JOURNAL OF TECHNOLOGY AND HUMANITIES VOL. 6 NO. 2 (2025) 026–038

JTH

https://jthkkss.com/
e-ISSN 2805-4431
DOI: https://doi.org/10.53797/jthkks.v6i2.4.2025

Behavior Based Learning Analytics in Pair Programming: A Conceptual Approach to Enhance Programming Competency

Norhaslinda Abdul Karim[1*] & Nur Syadhila Che Lah[1]

[1]Universiti Teknologi Petronas, Seri Iskandar, Perak, MALAYSIA

*Corresponding author email: norhaslinda_24004184@utp.edu.my

Received 24 July 2025; Accepted 19 August 2025; Available online 01 Dec 2025

Abstract: Programming remains a significant challenge for students in higher education, particularly within technical and vocational institutions where foundational computing skills are often limited. Despite the adoption of various instructional strategies, high failure rates and low programming competency persist. Pair programming, a collaborative learning approach, has shown potential to enhance students' coding skills, engagement, and confidence. However, traditional pairing methods—often based on random assignment or technical ability—fail to account for individual learning behaviors that critically influence collaborative outcomes. This concept paper proposes a behavior-based pairing framework that integrates learning analytics to enhance programming competency through more strategic pair formation. Drawing on data extracted from Learning Management Systems (LMS) such as login frequency, activity completion, forum participation, and assignment submissions, the framework identifies two key learning behaviors: engagement and self-regulated learning (SRL). Clustering techniques are employed to group students according to these behavioral attributes, and heterogeneous pairing is applied to match partners with contrasting learning profiles. This approach aims to promote complementary collaboration, foster peer support, and enhance problem-solving effectiveness in programming tasks. The proposed framework aligns with the Malaysian Education Blueprint 2015–2025 (Higher Education), the National TVET Policy, and global initiatives linked to Education 4.0 and Industry 4.0. It also addresses the national agenda on graduate employability and supports data-informed teaching practices. By integrating behavioral insights with learning analytics, this concept introduces a personalized and evidence-based approach to programming education. It provides a foundation for future empirical validation and offers practical implications for improving curriculum design, instructional strategies, and student outcomes in computing education.

Keywords: Learning behavior, learning analytics, pair programming, engagement, self-regulated learning, programming competency

1. Introduction

The teaching of computer programming has evolved significantly since the mid-20th century, starting with the introduction of low-level languages such as Assembly and FORTRAN, which were originally aimed at scientists and engineers (Zima, 2007). With the development of technology and the accessibility of computers, programming education began to expand into academic and vocational curricula by the 1980s and 1990s. The introduction of high-level languages such as Pascal, C++ and later Java allowed a wider group of students to understand programming concepts with a greater focus on problem solving and logical reasoning rather than syntax alone (Duvall et al., 2021; Tuveri et al., 2022; Vinueza-Morales et al., 2025).

However, despite the many advances in programming pedagogy, the subject is still considered difficult by new students. Failure and dropout rates in basic programming courses continue to be reported across institutions (Simon et al., 2019). Key difficulties include understanding abstract concepts, algorithmic thinking, debugging, and applying theory

*Corresponding author: norhaslinda_24004184@utp.edu.my

to practical tasks. This reality continues to fuel debate about the most effective methods of teaching programming in today's digital education era.

At the same time, the emergence of Industrial Revolution 4.0 and Education 4.0 has completely changed the expectations for programming education, especially in the context of Technical and Vocational Education and Training (TVET). Programming skills are no longer considered exclusive to the field of computer science, but have become a basic requirement in forming digital human capital that is ready to face the challenges of future jobs and a list of ten skills that will be most required in jobs by 2025, one of them being "technology design and programming" (WEF_Future_of_Jobs_2020, 2020). Therefore, educators now need to choose and use the best methods to adapt more adaptive, collaborative and data-driven teaching. In response to this challenge, various pedagogical approaches have been explored such as problem-based learning, project-based learning, flipped classroom, and pair programming. These methods are based on Agile software development practices that have been proven to promote collaboration and cognitive engagement (Zhu & Ergulec, 2023). However, a fundamental question that has not yet been fully answered is: how can this method be optimized to truly fit the diverse learning behaviors of students?

1.1 Research Motivation

The ongoing difficulty in mastering programming competencies is a widely documented issue in higher education. Globally, entry-level programming courses report the highest failure rates in computing-related programs, at 33% (Bennedsen & Caspersen, 2019). The literature consistently reports that programming courses suffer from significant levels of student failure and dropout (Figueiredo & García-Peñalvo, 2024; Roque Hernández et al., 2021). In Malaysia, a similar pattern can be seen in polytechnics and technical institutions, where a large proportion of students face difficulties in mastering basic programming concepts, logic construction, and code error detection (Ali et al., 2021). This challenge not only affects academic progress, but also weakens the level of readiness of graduates for the digital economy as outlined in the Malaysian Education Blueprint (Higher Education) 2015–2025.

Pair programming has been widely used in educational and industrial settings. This method can be used to overcome programming challenges by offering benefits such as improved code quality, problem-solving skills, and student motivation (Leow & Huang, 2021). However, its effectiveness is inconsistent. A critical factor influencing outcomes is the method of pairing. Existing approaches typically rely on random assignment or matching based on skill level. Random pairing risks creating incompatibility of work styles and motivation levels. Skill-based matching often ignores other important factors in collaboration such as learning behaviors and cognitive strategies. As a result, some students benefit from the collaboration, while others experience frustration, decreased engagement, and minimal learning achievement.

In parallel, Learning Analytics (LA) has shown significant potential in identifying and measuring student learning behaviors through digital footprints in Learning Management Systems (LMS). Attributes such as engagement and self-regulated learning (SRL) have strong correlations with academic achievement and collaborative success (Ifenthaler & Yau, 2020). Despite this evidence, existing research rarely integrates behavioral analytics into the design of instructional strategies such as pair programming. This lack of integration suggests that there is untapped potential to personalize and enhance the effectiveness of collaborative learning. Pair programming and LA have each been proven to have positive effects. However, the potential synergy of combining the two is still underexplored. This creates the need for a systematic approach using LA to identify behavioral profiles, apply clustering techniques, and form pairs with behavioral compatibility that can maximize the effectiveness of collaboration. It also has the potential to significantly improve programming competence, especially in the context of TVET and polytechnics that have a significant diversity of students in terms of engagement and self-regulated learning (SRL).

1.2 New Concept in Pair Programming

Although pair programming has been introduced as a collaborative pedagogical strategy that can improve conceptual mastery and problem-solving skills among students, its effectiveness still shows significant variation. Many studies report that learning outcomes obtained from pair programming are not necessarily consistent between pairs of students (Bowman et al., 2020; Graßl & Fraser, 2024). Among the main factors identified is the method of pair formation that does not take into account compatibility in terms of learning behavior, level of engagement, and tendency to cooperate. Random matching or based solely on academic level has the potential to produce incompatible pairs and subsequently affect student work dynamics and motivation.

Furthermore, traditional teaching approaches that are still practiced in programming courses are often linear, lecturer-centered, and do not take into account the unique learning patterns of each individual. In today's digital learning environment, students show diversity in the way they access content, manage time, interact with systems and peers, and complete assignments. This mismatch of teaching approaches with behavior contributes to unbalanced levels of engagement and inconsistent achievement. In this context, learning behavior can be explored as a new approach that allows for more strategic student pairing. It is data-based and responsive to individual learning behavior. Technologies such as learning analytics, which are now increasingly used in digital education, offer the opportunity to understand student behavior more deeply through activity data in LMS platforms. However, the potential of learning analytics in the context of pair programming pairing has not yet been systematically explored in previous studies, especially in polytechnic and TVET environments. Therefore, a new conceptual approach that integrates student behavioral data in the pair programming pair formation process is not only relevant, but also needed to improve pair compatibility, optimize collaboration, and subsequently strengthen programming competencies among today's students.

2. Literature Analysis

2.1 Learning Behavior, Learning Analytics, and Pair Programming

i. Learning Analytics

Learning analytics is defined as the process of measuring, collecting, analyzing, and reporting student-related data with the aim of understanding and improving the learning process. It leverages log data from the Learning Management System (LMS) to track learning patterns and make evidence-based pedagogical decisions. According to Clow (2012), learning analytics works in a cycle: data → understanding → action → impact → feedback. In this study, learning analytics was used to identify different learning behaviors, which then became the basis for matching students in pair programming. 62% of the student-facing dashboards reviewed still rely solely on descriptive analytics (Paulsen & Lindsay, 2024). This study advances the field by employing deeper indicators such as metacognition and clustering, which go beyond purely descriptive analytics.

ii. Learning Behavior

Learning behavior refers to the patterns of student actions during the learning process, including how they access content, interact with peers, complete assignments, and manage their time and learning resources (Joshi et al., 2022). In the context of digital learning, this behavior can be observed through interaction data in systems such as Learning Management System (LMS), for example, login frequency, material viewing duration, and assignment submission patterns (Ifenthaler, 2020). This behavior is typically divided into several dimensions such as:

Table 1: Definitions of Learning Behavior

Learning behavior	Definition
Engagement	The degree of student participation and interaction with LMS features.
Self-Regulated Learning	These include planning, monitoring, and evaluating one's learning activities.
Collaboration	Interacting with students and facilitating student-to-student interactions through forums, chats, and collaborative tools.

iii. Pair Programming

Pair programming is a collaborative programming method in which two students work together on a coding task. One takes turn as the "driver" (writing code) and the other as the "observer" or "navigator" (analyzing and suggesting improvements) (Chigona & Pollock, 2008). In an educational context, pair programming improves the quality of programming, enhances programming skills, and increases self-confidence in programming tasks (Colin et al., 2024). However, its effectiveness is highly dependent on the selection of appropriate pairs, and random matching often fails to address differences in student learning styles (Graßl & Fraser, 2024). Most early definitions of pair programming focused on technical aspects without considering behavioral compatibility.

Therefore, a new approach that integrates behavioural analytics is needed to streamline this practice in the context of digital learning. These three concepts, namely learning behavior, learning analytics, and pair programming, are core elements in the framework of this study. Through the integration of these three components, a more targeted and responsive learning approach to student needs can be realized. Fig. 1 shows the relationships concept in the study.

Fig. 1: Relationships Concepts in the Study

This diagram shows that student behavior observed through Learning Management System (LMS) data is analyzed using learning analytics. The results of the analysis are used to form more suitable pairs before they participate in pair programming activities, which is ultimately expected to improve programming competence.

2.2 Focused Issue: Ineffective Pair Formation in Collaborative Programming

Pair programming has been identified as a pedagogical strategy that can increase student engagement and understanding of programming concepts. However, its effectiveness often depends on the compatibility of the pairs formed. Pair compatibility has been identified as a critical factor influencing students' satisfaction, collaboration, and learning outcomes (Satratzemi et al., 2023). Common classroom practices of student pair formation, such as random matching or based on academic achievement, have significant weaknesses (Bowman et al., 2020; Salleh et al., 2011). This approach ignores important aspects such as learning styles, levels of engagement, and learning behaviors of students. When pairs are not compatible, students tend to have difficulty collaborating effectively. This situation can lead to poor communication, unbalanced division of tasks, decreased motivation, and ultimately affect student performance in programming activities (Sun et al., 2021). Students paired with more experienced partners tend to exert less effort on assignments, feel that their partner contributed more, and spend less time in the driving role. This imbalance can lead to poorer outcomes, including lower understanding of concepts and reduced interest in computer science overall.

This issue is increasingly critical in the context of digital learning using platforms such as Learning Management System (LMS). Student log data shows significant diversity in how students access materials, participate in activities, and complete assignments (Noh et al., 2024). However, this data is often not leveraged for more strategic pedagogical purposes. This weakness creates a large gap between the potential of technology and actual practice in the classroom. Therefore, a more systematic, data-driven, and adaptive pair-matching approach is a method that can be used in pair-matching. This approach should be able to take into account student learning behaviors to increase pair compatibility, strengthen collaborative dynamics, and subsequently increase the effectiveness of pair programming in programming education. Table 2 shows the pairing formation in pair programming.

Table 1: Pairing Formation in Pair Programming

Study / Author(s)	Matching Criteria	Matching Method	Data Type	Level of Education	Pairing Outcome
Yin et al. (2025)	Collaborative interaction	AI + NLP for dialogue pairing	Dialogue transcripts	High School & University	Pairing not fixed; insights used to reflect on dialogue effectiveness
Graßl & Fraser (2023)	Gender and fairness perception	Experimental (same/different gender pairing)	Observation + survey	Undergraduate (Bachelor's)	Pairing based on gender influenced comfort but not performance
Choi (2021)	Communication style & compatibility	Observational analysis (real pair interactions)	Video + chat transcripts + productivity logs	Undergraduate (Bachelor's)	Matching pairs with compatible communication patterns led to higher output & quality
Bjorn et al. (2022)	Not explicitly matched (random/assigned)	Assigned pairing (CS1 course)	Student reflection journals + thematic analysis	Undergraduate (Bachelor's)	Assigned/random pairing sometimes caused imbalance in effort & emotional burden
Izhikevich et al. (2022)	Behavioral patterns (dominance, cooperation), group dynamics	Qualitative coding of interactions within pairs	Group discussions, audio transcripts, engagement records	Undergraduate (Bachelor's)	Behavioral interaction styles (e.g., not dominating the task) correlate with satisfaction and performance

Fan et al. (2025)	AI –Human, Human - Human	Random for human	Log interaction	Undergraduate (Bachelor's)	Human–AI not developing human-to-human collaboration skills, offering limited emotional support, risking over-reliance on AI, and occasionally producing context-insensitive or inaccurate responses.
Choi et al. (2020)	Self-regulated learning (SRL)	Clustered SRL (high-low) pairing	LMS data + survey	Undergraduate (Bachelor's)	High-SRL students guided low-SRL partners, promoting balance
Bowman et al. (2019)	Gender-based perception and bias	Controlled remote pairing (gender combinations)	Surveys, code logs, interaction analysis	Undergraduate (Bachelor's)	Gender imbalance affects collaboration perception; matching must consider equity & inclusivity
Demir (2021)	Learning styles (VARK)	Heuristic – match differing styles	Visual, Aural, Read/Write, & Kinesthetic questionnaire	Undergraduate (Bachelor's)	Mixed-style pairs (e.g., Visual + Kinesthetic) improved mutual understanding
Faja (2014)	Partner involvement + experience	Observational + self-report	Engagement logs + surveys	Undergraduate (Bachelor's)	Active students preferred similar involvement level in pair
Salleh et al. (2011)	Personality (MBTI)	Heterogeneous pairing (Introvert-Extrovert)	Personality questionnaire	Undergraduate (Bachelor's)	Heterogeneous pairs outperformed homogeneous ones in motivation and communication

Many studies have examined the effectiveness of pair programming in programming education and found that this method can improve students' conceptual understanding, motivation, and collaboration skills (Tan et al., 2024). Although these positive results are recognized, previous research findings show that the effectiveness of pair programming is not uniform across all learning contexts. One of the main factors contributing to this variation is the method of pair formation used (Wang & Zhang, 2024). Previous studies often use random matching approaches or based on academic performance, but both of these approaches do not take into account the uniqueness of students' learning styles and behaviors (Bowman et al., 2020). The effects of incompatible matching include ineffective interactions, imbalances in task sharing, and decreased student engagement. Although there are studies that recommend matching based on personality or skill level, this method is still static and less responsive to changes in student behavior in the learning process (Chai et al., 2021).

Programming remains a persistent challenge in higher education, particularly within Malaysia’s Technical and Vocational Education and Training (TVET) institutions, where students often struggle to achieve programming competency despite the growing emphasis on digital transformation and Education 4.0 initiatives. Learning analytics has emerged as a powerful, data-driven pedagogical tool with the potential to enhance learning experiences by identifying behavioral patterns through Learning Management System (LMS) data such as login frequency, material viewing time, quiz participation, and assignment submission (Alowayr, 2025; Vetrivel et al., 2025). With strong guidance based on Learning Analytics, there was a significant improvement in self-regulated learning skills, including metacognitive activities, time management, perseverance, help-seeking, and final grades (Tzimas & Demetriadis, 2024). However, existing studies predominantly employ learning analytics for performance tracking or early intervention rather than to optimize collaborative learning strategies like pair programming. This creates a critical research gap in understanding how behavioral data can inform adaptive and effective pair formation, offering opportunities to improve collaborative pedagogy and strengthen students' programming competencies.

2.3 The Proposed Conceptual Approach: Behavior-Based Pair Formation

This study proposes a new concept that integrates student learning behavior through learning analytics as the basis for pair matching in pair programming. This approach is developed to overcome the weaknesses of random or academic performance-based matching commonly used in current practice.

i. Concept Basis

This concept operates by utilizing student log data obtained from the Learning Management System (LMS). This data includes behavioral indicators such as:

- Engagement level: frequency of login, interaction time with learning materials.
- Self-Regulated Learning: task completion rate, time taken to complete tasks, material referencing activities.

These indicators are analyzed using clustering methods to classify students into several behavioral categories, for example: Highly Engaged, Self-Regulated, Low-Engaged, etc. This information is then used to form complementary (heterogeneous) student pairs based on learning style compatibility.

ii. Concept Contribution

This concept not only utilizes learning analytics to identify student behavior, but also applies the data directly in collaborative pedagogical design. With more targeted matching, this concept is expected to:

- Improve pair compatibility in pair programming.
- Strengthen collaboration dynamics and more balanced task distribution.
- Encourage more active student engagement.
- Support the achievement of better programming competencies.

iii. Relationship with Conceptual Framework

The proposed conceptual framework connects three main elements:

- Input: Student behavior data from the Learning Management System (LMS).
- Process: Behavior analysis through learning analytics and pair matching based on behavior categories.
- Output: Implementation of more effective pair programming activities and increased student competence.

This approach is in line with the principles of Education 4.0, namely the use of technology and data to personalize learning and strengthen students' digital skills in the context of TVET and higher education. Fig. 2 shows the propose conceptual behavior-based pair using learning analytics.

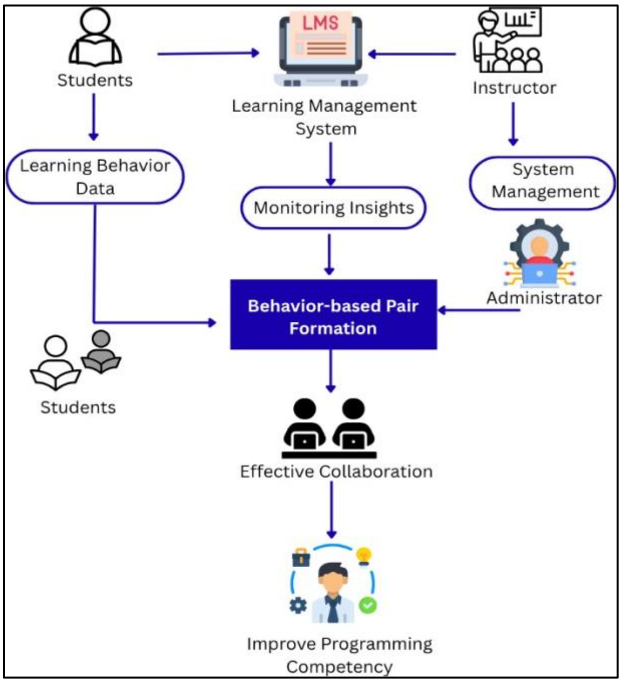

Fig. 2: **Propose Conceptual Behavior-based Pair using Learning Analytics**

iv. Potential for Integration into Learning Management System (LMS)

One of the main strengths of this behavioral-based pair matching concept is its potential for direct integration into Learning Management Systems (LMS) that are widely used in educational institutions such as Moodle, Google Classroom, Canvas and Blackboard (Heng Lim et al., 2019). This integration makes the concept practical, large-scale and user-friendly, rather than just a theoretical proposition. Table 3 show that a variety of quantitative metrics have been used to measure student learning behaviors in programming courses, particularly through data collected from Learning Management System (LMS) and programming software.

Table 3: List of Metrics to Measure Learning Behaviors

No	Learning Behavior Dimension	Metric
1	Engagement	Video Viewing Frequency, Engagement Timing, Regular Viewing Pattern, Repeated Viewing, Total Submissions, Accepted (Success Rate), Runtime Error Rate, Time Limit Exceeded Rate, Duration Between Submissions, Class Attendance, Assignment Practice, Assignment Completion Behavior, Quiz Participation, Zoom Session Attendance, Forum Participation, Total number of commits, Time Investment in Tasks, Number of Attempts, Number of Interaction Events, Total Clicks, Total Sessions, Continuously Active, In Class Engagement, events
2	Self-Regulation Learning	Reviewing Correct Questions, Filtering Questions, Reflecting on Learning, Reviewing Exam, Behavioral Consistency Over Time, Resolution Time of Compilation Errors, Compile Invocation Frequency, Self-efficacy, Undo/Remove Actions, Add/Change Statements, Generate Getters/Setters, Watch Variable Use, Task Completion Behavior, Persistence in Solving Problems, Typing and Code Editing Behavior, Impasse Behavior, Activity Metrics, Error Patterns, Session Time / Last Access, Assessment Metrics, Self-Regulation Indicator, Help-seeking, Self-talk

All of these metrics serve as indirect indicators (proxies) of behaviors such as engagement and self-directed learning. However, the study also shows that there is a diversity of definitions and uses of metrics, which requires attention to the validity and uniformity of measurement. All of these metrics serve as indirect indicators (proxies) of behaviors such as engagement, self-directed learning, collaboration, and motivation. However, the study also shows that there is a diversity of definitions and uses of metrics, which requires attention to the validity and uniformity of measurement.

3. Methodology

This study adopts a quasi-experimental design involving two groups of students: an experimental group and a control group. Participants will consist of 80 diploma-level students enrolled in a programming course at a Malaysian polytechnic institution. A purposive sampling method will be applied to select participants with comparable academic backgrounds to ensure homogeneity across groups. The experimental group will undergo behaviour-based pair programming sessions, where pair formation is determined by clustering students according to their engagement and self-regulated learning (SRL) behaviours extracted from Learning Management System (LMS) data. In contrast, the control group will follow conventional pair programming with random pairing. The intervention will be conducted over six pair programming sessions, each lasting approximately 90 minutes, integrated into regular programming lab activities. Programming competency will be measured using pre-test and post-test assessments that have undergone expert validation to ensure content relevance and reliability. Data will be analysed using descriptive statistics and paired t-tests to evaluate performance differences between groups and determine the effectiveness of behaviour-based pair programming in enhancing programming competency. Data will be collected from the institution's Learning Management System (LMS) throughout the semester, focusing on two key learning behaviours: engagement and self-regulated learning (SRL). Indicators of engagement include login frequency, time spent accessing learning materials, participation in discussion forums, and patterns of resource access, while self-regulated learning (SRL) will be measured through the timeliness of assignment submissions, frequency of revisiting materials, and participation in self-assessment activities.

The behavioural data will be processed using Learning Analytics techniques to quantify each indicator, followed by analysis through a clustering algorithm such as K-Means to identify students with similar behavioural profiles. For the experimental group, student pairs will be formed heterogeneously within the behavioural cluster, while the control group will be formed through random pairing without consideration of behavioural profiles. Both groups will participate in structured pair programming sessions in which each pair works collaboratively on programming tasks, alternating roles as "driver" and "navigator" at regular intervals to ensure balanced participation.

The effectiveness of the intervention will be measured through a pre-test and post-test assessing programming competency in terms of code correctness, efficiency, and readability. The collected data will be analysed using descriptive statistics to summarise behavioural profiles, followed by a paired t-test to examine within-group performance differences before and after the intervention, and an independent samples t-test to compare the effectiveness between the experimental and control groups.

4. Analysis

4.1 Data Analysis Approach

To evaluate the preliminary effectiveness of the Behavior-Based Learning Analytics in Pair Programming (BBLA-PP) framework, a pilot study was conducted involving 26 diploma students enrolled in an Object-Oriented Programming (OOP) course at Politeknik Ungku Omar. Pre- and post-tests on programming competency were administered, while behavioral data (engagement and self-regulated learning) were extracted from the institutional Learning Management System (CIDOS). Descriptive and inferential analyses were performed using Jamovi. Descriptive statistics summarized changes in mean scores, while paired-sample t-tests assessed the significance of improvement in programming competency. Effect sizes (Cohen's d) were computed to measure the magnitude of the observed change, following guidelines by Cohen (1988). A Wilcoxon signed-rank test indicated that post-test scores (Median = 5.0) were significantly higher than pre-test scores (Median = 3.5), $Z = -3.12$, $p < .05$. Table 4 shows comparison of students' programming competency before and after intervention and Fig. 3 illustrates the mean and median scores for pre-test and post-test conditions.

Table 4: Comparison of students' programming competency before and after intervention.

Test Type	N	Mean	SD	t(df)	p	Cohen's d	Interpretation
Pre-Test	26	3.42	1.63				
Post-Test	26	4.15	1.54	3.18 (25)	< .01	0.62	Moderate – large effect

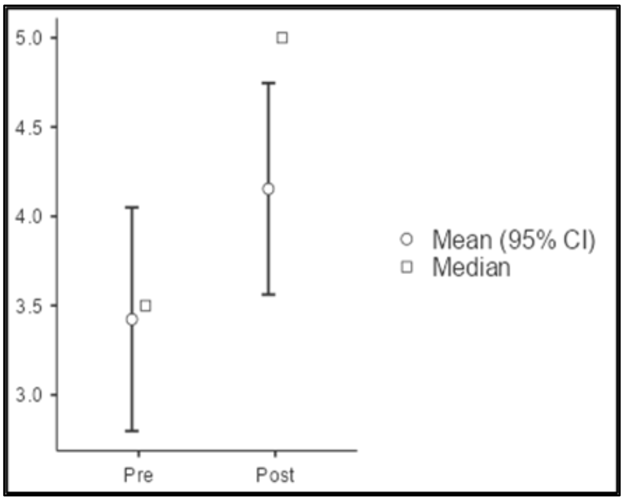

Fig. 3: **Mean and Median Comparison Plot with 95% Confidence Intervals**

There was an observable increase in both mean and median values after the intervention, suggesting that students' performance improved. The confidence intervals of the post-test mean did not overlap substantially with those of the pre-test, implying a statistically meaningful improvement. A Wilcoxon signed-rank test was conducted to compare students' programming performance scores before and after the intervention. Results indicated a statistically significant increase in post-test scores ($Z = -3.12$, $p = .002$, $r = 0.61$), suggesting that the intervention had a large positive effect on students' performance. The median score increased from 3.5 (pre-test) to 5.0 (post-test), indicating that most students improved their programming competency after the learning session.

Although these findings are preliminary, it provides empirical validation for the conceptual model proposed in this paper. Future large-scale studies should examine the stability of these effects across multiple cohorts and courses, and explore alternative clustering techniques to refine pair formation strategies. Furthermore, integrating engagement dashboards and real-time feedback mechanisms into the LMS may extend the practical utility of the BBLA-PP model in Malaysian TVET institutions and other higher-education contexts aligned with Education 4.0.

5. Discussion

5.1 Practical Significance and Educational Interpretation

Beyond statistical significance, the effect size denotes that behaviour-based pairing exerts a practically meaningful influence on programming competency. The observed improvement corresponds to a transition from surface-level code comprehension to more structured object-oriented design skills. Students exhibited higher engagement in collaborative debugging, greater self-monitoring, and stronger confidence when explaining coding logic to peers. These results also reinforce the Zone of Proximal Development (ZPD) principle, showing that pairing students with similar or complementary behavioural profiles promotes peer scaffolding and mutual regulation. In practice, this implies that educators can use Learning Analytics data (e.g., login frequency, timely submissions, quiz engagement) to automatically form pairs that optimise collaborative synergy, rather than relying on random assignment.

5.2 The Importance and Contributions to Knowledge

The new concept proposed in this study, namely the formation of pair programming pairs based on learning behavior through learning analytics, is an innovative and responsive approach to the challenges of 21st century education. In the increasingly complex era of digital learning, teaching strategies need to be adapted not only based on curriculum content, but also on student interaction, engagement and motivation patterns that can be detected through data. This is in line with the main goal of Dasar e-Pembelajaran Negara (DePAN) which is to use information and communication technology as a tool to improve the quality of teaching and learning to develop world-class human capital.

Learning analytics can be used to identify student behavior through Learning Management System (LMS) log data such as login frequency, time spent on learning materials, participation in quizzes and assignment submission. This concept allows lecturers to form student pairs based on compatibility or complementarity of behavior. This smart matching is expected to solve the issue of mismatch of pairs which is often the main cause of failure in the implementation of pair programming. In terms of contribution to knowledge, this concept extends the application of learning analytics beyond simply monitoring student performance to strategic support in collaborative pedagogical design. It also connects three key areas, namely learning behavior, collaborative programming and educational analytics, in a mutually reinforcing conceptual framework. This study also responds to the call for digital education transformation as outlined in the Digital Education Action Plan and TVET Education Policy, which emphasizes the need for the integration of smart technologies and student-centered learning practices.

The proposed concept extends the application of learning analytics beyond performance monitoring to adaptive pedagogical design. It brings together three key areas pair programming, learning behavior, and learning analytics in a mutually reinforcing conceptual framework. This approach is also in line with the principles of Education 4.0 and the digital transformation of TVET, which emphasize data-driven learning and personalized educational practices (UNESCO, 2022; World Economic Forum, 2020). In addition, this concept introduces the idea that pair compatibility is not only based on skill level or personality, but can also be objectively measured through student behavior patterns derived from Learning Management System (LMS) data (Pecuchova & Drlik, 2024). This is a new dimension that has rarely been explored in previous pair programming research, and it opens up new opportunities in strengthening collaborative learning theory.

Although not yet empirically tested, this concept offers a solid foundation for further research in the form of experiments, the development of data-based matching systems, and the evaluation of impact on student achievement. Its broad potential for application, including in the context of polytechnics, universities and TVET, makes this contribution significant in the current digital pedagogy research landscape (Akçapınar et al., 2024).

5.3 Practical Implications

From a practical perspective, this concept has the potential to be directly integrated into Leaning Management System (LMS) such as Moodle, Google Classroom, Canvas or Blackboard through additional analytics modules. This integration allows for automatic and large-scale pair matching, without increasing the workload of lecturers. In addition, monitoring dashboards can be developed to help lecturers identify pairs that require early intervention, thus supporting more targeted teaching practices. This concept can also form the basis for the development of artificial intelligence (AI) algorithms for more dynamic pair matching. Machine learning algorithms can be used to refine the student behavior classification model and improve the matching recommendations over time based on new data (Sai Sharvesh et al., 2023). From a practical perspective, this research will provide a data-driven pairing method that can be implemented by educators using readily available Learning Management System (LMS) data. By forming pairs based on homogeneous behavioural clusters, the approach aims to improve the effectiveness of pair programming sessions, minimise mismatched working styles, and encourage balanced participation between partners. Moreover, the study will deliver clustering-based tools and detailed guidelines that instructors can use to optimise student pairing without relying solely on intuition or subjective judgement. These practical outcomes have the potential to enhance teaching efficiency and student collaboration in real classroom settings.

Although these findings are preliminary, they provide empirical validation for the conceptual model proposed in this paper. Future large-scale studies should examine the stability of these effects across multiple cohorts and courses, and explore alternative clustering techniques (e.g., hierarchical or fuzzy clustering) to refine pair formation strategies. Furthermore, integrating engagement dashboards and real-time feedback mechanisms into the LMS may extend the practical utility of the BBLA-PP model in Malaysian TVET institutions and other higher-education contexts aligned with Education 4.0.

5.4 Policy and Strategic Contributions

At the policy level, the findings of this study are aligned with national and institutional strategies, particularly the Malaysia Education Blueprint (Higher Education) 2015–2025, which underscores the importance of producing digitally competent graduates. The proposed approach also supports the TVET transformation agenda by promoting innovative, analytics-based pedagogical practices that are scalable and adaptable across diverse institutional contexts. Furthermore, by demonstrating the value of Learning Analytics in guiding collaborative learning strategies, this research offers policymakers concrete evidence to consider integrating such data-driven approaches into broader digital transformation initiatives in education.

6. Future Research

The concept of behavior-based pair matching through learning analytics proposed in this paper provides a new data-driven approach to improving the effectiveness of pair programming in programming education. With the support of technology, this concept can not only optimize student collaboration, but also contribute to improving the programming competence and marketability of graduates in the digital era (World Economic Forum, 2020; UNESCO, 2022). Mastery of programming skills is a core skill in the era of Industrial Revolution 4.0 and Education 4.0, yet it remains one of the most challenging learning outcomes to achieve at the tertiary level, especially in polytechnics and TVET institutions in Malaysia (Watson & Li, 2014; Bennedsen & Caspersen, 2019). Although the pair programming method has been proven to have the potential to improve student achievement, its effectiveness is highly dependent on the compatibility of student pairs — a factor that is often overlooked in conventional pair formation methods (Salleh et al., 2011). This concept paper proposes an innovative integration between Learning Analytics and pair programming, using learning behavior indicators such as engagement and self-regulated learning (SRL) to form homogeneous pairs through clustering techniques (Ifenthaler & Yau, 2020). This approach is expected to create more balanced collaboration, reduce collaboration conflicts, and maximize each member's contribution to the programming task. The proposed conceptual framework not only fills a critical gap in the literature, namely the lack of a behaviorally-based pair-building model, but also offers a practical and scalable solution for educators. If implemented effectively, this approach has the potential to improve students' programming competencies, support national digital education strategies, and contribute to the global discussion on data-driven instructional design (De Witte & Chenier, 2023). Ultimately, this research envisions a shift from random or skill-based pair-building to a new paradigm driven by behavior and supported by learning analytics, which can be applied across a variety of disciplines and educational contexts.

Acknowledgment
The authors would like to express their deepest appreciation to Universiti Teknologi PETRONAS (UTP) for providing academic guidance and research support throughout this study. Special thanks are also extended to Politeknik Ungku Omar (PUO) and the Department of Polytechnic and Community College Education (JPPKK), Malaysia, for their cooperation in facilitating access to students and learning data for the pilot study. The authors also acknowledge the invaluable supervision of Dr. Nur Syadhila binti Che Lah, whose insights greatly contributed to the conceptual and methodological development of this paper.

References

Alowayr, A. (2025). Learning analytics systems to improve the quality of students' outcomes. *International Journal for Quality Research*, 19(1), 297–312. doi: 10.24874/IJQR19.01-19

Akçapınar, G., López-Pernas, S., Er, E., & Saqr, M. (2024). How a Learning Analytics Dashboard Intervention Influences the Dynamics of Students' Learning Behavior. In J. A. D. C. Gonçalves, J. L. S. D. M. Lima, J. P. Coelho, F. J. García-Peñalvo, & A. García-Holgado (Eds.), *Proceedings of TEEM 2023* (pp. 810–819). Springer Nature Singapore. doi: 10.1007/978-981-97-1814-6_79

Bennedsen, J., & Caspersen, M. (2019). Failure rates in introductory programming: 12 years later. *ACM Inroads*, 10(2), 30–36. doi: 10.1145/3324888

Bjorn, C., Haglund, P., Munz, K., & Stromback, F. (2022). It's okay because I worked really hard! – Student justifications for questionable collaboration while solving computer labs. *Proceedings of the 2022 IEEE Frontiers in Education Conference (FIE)*, 1–9. doi: 10.1109/FIE56618.2022.9962546

Bowman, N. A., Jarratt, L., Culver, K. C., & Segre, A. M. (2020). Pair programming in perspective: Effects on persistence, achievement, and equity in computer science. *Journal of Research on Educational Effectiveness, 13*(4), 731–758. doi: 10.1080/19345747.2020.1799464

Chai, S. S., Goh, K. L., Wang, H. H., & Wee, B. L. (2021). Incorporating unstructured text in multi-layer perceptron (MLP) network: Factors affecting partner selection in pair programming. *Proceedings of the 3rd IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET 2021).* doi: 10.1109/IICAIET51634.2021.9573795

Chigona, W., & Pollock, M. (2008). Pair programming for information systems students new to programming: Students' experiences and teachers' challenges. *Proceedings of the Portland International Conference on Management of Engineering & Technology (PICMET)*, 1587–1594. doi: 10.1109/PICMET.2008.4599777

Clow, D. (2012). The learning analytics cycle: Closing the loop effectively. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge,* 134–138. doi: 10.1145/2330601.2330636

Colin, J., Hoarau, S., Declercq, C., & