between different groups of students. Finally, combining these approaches will lead to a fine-grained outcome in detail with regard to the cognitive process lying under the surface of computational thinking and how these are influenced by logical puzzle activities.

3 Methodology

This research was to find out how the introduction of logical puzzles to these subjects could further enhance computational thinking skills in a child aged 7-10 from the non profit CodeWorld, and if such pre-coding games would have any positive implications for children when they learn programming. It also includes a description of designing the methodology for conducting research to collect data on the cognitive effect and assessment of the development in computational thinking ability using ML algorithms for finding important patterns and correlations.

Participants: The sample will include 100 participants, aged 7-10 years, who are complete beginners in the field of programming. They will be taken from elementary schools and, thus, will have no formal experience in coding or computer science to date, so the results would reflect the effect of logical puzzles as a pre-coding activity. Participants will be randomly allocated into two groups as follows:

Group A (Intervention Group): Students subjected to logical-oriented puzzles, such as Sudoku and pattern recognition games, integrated into the curriculum.

Group B: The control group receives the same lessons regarding programming but does not have any exposure to logical puzzles. Intervention: Regularly exposing the subjects in Group A to logical puzzles for 8 weeks, integrated into daily activities, is the intervention. These include problem-solving games that the students will have to solve. These focus on pattern recognition, abstraction, and logical reasoning in a work-based way of developing computational thinking skills. Group B will receive standard coding lessons without the integration of puzzle-based activities over the same duration. Both groups will then be introduced to fundamental programming concepts using block-based coding environments, Scratch for example, two weeks after the intervention. This allows the logical puzzle exposure to be analyzed on understanding and applying basic programming concepts.

3.1 Testing Tools

The development in computational thinking and programming skills will be measured by two forms of assessment. These include:

Pre-Test and Post-Test Computational Thinking Assessments: The study incorporates two standardized tests, one to be administered at the start and one at the close of the 8-week period. The main aspects to be tested in computational thinking are problem-solving, pattern recognition, abstraction, and logical reasoning.

Programming Task Assessments: Students will be asked to complete a set of simple programming tasks in block-based coding environments after initial programming lessons. Students' performance on those tasks will be measured for assessing their ability to understand and apply some basic programming concepts.

3.2 Machine Learning-Based Data Analysis

The outcomes of both series of computational thinking and programming tests will be evaluated using machine learning algorithms to investigate the influence of logical puzzles on developing their skills.

Random Forest: this ensemble learning algorithm will predict the students who will show the biggest improvement in computational thinking and programming due to exposure to logical puzzles. It will look into how different variables interact, like the completion rates of the puzzles, pre-test results, and even student characteristics, to define improvement patterns.

K-Means Clustering: this unsupervised learning algorithm will consequently group the students by test results and performance on programming tasks. Clustering students with similar profiles of cognition, the study points out those aspects of computational thinking which have been most influenced by logical puzzles. It will help in finding out whether any distinct learning paths begin to emerge between the students who were exposed to puzzles and those who were not.

Logistic Regression: In this model, logistic regression will determine the odds of students in Group A doing better as compared to Group B in both computational thinking and programming tests. Logistic regression will measure the relationship between exposure to puzzles and enhancement of cognitive and programming skills.

3.3 Statistical Analysis

Complementing the machine learning models, traditional statistical methods will also be conducted to confirm results:

The paired t-tests will help assess the results of pre- and post-tests individually in the two groups to ascertain whether large and statistically significant improvements in computational thinking occurred over the 8-week period. The post-test findings are to be tested using independent t-tests across the two conditions: Group A, exposed to puzzles, and Group B, which is to serve as the control group, to determine whether logical puzzles have a statistically significant effect on performance in both computational thinking and programming. Ethical Considerations:

The research will be carried out with complete ethics approval from an Institutional Review Board. Informed consent will be taken from the participants as well as their guardians before the study begins. The participant's data will be anonymized; the privacy of participants will be taken into consideration and ensured in the entire research process. There is no potential harm to the participants, and the puzzles and programming activities will be age-appropriate and engaging.

