International Journal of Information and Education Technology, Vol. 16, No. 4, 2026

Factors Influencing Algorithmic Thinking among Pre-Service Primary Teachers: A Structural Equation Modeling Approach

Thai Doan Thi Minh, Huong Le Thi Thu*, Duong Lam Thuy, Binh Le Thi,
and Vinh Nguyen Huy

Faculty of Primary Education, Thai Nguyen University of Education, Thai Nguyen, Vietnam
Email: thaidtm@tnue.edu.vn (T.D.T.M.); lethithuhuong@tnue.edu.vn (H.L.T.T.); duonglt@tnue.edu.vn (D.L.T);
binhlt@tnue.edu.vn (B.L.T.); vinhnh@tnue.edu.vn (V.N.H.)
*Corresponding author

Manuscript received September 28, 2025; revised October 9, 2025; accepted November 14, 2025; published April 13, 2026

***Abstract*—Algorithmic Thinking (AT) is increasingly recognized as a vital 21st-century competency, especially within the context of educational digital transformation. This study investigates the factors influencing AT among prospective primary school teachers by developing and validating a theoretical framework using Structural Equation Modeling (SEM). Data were gathered from 215 Primary Education students enrolled in several pedagogical universities in Northern Vietnam. The results reveal that Programming Competence (PC) ($\beta = 0.334$, $p < 0.05$) and Learning Environment (LE) ($\beta = 0.290$, $p < 0.05$) significantly and directly impact this competence, while Learning Attitude (LA) ($\beta = 0.294$, $p < 0.05$) and Technological Self-Efficacy (TSE) ($\beta = 0.436$, $p < 0.05$) exert indirect effects through PC. Logical Thinking (LT), however, shows no significant influence in the model, indicating the need for further exploration of potential mediating or moderating factors. This study contributes novel evidence by validating a multidimensional structural model of AT among pre-service teachers in a developing-country context, thereby extending existing frameworks of computational and digital competence to the field of primary teacher education. The findings contribute to the theoretical discourse on AT in teacher education and offer practical insights for the design of teacher training programs, particularly in fostering programming practice, technological integration, and supportive learning environments.**

***Keywords*—algorithmic thinking, prospective primary school teachers, structural equation modeling, programming competence, learning environment**

I. INTRODUCTION

In the context of digital transformation and rapid technological advancement, Algorithmic Thinking (AT) has emerged as an essential competency in modern education. AT is not only the foundation for learning programming but also a critical tool that supports students in developing problem-solving skills, logical reasoning, and the ability to abstract. According to Wing [1], AT is a cognitive approach that enables individuals to solve problems systematically and efficiently. It is not confined to the field of computer science but is widely applicable across various disciplines, including mathematics, science, and the arts. Similarly, Csizmadia *et al.* [1, 2] and Dejarnette and Thomas [3, 4] emphasized AT as the foundation for learning programming and fostering analytical and problem-solving abilities across disciplines. Güler [3, 4] and Brown [2, 5] further underscored its importance as a transferable skill that extends beyond computer science, supporting students in developing structured reasoning, creativity, and the ability to design solutions applicable to real-world contexts. Within teacher education, cultivating AT is particularly vital for future primary school teachers to strengthen their problem-solving and digital-integration capacities, in alignment with the requirements of the 2018 General Education Curriculum in Vietnam. According to various scholars, algorithmic thinking not only supports the resolution of current situations but also lays the groundwork for developing generalized solutions that can be applied to similar problems in the future [6, 7]

Algorithmic thinking plays a vital role not only in computer science but also across various domains of life and education-particularly in the learning process, where students need to develop logical reasoning, creative thinking, and the capacity to solve real-world problems [5, 8]. In addition, practicing algorithmic thinking helps students develop critical and creative thinking by engaging in the design, testing, and refinement of algorithms to address real-life situations.

Recent research on Algorithmic Thinking (AT) can be grouped into three major thematic directions. First, a number of studies have concentrated on the development and validation of assessment instruments for evaluating AT competence. Adorni and Piatti [9] developed an assessment tool for evaluating algorithmic thinking skills among students in general education. Villalustre and Cueli [10] constructed an assessment instrument to measure the proficiency level of this cognitive approach in prospective primary school teachers; Jin and Cutumisu [11] developed an online training environment integrated with machine learning technology to predict and classify algorithmic thinking abilities based on teachers' learning activity data. Similarly, Li *et al.* [12] investigated the perceptions and beliefs of preservice primary school teachers regarding algorithmic thinking and subsequently designed a detailed survey instrument.

Second, a large body of research has focused on teaching strategies and learning environments for developing AT. The study by Adorni *et al.* [9] compared the effectiveness of teaching algorithmic thinking through traditional (unplugged) methods versus digital technology-based approaches in the context of K-12 education. In addition, numerous studies have proposed effective strategies for developing this competence, such as using block-based programming tools that make it easier for students to approach and understand fundamental programming concepts, thereby enhancing their algorithmic thinking skills [13]; Hsu, Chang, and Hung [14] identified problem-based learning as one of the most effective strategies for teaching algorithmic thinking. Doğan [5, 8] and Dong *et al.* [15] emphasized the role of

doi: 10.18178/ijiet.2026.16.4.2566

technology-supported environments—such as Scratch, Arduino, or other visual programming interfaces—in promoting programming competence and algorithmic reasoning among pre-service teachers. Pala and Türker [16] suggested using the Arduino IDE in combination with C++ programming to cultivate creative, algorithmic, and critical thinking skills among future primary school teachers.

Third, recent trends have underscored the integration of AT within STEM education as a cross-disciplinary approach to enhance problem-solving and logical reasoning. There is a growing trend toward integrating algorithmic thinking into Science, Technology, Engineering, and Mathematics (STEM) subjects to foster students' problem-solving and logical reasoning skills [17, 18]. Integrating algorithmic thinking activities into subjects like mathematics, science, and the arts helps students recognize its practical applications [19–21]. Eli Sheldon [22] emphasized the importance of embedding algorithmic thinking across various educational domains.

However, many studies have pointed out that, globally, most models or approaches for developing algorithmic thinking have primarily focused on secondary education, leaving a gap in the training of future teachers [8, 15, 23, 24]. In practice, many prospective primary school teachers still exhibit limited algorithmic thinking competence, particularly in applying it to lesson planning and organizing technology-enhanced learning activities [8, 12, 15, 25]. Algorithmic thinking plays a vital role in enabling teachers to design learning activities that are logical, analytical, and problem-solving-oriented. It also facilitates the flexible integration of information technology into instruction, thereby enhancing the quality of teaching and learning [26].

Primary school teachers play a pivotal role in shaping and fostering Algorithmic Thinking (AT) in students from the earliest years. Integrating AT into the primary curriculum not only familiarizes students with fundamental programming concepts but also promotes critical and creative thinking. However, to implement this effectively, teachers must be well-equipped with the necessary knowledge and skills related to AT, as well as maintain a positive attitude and confidence in applying technology in their teaching. Numerous studies have highlighted that the development of AT in prospective primary school teachers depends on various influencing factors. According to Yadav *et al.* [27], a positive learning attitude and high technological self-efficacy are strongly associated with the development of Algorithmic Thinking (AT). Additionally, a supportive learning environment and programming competence are also considered key factors influencing the AT of future teachers. However, in Vietnam, research on this topic remains limited—particularly in the application of Structural Equation Modeling (SEM) to examine the relationships among the factors affecting AT. In response to this situation, the present study was conducted to analyze the factors influencing the Algorithmic Thinking (AT) of prospective primary school teachers in Vietnam. Specifically, the research focuses on examining the relationships among the following variables: Logical Thinking (LT), programming Competence (PC), Learning Attitude (LA), technological Self-Efficacy (TSE), Learning Environment (LE), and Algorithmic Thinking (AT).

Despite the increasing attention to Algorithmic Thinking research worldwide, there remains a lack of empirical evidence applying Structural Equation Modeling (SEM) to examine the interrelationships among cognitive, affective, and environmental factors influencing AT development among pre-service primary teachers in developing-country contexts such as Vietnam.

The main research questions include:

1) What factors have direct and indirect effects on the Algorithmic Thinking (AT) of prospective primary school teachers?
2) To what extent does each factor influence AT?
3) Is the proposed SEM model consistent with the empirical data?

This study has significant theoretical implications for expanding understanding of the factors that influence AT among future primary school teachers, particularly within the Vietnamese educational context. The application of Structural Equation Modeling (SEM) enables a comprehensive and precise examination of the relationships among variables, thereby providing a scientific foundation for designing appropriate teacher education programs.

Based on the theoretical framework and prior studies, a conceptual model was proposed to examine the hypothesized relationships among the six latent variables: Logical Thinking (LT), Learning Attitude (LA), Technological Self-Efficacy (TSE), Learning Environment (LE), Programming Competence (PC), and Algorithmic Thinking (AT). The model is presented in Fig. 1 below.

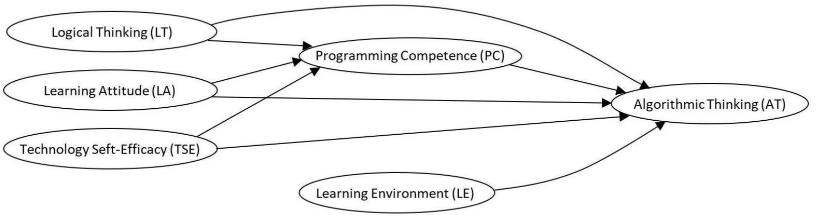


Fig. 1. Conceptual model illustrating hypothesized relationships among LT, LA, TSE, LE, PC, and AT.

From a practical perspective, the findings of this study will help teacher training institutions identify key factors to focus on in the teaching process to enhance AT competence among education students. Moreover, the study offers valuable insights for education policymakers in developing training policies and programs that meet the demands of the digital era.

II. MATERIALS AND METHODS

A. Research Design

This study employed a quantitative approach with a research model design aimed at testing the causal relationships among factors influencing the Algorithmic Thinking (AT) of prospective primary school teachers. Based on the theoretical framework and a comprehensive review of prior studies, a theoretical model was developed comprising six latent variables: Logical Thinking (LT), Learning Attitude (LA), Technological Self-Efficacy (TSE), Learning Environment (LE), Programming Competence (PC), and Algorithmic Thinking (AT). Data were collected through a questionnaire and analyzed using Structural Equation Modeling (SEM) with the GSCA software.

B. Participants and Sample

The participants of this study were third- and fourth-year students majoring in Primary Education at pedagogical universities located in the midland and mountainous regions of Northern Vietnam. This group was selected due to their exposure to programming content and the teaching of STEM subjects at the primary level. This study employed a convenience sampling approach. The questionnaire was designed and distributed to students majoring in Primary Education at Thai Nguyen University of Education through the Zalo application between May 25, 2025 and June 3, 2025. A total of approximately 300 invitations were sent out, and 243 responses were received, representing a response rate of 81%. After data screening, 28 responses were excluded due to invalid or incomplete answers (e.g., selecting only one option for all items). The final valid dataset comprised 215 responses, accounting for 71.7% of the total invitations.

Regarding sample size requirements, Kock and Hadaya [28] suggest a minimum sample size of 100 to 200 participants, while Anderson and Gerbing [29] argue that a sample size of 100 is sufficient for convergence, and a sample of 150 is adequate to achieve both convergence and accuracy when analyzing constructs with at least three indicators. In this study, the sample size was determined using a tool recommended by Kline [30], which indicated a minimum required size of 200. The actual sample collected exceeded this threshold, satisfying the requirements for both reliability and statistical representativeness. A total of 215 valid responses were collected and included in the analysis.

C. Instruments and Measurements

The questionnaire was designed with 27 items corresponding to six latent variables, using a five-point Likert scale (1 = Strongly Disagree; 5 = Strongly Agree). The measurement scales for each variable were developed based on a synthesis of reputable studies in the field.

- Algorithmic Thinking (AT): The scale was adapted from the model proposed by Román-González *et al.* [31].
- Programming Competence (PC): Developed based on Brennan and Resnick [26] focusing on the ability to write code, debug, use variables, and implement control structures.
- Technological Self-Efficacy (TSE): Adapted from Bandura [32] self-efficacy scale, including indicators related to confidence in using digital tools.
- Learning Attitude (LA): Inherited from Yadav *et al.* [27], measuring students' proactivity, persistence, and interest in learning programming.
- Logical Thinking (LT) and Learning Environment (LE): Constructed based on studies conducted in the context of programming education for teachers [8, 33].

The questionnaire used in this study was developed based on validated scales from prior research on algorithmic thinking, programming competence, learning attitude, and related constructs. The full questionnaire, organized by construct, is presented in Table 1 below.

Table 1. Survey items

Items	Survey Items
lt1	I often break down a teaching problem into small, clear steps.
lt2	I have the ability to reason sequentially, using the result of the previous step as the foundation for the next.
lt3	I verify my hypotheses in a logical sequence before proposing a solution.
lt4	I continuously adjust my reasoning based on newly collected information.
pc1	I am confident in writing or editing simple code (e.g., branching, loops, conditions) to solve problems.
pc2	I have a clear understanding of how to use variables and basic functions in a programming language.
pc3	I can identify and fix syntax or logic errors in a piece of code.
pc4	I am able to read and interpret block diagrams/algorithm flowcharts.
la1	I believe algorithmic thinking is an important skill worth investing time in learning.
la2	I actively seek opportunities to practice programming or explore new algorithms.
la3	I am excited to share my programming experiences with friends.
la4	I persist in trying multiple approaches when faced with difficult programming problems.
tse1	I am confident in selecting and using software and digital tools to support teaching and learning activities.
tse2	When encountering technical issues, I can find solutions on my own or learn quickly.
tse3	I feel competent enough to guide others (friends, students) in using technology.
tse4	I proactively update and familiarize myself with new technologies for teaching purposes.
le1	I am provided with resources (devices, internet access, documents) to practice programming in a training environment.
le2	I receive support from instructors and peers when I face difficulties in algorithmic thinking.
le3	I have opportunities to participate in discussions, workshops, or group projects to develop algorithm skills.
le4	I am encouraged to self-reflect and self-assess my learning outcomes.
le5	I am encouraged to share my results and experiences with friends and teaching communities.
at1	I can decompose a complex problem into simpler sub-problems for better understanding and resolution.
at2	I can identify and clearly understand the inputs, outputs, and constraints of a specific problem.
at3	I know how to abstract information, focusing on the core elements of a problem.
at4	I can generalize an operation on individual objects into an operation on a class of objects, thereby abstracting it into a new concept.
at5	I can identify basic operations and build accurate algorithms to solve problems.
at6	I can compare different approaches to accomplishing a task and identify the optimal one.

D. Data Analysis

To validate the proposed research model, the study employed the Generalized Structured Component Analysis (GSCA) technique. Comparative studies have shown that GSCA-SEM outperforms PLS-SEM in terms of loading consistency, standard errors, and parameter estimation capability [34, 35]. Moreover, GSCA is a full-information method that optimizes a unified criterion for both measurement and structural models. It supports both reflective and formative constructs—unlike PLS-SEM, which only performs partial optimization and does not assess overall model fit indices (such as FIT, AFIT, and Goodness-of-Fit Index) [36]. In addition, simulation-based experimental models indicate that GSCA (with reflective indicators) yields better parameter recovery and statistical power compared to PLSPM [37]. Finally, GSCA does not require data to follow a normal distribution, which helps avoid the risk of invalid solutions and allows for overall model fit assessment—an aspect particularly well-suited to the present research context.

III. RESULTS

A. Measurement Model Evaluation

To ensure the validity and reliability of the latent constructs in the model, the study assessed several measurement quality indicators, including internal consistency reliability (Cronbach's Alpha). A summary of the construct reliability and validity indicators is provided in Table 2.

Table 2. Construct reliability and convergent validity indicators

Construct property	LT	PC	LA	TSE	LE	AT
AVE	0.726	0.812	0.695	0.755	0.742	0.756
Alpha	0.874	0.923	0.853	0.892	0.913	0.935
Rho	0.914	0.945	0.901	0.925	0.935	0.949
Dimensionality	1.0	1.0	1.0	1.0	1.0	1.0

All constructs meet the threshold values for AVE > 0.50 and Alpha > 0.70, indicating adequate reliability. The results show that all latent constructs meet the required reliability thresholds. Specifically, the Cronbach's Alpha coefficients for the variable groups range from 0.853 to 0.935, exceeding the minimum threshold of 0.70 recommended by Hair *et al.* [38]. These results reflect a high level of internal consistency among the observed variables within each construct. Notably, the "Algorithmic Thinking" construct achieved the highest Cronbach's Alpha (α = 0.935), indicating exceptionally strong coherence among its component indicators. Additionally, the Composite Reliability (CR) values for all constructs exceeded 0.90, ranging from 0.901 to 0.949. This is considered a robust indicator in modern SEM analysis, suggesting that the observed variables reliably represent their respective latent constructs. For example, the "Programming Competence" construct recorded a CR (Rho) of 0.949—an exceptionally high value that reflects both accuracy and stability in measurement.

The Average Variance Extracted (AVE) values for all constructs surpassed the recommended threshold of 0.50, ranging from 0.695 (Learning Attitude—LA) to 0.812 (Programming Competence—PC). This indicates that each set of indicators explains the majority of variance in its associated latent construct, thereby confirming the convergent validity of the measurement scales. Finally, all constructs achieved perfect unidimensionality (Unidimensionality = 1.0), confirming that each group of indicators measures a single dimension of its respective latent concept.

Thus, the overall measurement model in this study satisfies the requirements for both reliability and validity, providing a solid foundation for subsequent structural model testing.

B. Factor Loadings

To evaluate the strength of association between the observed variables and the latent constructs in the model, the study analyzed the standardized factor loadings for each measurement scale. The results indicate that all indicators had loadings greater than 0.70—the minimum recommended threshold to ensure convergent validity [38]. This confirms that the observed variables were well-designed and accurately reflect the intended latent constructs. The standardized factor loadings for all observed variables are presented in Table 3.

Table 3. Standardized factor loadings of observed indicators

Construct	Indicator	Estimate	SE	95% CI	
LT	lt1	0.853	0.025	0.803	0.9
	lt2	0.878	0.024	0.809	0.918
	lt3	0.859	0.024	0.817	0.904
	lt4	0.817	0.038	0.733	0.882
PC	pc1	0.867	0.019	0.831	0.902
	pc2	0.918	0.013	0.896	0.943
	pc3	0.911	0.017	0.87	0.938
	pc4	0.907	0.016	0.872	0.93
LA	la1	0.76	0.032	0.691	0.816
	la2	0.855	0.021	0.811	0.894
	la3	0.877	0.021	0.833	0.917
	la4	0.839	0.025	0.785	0.881
TSE	tse1	0.854	0.024	0.806	0.898
	tse2	0.871	0.018	0.842	0.902
	tse3	0.885	0.018	0.844	0.914
	tse4	0.866	0.021	0.825	0.904
LE	le1	0.825	0.027	0.775	0.881
	le2	0.867	0.023	0.826	0.917
	le3	0.824	0.036	0.74	0.887
	le4	0.893	0.023	0.843	0.933
	le5	0.894	0.018	0.856	0.924
AT	at1	0.876	0.018	0.84	0.91
	at2	0.83	0.031	0.762	0.883
	at3	0.899	0.013	0.874	0.926
	at4	0.894	0.017	0.86	0.925
	at5	0.881	0.02	0.848	0.92
	at6	0.834	0.029	0.769	0.88

Specifically, the "Logical Thinking" (LT) construct comprises four indicators (lt1–lt4), with factor loadings ranging from 0.817 to 0.878, and 95% confidence intervals consistently above 0.70. This indicates that abilities such as problem analysis, sequential reasoning, and logical inference strongly reflect learners' logical thinking competence. The indicators demonstrated high stability, with standard errors below 0.04.

The "Programming Competence" (PC) construct includes four indicators (pc1–pc4), all with loadings exceeding 0.90, ranging from 0.867 to 0.918. This construct exhibited the highest level of internal consistency within the entire model, clearly capturing learners' practical programming abilities—from writing code and using functions to debugging and adjusting algorithms. The narrow confidence intervals indicate high estimation precision.

For the "Learning Attitude" (LA) construct, four indicators (la1–la4) had loadings between 0.760 and 0.877. Although la1 had the lowest loading (0.760), it still met the threshold for convergent validity. These indicators reflect learners' readiness, enthusiasm, proactivity, and persistence in learning programming—key psychological traits that significantly influence the development of professional competence.

The "Technological Self-Efficacy" (TSE) construct consisted of four indicators (tse1–tse4) with strong loadings ranging from 0.854 to 0.885. These indicators represent learners' confidence in using technology to solve problems, assist others, update software, and troubleshoot. This construct combines skill-based and attitudinal components, and the indicators demonstrated reliable and stable measurement.

The "Learning Environment" (LE) construct included five indicators (le1–le5), with loadings ranging from 0.824 to 0.894. Indicators such as facilities, support from instructors and peers, group academic activities, and learning feedback were rated by participants as strong representations of the learning environment that supports the development of thinking skills.

Finally, the "Algorithmic Thinking" (AT) construct—the main dependent variable in the model—comprised six indicators (at1–at6), with loadings ranging from 0.830 to 0.899. This construct demonstrated high convergence, capturing key features of algorithmic thinking such as algorithm representation, problem analysis, abstraction, and optimization.

Overall, the analysis results show that all observed variables had factor loadings greater than 0.70, with 95% confidence intervals clearly within the positive value range. This confirms the strong convergent validity and reliable measurement of each indicator with respect to its corresponding latent construct. These findings provide a robust basis for analyzing the structural relationships in the SEM model

C. Model Fit

To assess the degree of alignment between the proposed theoretical model and the empirical data, the study employed a range of model fit measures, including FIT, AFIT, GFI and SRMR. The results indicated that the model achieved a high and reliable level of overall fit (see Table 4).

Table 4. Model fit indices for the structural equation model

FIT	AFIT	GFI	SRMR
0.649	0.645	0.991	0.051

Specifically, the FIT index—which reflects the overall goodness-of-fit between the structural model and the data—reached a value of 0.649, while the Adjusted FIT (AFIT) was 0.645. Both values exceed the recommended minimum threshold of 0.50 [38]. This indicates that the model is capable of explaining a substantial portion of the variance in the observed data.

In addition, the Goodness-of-Fit Index (GFI) achieved a value of 0.991, which is close to the ideal value of 1.0, confirming the excellent compatibility between the model and the data. Meanwhile, the Standardized Root Mean Square Residual (SRMR)—one of the most widely used indices for assessing overall model fit—was 0.051, below the recommended threshold of 0.08. This suggests that the standardized average residual between the observed and predicted covariance matrices is very small, reinforcing the model's overall fit.

Overall, the fit indices collectively demonstrate that the proposed theoretical model exhibits a very good fit with the survey data. This provides a reliable foundation for hypothesis testing and further analysis of the relationships among the factors influencing algorithmic thinking in prospective primary school teachers.

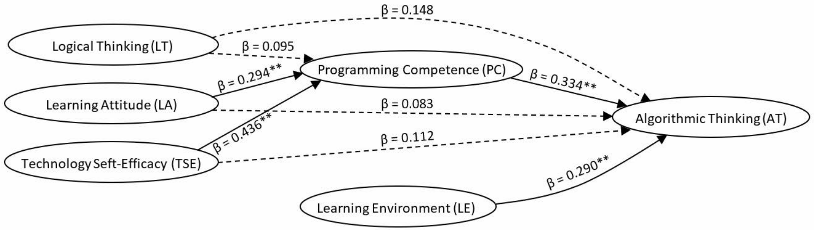
D. Path Coefficients

The results of hypothesis testing within the structural model are presented through path coefficients, Standard Errors (SE), and 95% Confidence Intervals (CI). Statistically significant relationships were identified when the confidence intervals did not include the value zero. These results are summarized in Table 5.

Table 5. Standardized path coefficients and significance levels in the structural model

Path	Estimate	SE	95%CI	
LT→PC	0.095	0.08	−0.034	0.286
LA→PC	0.294	0.094	0.105	0.455
TSE→PC	0.436	0.087	0.277	0.691
LT→AT	0.148	0.074	−0.006	0.293
PC→AT	0.334	0.08	0.22	0.525
LA→AT	0.083	0.081	−0.076	0.236
TSE→AT	0.112	0.102	−0.095	0.288
LE→AT	0.29	0.091	0.101	0.447

The results of the hypothesis testing are visually summarized in Fig. 2, which presents the final structural model with standardized path coefficients and statistical significance levels. This model illustrates the direct and indirect relationships among the latent variables examined in this study.



Solid arrows indicate significant paths ($p < 0.05$).
Dashed arrows indicate non-significant paths ($p > 0.05$).
Values on the paths represent standardized coefficients (β).

Fig. 2. The final structural model with standardized path coefficients.

As shown in Fig. 2, the effects of Learning Attitude (LA) and Tech Self-Efficacy (TSE) on Algorithmic Thinking (AT) are mediated through Programming Competence (PC), while Logical Thinking (LT) does not exhibit any significant direct or indirect influence.

First, the results indicate that Technological Self-Efficacy (TSE) has a strong and statistically significant effect on Programming Competence (PC), with an estimated coefficient of $\beta = 0.436$, CI [0.277; 0.691]. This suggests that students with higher confidence in using technology are more likely to develop stronger programming skills. Similarly, Learning Attitude (LA) also exerts a significant positive effect on PC ($\beta = 0.294$, CI [0.105; 0.455]), highlighting the importance of motivation and enthusiasm in enhancing programming skills among prospective primary school teachers.

Another key finding is that Programming Competence (PC) has a direct and statistically significant effect on Algorithmic Thinking (AT) ($\beta = 0.334$, CI [0.220; 0.525]). This relationship confirms the mediating role of PC in translating foundational psychological and technical factors into higher-order thinking capabilities.

In addition, the Learning Environment (LE) demonstrates a significant direct impact on AT ($\beta = 0.290$, CI [0.101; 0.447]), underscoring the importance of supportive learning conditions such as infrastructure, expert guidance, and collaborative activities in fostering algorithmic thinking.

Conversely, several hypotheses in the model were not statistically supported. Specifically, the effects of Logical Thinking (LT) on Programming Competence (PC) ($\beta = 0.095$, CI [–0.034; 0.286]) and on Algorithmic Thinking (AT) ($\beta = 0.148$, CI [–0.006; 0.293]) were not significant. Similarly, the direct effects of Learning Attitude (LA) on AT ($\beta = 0.083$, CI [–0.076; 0.236]) and Technological Self-Efficacy (TSE) on AT ($\beta = 0.112$, CI [–0.095; 0.288]) did not reach statistical significance. These findings suggest that LT, LA, and TSE may influence algorithmic thinking indirectly through the mediating role of programming competence, a possibility that warrants further examination through mediation analysis.

In summary, the proposed theoretical model confirmed four statistically significant direct relationships, clarifying the central role of programming competence in the development of algorithmic thinking. It also highlighted the influence of psychological–technological and environmental factors in shaping this essential competency among prospective primary school teachers.

IV. DISCUSSIONS

Although the present study focused primarily on testing the hypothesized structural relationships among variables, future research is encouraged to include additional statistical indicators such as the correlation matrix, R^2 and f^2 effect sizes, and explicit significance level annotations to provide deeper insights into the model's explanatory power.

The findings of this study clarified the direct and indirect factors influencing Algorithmic Thinking (AT) among prospective primary school teachers, thereby contributing to the theoretical understanding of digital competence development in teacher education. Through the application of Structural Equation Modeling (SEM), the study identified Programming Competence (PC) and Learning Environment (LE) as two factors with direct and statistically significant effects on AT, while Learning Attitude (LA), Technological Self-Efficacy (TSE), and Logical Thinking (LT) played indirect roles mediated by PC. These findings not only support the initial theoretical hypotheses but also offer important implications for improving teacher training programs in the digital era.

One of the key findings of this study is the strong influence of programming competence on Algorithmic Thinking (AT) ($\beta = 0.334$, CI [0.220; 0.525]). This result aligns with the theoretical framework proposed by Brennan and Resnick [26], which views programming as both a means of expressing and developing algorithmic thinking. Similarly, Román-González *et al.* [31] emphasized the close relationship between programming competence and core components of AT, such as problem decomposition, abstraction, and algorithm design.

In the context of teacher education, programming skills are not merely a technological requirement but also serve as a cognitive tool that enables teachers to approach and structure learning content in a logical, systematic, and reusable manner—hallmarks of algorithmic thinking.

In addition, the learning environment was identified as a significant direct factor influencing Algorithmic Thinking (AT) ($\beta = 0.290$, CI [0.101; 0.447]). This finding is consistent with the study by Doğan [8] which highlighted that a learning environment supported by digital resources, active interaction between learners and instructors, and opportunities for hands-on practice plays a critical role in fostering AT development among preservice teachers.

Another noteworthy point is that, although Learning Attitude (LA) and Technological Self-Efficacy (TSE) did not have direct effects on Algorithmic Thinking (AT), they significantly influenced Programming Competence (PC)—the mediating factor in the model. Specifically, TSE had the strongest impact on PC ($\beta = 0.436$, CI [0.277; 0.691]), followed by LA ($\beta = 0.294$, CI [0.105; 0.455]). This finding supports previous studies by Yadav *et al.* [39], which emphasized that a positive attitude and confidence in one's technological abilities are strong predictors of programming behavior and persistence in learning. The study by Butler and Leahy [33] further reinforces this perspective, suggesting that improving learning attitudes and technological self-efficacy can indirectly enhance algorithmic thinking by strengthening programming competence. Therefore, while LA and TSE may not directly boost AT, their development is nonetheless foundational, as they prepare learners to engage with and master technology—ultimately supporting gains in both programming skills and algorithmic thinking.

In contrast, the effects of logical thinking on programming competence ($\beta = 0.095$) and algorithmic thinking ($\beta = 0.148$) were not statistically significant. This may stem from the fact that technical, software-related, and experiential learning factors play a more dominant role in shaping programming competence than abstract reasoning alone. Another possible explanation is that the measurement of Logical Thinking (LT) in this study focused primarily on general cognitive reasoning rather than task-specific logical sequencing used in programming practice. This conceptual gap might have reduced its predictive strength in the SEM model. Additionally, many pre-service teachers in Vietnam have limited exposure to formal logic training or algorithm-based tasks during their undergraduate studies, which could weaken the observed link between LT and AT. Cultural and curricular differences may also account for the divergence from findings in studies conducted in Western contexts, where logic-oriented coursework is more explicitly integrated into teacher education programs. This finding contrasts with some traditional assumptions in computer science education, which posit logical thinking as a prerequisite for developing algorithmic thinking [40]. The findings of this study are consistent with several international studies that emphasize the contextual and affective dimensions of algorithmic thinking development. For instance, Adorni and Piatti [9] in Italy and Villalustre and Cueli [10] in Spain both highlighted that the development of algorithmic thinking depends not only on logical reasoning but also on pedagogical experience and environmental support. Similarly, Li *et al.* [12] reported that in the Chinese context, pre-service teachers' algorithmic competence is more strongly influenced by their learning attitudes and technological self-efficacy than by cognitive ability alone. These results are in line with Doğan [8] and Hsu *et al.* [14], who found that digital learning environments and problem-based learning strategies play a critical mediating role in connecting logical thinking with practical programming and algorithmic skills. Together, these findings suggest that the non-significant effect of Logical Thinking (LT) observed in this study reflects a broader international trend—where cognitive factors serve as a foundation but are not sufficient predictors of algorithmic competence without motivational and contextual reinforcement. However, in the context of preservice teachers who may lack a strong background in mathematics or computer science, this result highlights the need to reorganize programming instruction toward project-based learning. Such an approach can foster algorithmic thinking in a more experiential, intuitive, and practice-oriented manner—better aligned with the realities of classroom teaching.

Based on the above findings, this study contributes to the theoretical framework by validating a multi-factor model for developing Algorithmic Thinking (AT) among future teachers, in which programming competence serves as a key mediating variable. At the same time, the research offers a practical perspective: developing AT should not rely solely on technical content, but also requires interventions in learners' attitudes, technological self-beliefs, and—most importantly—the creation of interactive, supportive, and hands-on learning environments. These results align with UNESCO's vision for 21st-century teacher digital competence development [41], and have high applicability in current primary teacher training programs in Vietnam.

The findings of this study offer several practical implications for the design and implementation of teacher training programs in the context of Algorithmic Thinking (AT) development. First, the central role of programming competence in the model indicates that teacher education curricula should incorporate applied programming modules that emphasize basic algorithmic manipulation, debugging skills, and the development of sequential logic through visual programming languages such as Scratch or Python.

In addition, the significant influence of the learning environment on AT highlights the importance of creating flexible learning spaces with adequate technological support, open-access resources, and collaborative learning activities such as group projects, workshops, or coding clubs. The results also suggest the need to strengthen activities that foster cognitive and emotional engagement, particularly the cultivation of positive attitudes and confidence in using technology—critical foundations that encourage learners to independently explore and utilize digital tools to enhance their competencies.

Thus, education managers and curriculum designers should systematically integrate academic content, hands-on experience, and supportive learning environments to create the optimal conditions for fostering algorithmic thinking—a core competency in modern education.

This study primarily relied on self-reported survey data, which may be subject to response bias. Future research should consider incorporating more objective assessment methods, such as performance evaluations or classroom observations. Moreover, extending the study to include teachers at other educational levels or in diverse instructional settings would provide a more comprehensive understanding of the factors that influence the development of algorithmic thinking.

V. CONCLUSION

This study validated a structural model explaining the interrelationships among cognitive, affective, and environmental factors influencing Algorithmic Thinking (AT) among pre-service primary teachers in Vietnam. The results confirmed that Programming Competence (PC), Technological Self-Efficacy (TSE), Learning Attitude (LA), and Learning Environment (LE) significantly contribute to the development of AT, while Logical Thinking (LT) exerts no direct effect. These findings suggest that algorithmic thinking is a multidimensional competence shaped not only by cognitive ability but also by motivation and contextual learning conditions.

These results suggest that sustainable development of algorithmic thinking in teacher education requires more than technical content; it also demands fostering learners' technological confidence and positive learning attitudes. Simultaneously, the learning environment—including instructional support, infrastructure, digital resources, and collaborative learning activities—should be recognized as a critical component in cultivating higher-order thinking skills in modern education.

From a policy perspective, the findings suggest that teacher-education programs should integrate algorithmic thinking through project-based programming and digital learning environments. Strengthening institutional support and teacher-training policies in digital pedagogy will help align future teachers' competencies with the national goals of educational digital transformation.

The study also raises theoretical questions regarding the inconsistent role of Logical Thinking (LT) within the model. While previous research has emphasized LT as foundational to AT, this study found its effect to be non-significant, pointing to the need for further exploration into potential indirect mechanisms or moderating variables. Finally, limitations include the reliance on self-reported data and a sample restricted to several teacher training institutions.

In summary, this study advances the understanding of algorithmic thinking among pre-service teachers by integrating cognitive, affective, and environmental dimensions into a unified structural model. It extends the digital competence framework by highlighting algorithmic thinking as a multifaceted competence shaped by self-efficacy, learning attitude, and supportive learning environments, offering a foundation for future cross-cultural research.

Despite its contributions, this study has certain limitations. The use of self-reported data may involve subjective bias, and the sample was confined to pre-service teachers in a single region, limiting the generalizability of the results. In addition, the study employed a cross-sectional design, which restricts causal inference. Future research should expand the sample to multiple regions and institutions, adopt mixed-method or longitudinal approaches, and examine additional moderating factors—such as teaching experience, gender, or institutional resources—to provide a more comprehensive understanding of algorithmic thinking competence in teacher education.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Thai Doan Thi Minh and Huong Le Thi Thu; Methodology, Thai Doan Thi Minh and Duong Lam Thuy; Formal analysis, Huong Le Thi Thu and Vinh Nguyen Huy; Investigation, Duong Lam Thuy and Binh Le Thi; Data curation, Binh Le Thi; Writing—original draft, Thai Doan Thi Minh and Duong Lam Thuy; Writing—review and editing, Huong Le Thi Thu and Thai Doan Thi Minh; Visualization, Vinh Nguyen Huy; Supervision, Huong Le Thi Thu. All authors have read and agreed to the published version of the manuscript.

Funding

This research was funded by Thai Nguyen University and Thai Nguyen University of Education under grant number ĐH2024-TN04-02.

References

[1] J. M. Wing, "Computational thinking," *Commun ACM*, vol. 49, no. 3, pp. 33–55, 2006. doi: 10.1145/1227504.1227378

[2] A. Csizmadia *et al. Computational Thinking: A Guide for Teachers*, Computing at School, BCS, The Chartered Institute for IT, 2015.

[3] E. Lockwood, A. Asay, A. F. Dejarnette, and M. Thomas, "Algorithmic thinking: An initial characterization of computational thinking in mathematics," *North American Chapter of the International Group for the Psychology of Mathematics Education*, pp. 1588–1595, 2016.

[4] Ç. Güler, "Algorithmic thinking skills without computers for prospective computer science teachers," *Journal of Theoretical Educational Science*, vol. 14, no. 4, pp. 570–585, Oct. 2021. doi: 10.30831/akukeg.892869

[5] W. Brown, (2015). *Introduction to Algorithmic Thinking*. [Online]. Available: https://www.scribd.com/document/164778850/Introduction-to-Algorithmic-Thinking

[6] M. G. Voskoglou and S. Buckley, "Problem solving and computational thinking in a learning environment," *Egyptian Computer Science Journal, ECS*, vol. 36, no. 4, Dec. 2012.

[7] J. Lockwood and A. Mooney, "Computational thinking in education: Where does it fit? A systematic literary review," *International Journal of Computer Science Education in Schools*, vol. 2, no. 1, pp. 1–58, Mar. 2017.

[8] A. Doğan, "Algorithmic thinking in primary education," *International Journal of Progressive Education*, vol. 16, no. 4, pp. 286–301, 2020. doi: 10.29329/ijpe.2020.268.18

[9] G. Adorni and A. Piatti, "Designing the virtual CAT: A digital tool for algorithmic thinking assessment in compulsory education," arXiv preprint, arXiv:2408.01263v3, Aug. 2024.

[10] L. Villalustre and M. Cueli, "Assessing the computational thinking of pre-service teachers: A gender and robotics programming experience analysis," *Education Sciences*, vol. 13, no. 10, 1032, Oct. 2023. doi: 10.3390/EDUCSCI13101032.

[11] H. Y. Jin and M. Cutumisu, "Predicting pre-service teachers' computational thinking skills using machine learning classifiers," *Educ. Inf. Technol. (Dordr)*, vol. 28, no. 9, pp. 11447–11467, Sep. 2023. doi: 10.1007/S10639-023-11642-7

[12] B. Qing Li, S. W. McNary, and T. Boyd, "Assessment of computational thinking: A study of preservice teachers' knowledge and beliefs," *Athens J. Sci.*, vol. 10, no. 2, pp. 65–82, 2023. doi: 10.30958/ajs.10-2-1

[13] G. K. W. Wong, S. Jian, and H. Y. Cheung, "Engaging Children in developing algorithmic thinking and debugging skills in primary achools: A mixed-methods multiple case study," *Educ. Inf. Technol. (Dordr)*, vol. 29, no. 13, pp. 16205–16254, Sep. 2024. doi: 10.1007/s10639-024-12448-x

[14] T. C. Hsu, S. C. Chang, and Y. T. Hung, "How to learn and how to teach computational thinking: Suggestions based on a review of the literature," *Comput. Educ.*, vol. 126, pp. 296–310, Nov. 2018. doi: 10.1016/J.COMPEDU.2018.07.004

[15] W. Dong, Y. Li, L. Sun, and Y. Liu, "Developing pre-service teachers' computational thinking: A systematic literature review," *Int. J. Technol. Des. Educ.*, vol. 34, no. 1, pp. 191–227, Mar. 2024. doi: 10.1007/S10798-023-09811-3

[16] F. K. Pala and P. Mıhcı Türker, "The effects of different programming trainings on the computational thinking skills," *Interactive Learning Environments*, vol. 29, no. 7, pp. 1090–1100, 2021. doi: 10.1080/10494820.2019.1635495

[17] D. Yang, Y. Baek, Y.-H. Ching, S. Swanson, B. Chittoori, and S. Wang, "Infusing computational thinking in an integrated STEM curriculum: user reactions and lessons learned," *European Journal of STEM Education*, vol. 6, no. 1, 4, 2021. doi: 10.20897/ejsteme/9560

[18] S. I. Swaid, "Bringing computational thinking to STEM education," *Procedia Manuf.*, vol. 3, pp. 3657–3662, 2015. doi: 10.1016/J.PROMFG.2015.07.761

[19] F. Mumcu, E. Kıdıman, and F. Özdinç, "Integrating computational thinking into mathematics education through an unplugged computer science activity," *Journal of Pedagogical Research*, vol. 7, no. 2, pp. 72–92, Jun. 2023. doi: 10.33902/JPR.202318528

[20] R. Tariq, B. M. Aponte Babines, J. Ramirez, I. Alvarez-Icaza, and F. Naseer, "Computational thinking in STEM education: Current state-of-the-art and future research directions," *Front Comput Sci*, vol. 6, 1480404, Jan. 2024. doi: 10.3389/FCOMP.2024.1480404

[21] I. S. Milara and M. C. Orduña, "Possibilities and Challenges of STEAM Pedagogies," arXiv preprint, arXiv:2408.15282, Aug. 2024.

[22] Eli Sheldon. Computational Thinking across the Curriculum | Edutopia. [Online]. Available: https://www.edutopia.org/blog/computational-thinking-across-the-curriculum-eli-sheldon

[23] O. Meerbaum-Salant, M. Armoni, and M. Ben-Ari, "Learning computer science concepts with scratch," *Computer Science Education*, vol. 23, no. 3, pp. 239–264, Sep. 2013. doi: 10.1080/08993408.2013.832022.

[24] A. Repenning, D. Webb, and A. Ioannidou. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. [Online]. Available: http://scalablegamedesign.cs.colorado.edu

[25] E. Jasute and V. Dagiene, "The effects of teaching programming via scratch on problem solving skills: A discussion from learners' perspective," *Informatics in Education*, vol. 13, no. 1, pp. 33–50, Apr. 2014. doi: 10.15388/INFEDU.2014.03

[26] K. Brennan and M. Resnick, "New frameworks for studying and assessing the development of computational thinking," in *Proc. the 2012 Annual Meeting of the American Educational Research Association*, Vancouver, Canada, 2012, pp. 1–25. doi: 10.1007/978-3-319-64051-8_9

[27] A. Yadav, S. Gretter, S. Hambrusch, and P. Sands, "Computer science education expanding computer science education in schools: Understanding teacher experiences and challenges expanding computer science education in schools," *Computer Science Education*, 2016. doi: 10.1080/08993408.2016.1257418

[28] N. Kock and P. Hadaya, "Minimum sample size estimation in PLS-SEM: The inverse square root and gamma-exponential methods," *Information Systems Journal*, vol. 28, no. 1, pp. 227–261, Jan. 2018. doi: 10.1111/isj.12131

[29] J. C. Anderson and D. W. Gerbing, "The effect of sampling error on convergence, improper solutions, and goodness-of-fit indices for maximum likelihood confirmatory factor analysis," *Psychometrika*, vol. 49, no. 2, pp. 155–173, Jun. 1984. doi: 10.1007/BF02294170

[30] R. B. Kline, "Response to Leslie Hayduk's review of principles and practice of structural equation modeling," *Can Stud Popul*, vol. 45, no. 3–4, pp. 188–195, 2018. doi: 10.25336/CSP29418

[31] M. Román-González, J. C. Pérez-González, and C. Jiménez-Fernández,

"Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test," *Comput. Human Behav.*, vol. 72, pp. 678–691, Jul. 2017. doi: 10.1016/j.chb.2016.08.047

[32] A. Bandura, F. Pajares, and T. Urdan. (2006). Self-efficacy beliefs of adolescents. [Online]. pp. 307–337. Available: https://scispace.com/papers/self-efficacy-beliefs-of-adolescents-2xat2hl5k4?citations_page=29

[33] D. Butler and M. Leahy, "Developing preservice teachers' understanding of computational thinking: A constructionist approach," *British Journal of Educational Technology*, vol. 52, no. 3, pp. 1060–1077, May 2021. doi: 10.1111/BJET.13090

[34] H. Hwang, N. K. Malhotra, Y. Kim, M. A. Tomiuk, and S. Hong, "A comparative study on parameter recovery of three approaches to structural equation modeling," *Journal of Marketing Research*, vol. 47, no. 4, pp. 699–712, 2010. doi: 10.1509/JMKR.47.4.699

[35] H. Hwang and Y. Takane, "Generalized structured component analysis," *Psychometrika*, vol. 69, no. 1, pp. 81–99, 2004. doi: 10.1007/BF02295841

[36] U. Narimawati and J. Sarwono, "Theoretical approaches review on covariance based SEM using Lisrel, partial least based SEM using smart PLS and component based SEM using Gesca," *American Journal of Applied Mathematics*, vol. 12, no. 5, pp. 133–140, Sep. 2024. doi: 10.11648/J.AJAM.20241205.13

[37] G. Cho and J. Y. Choi, "An empirical comparison of generalized structured component analysis and partial least squares path modeling under variance-based structural equation models," *Behaviormetrika*, vol. 47, no. 1, pp. 243–272, Jan. 2020. doi: 10.1007/S41237-019-00098-0

[38] J. Hair, R. Anderson, B. Babin, and W. Black, *Multivariate Data Analysis*, vol. 7 ed., Pearson Prentice Hall, 2010.

[39] A. Yadav, C. Stephenson, and H. Hong, "Computational thinking for teacher education," *Commun ACM*, vol. 60, no. 4, pp. 55–62, Apr. 2017. doi: 10.1145/2994591

[40] F. J. García-Peñalvo, "What computational thinking is," *Journal of Information Technology Research*, vol. 9, no. 3, pp. 5–8 2016. doi: 10.1109/RITA.2018.2809939

[41] UNESCO. *Digital Frameworks*. [Online]. Available: https://unevoc.unesco.org/home/Digital+Competence+Frameworks