3.4 Expected Outcomes

The treatment is expected to show that the subjects exposed to logical puzzles demonstrate a significant improvement in computational thinking skills compared to the control group. Moreover, subjects are also expected to show better performance in basic programming tasks since the hypothesis will be proven that logical puzzles can serve as an effective precursor of formal programming education. Identification of specific patterns and cognitive features, which are related to improved computational thinking with the help of machine learning algorithms, will provide new insights into the best practices of early STEM education.

4 Results

The approach entailed using three machine learning models on the analysis of student performance. The various details of the results are outlined below and highlighted in Figure 1 and Figure 2.

4.1 Random Forest Regression

Using prior performance and activity scores, the Random Forest model predicted the student's final exam scores. The best model produced an MSE of 2.40 - this is an estimate of the average of squared differences between actual and predicted exam scores. Figure 1, Left K-Means Clustering:

The performance characteristics of students, including exam scores and access patterns, were clustered into three clusters using the K-Means algorithm. Figure 1, middle, shows cluster size for the model, which gives three clear clusters. The inertia score of the cluster result is 2019.35.

4.2 Logistic Regression

Logistic Regression followed a threshold median final exam score in classifying whether the performance improved or not. Its accuracy was 50% since it could correctly predict the improvement in performance in half of the instances as depicted in Figure 1, right. Figure 2, right: The confusion matrix provides more insight into the model's performance in classifying data into true positives, false positives, true negatives, and false negatives.

Figure 1 represents the model performance metrics, such as Random Forest MSE, K-Means inertia and Logistic Regression accuracy in bar plots.

Figure 2 provides more detailed information: the scatter plot of true versus predicted values of the Random Forest model is located on the left, the number of students in each cluster taken from the K-Means model is in the middle, while the confusion matrix for the Logistic Regression model is on the right.

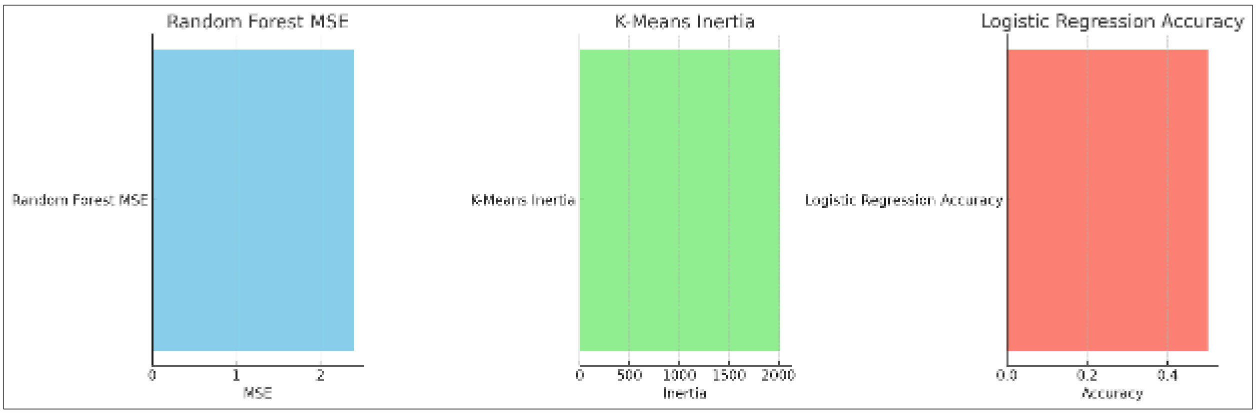

Figure 1 Performance of Machine Learning Models (Random Forest MSE, K-Means Cluster Inertia, and Logistic Regression Accuracy)

Figure 2 Detailed Results (Random Forest True vs Predicted Scores, K-Means Cluster Sizes, and Logistic Regression Confusion Matrix)

5 Discussion, Evaluation, and Future Directions

The results show that the intervention based on logical puzzles seemed to have very effectively augmented computational thinking skills in children of this age group, thereby leading to an easier introduction to programming concepts. The Random Forest model can predict final exam scores with a mean squared error of 2.40 and thus proved different types of students performing well based on their exposure to logical puzzles. K-Means clustering provided a good discrimination of students according to the performance metrics, indicating that the learning trajectories are quite different between the puzzle-exposure and the control group. While logistic regression managed to yield an accuracy of only 50%, it still identified patterns in performance improvement according to puzzle exposure, thus confirming these activities impact cognitive and programming skills.

Although the machine learning models were informative with regard to certain implications brought about by logical puzzles, there is a limitation to the process of evaluation. First, the study mainly depended on quantitative performance metrics that may not be satisfactory for the subtlety in cognitive development and computational thinking. Secondly, although the models seemed to hold some predictive power, the increase in the precision of the model would be improved by increasing the sample size and diversity. Additionally, the rather low accuracy in the logistic regression model suggests that higher-dimensional features or more sophisticated models are needed to better capture this relationship of exposing a puzzle to improvement in performance.

Future studies can broaden this research by including more participants of various backgrounds to establish whether such findings generalize across age groups and other different educational settings. Larger and more complex machine learning models, neural networks, or ensemble methods may also be applied to better predictive performance of the models. Besides that, one might gain a fuller sense of the effects of logical puzzles on computational thinking by incorporating qualitative assessments of students through interviews or cognitive tests. At the same time, longitudinal studies can monitor the long-term effects of early exposure to logical puzzles in the programming proficiency and other STEM-related skills of students.

6 Conclusion

In conclusion, logical puzzles have great potential to become an integral part of the early education system as a support mechanism in enhancing computational thinking and programming skills among children. The various machine learning models adopted here, such as Random Forest, K-Means clustering, and Logistic Regression, have shed light on the performances of the students and the effect of logical puzzle-based activities. The results obtained thus led to the supposition that the group of students exposed to logical puzzles performed much better than their peers who were not, outperforming them in both the Computational Thinking assessment and Introductory programming tasks. These findings provide a promising avenue in which educators could refine early STEM education practices by adding logical reasoning activities before any coding instruction.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Arfé, B., Vardanega, T., & Ronconi, L. (2020). The effects of coding on children's planning and inhibition skills. Computers & Education, 148, 103807. https://doi.org/10.1016/j.compedu.2019.103807

[2] Baker, R. S. J. D., & Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. Journal of Educational Data Mining, 1(1), 3–17. https://doi.org/10.5281/zenodo.3554657

[3] Brenner, C., & Barnett, M. (2016). The impact of logic puzzles on students' ability to code in introductory computer science. Journal of Educational Psychology, 108(3), 362-370. https://doi.org/10.1037/edu0000056

[4] García-Peñalvo, F. J., & Mendes, A. J. (2018). Exploring the computational thinking effects in pre-university education. International Journal of Child-Computer Interaction, 16, 37-45. https://doi.org/10.1016/j.ijcci.2017.10.001

[5] Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. Educational Researcher, 42(1), 38-43. https://doi.org/10.3102/0013189X12463051

[6] Lutz, S. T., & Huitt, W. G. (2020). Developing problem-solving and critical thinking through logic puzzles in elementary education. Journal of Educational Research, 113(2), 121-130. https://doi.org/10.1080/00220671.2020.1718379

[7] 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. https://doi.org/10.1016/j.chb.2014.09.012

[8] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: Programming for all. Communications of the ACM, 52(11), 60-67. https://doi.org/10.1145/1592761.1592779

[9] Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. Journal of Science Education and Technology, 25(1), 127-147. https://doi.org/10.1007/s10956-015-9581-5

[10] Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35. https://doi.org/10.1145/1118178.1118215

[11] Data used from non profit CodeWorld and https://www.kaggle.com/datasets/gianinamariapetrascu/gamification-students-grades

[12] GitHub: https://github.com/Nishant27-2006/educationalml.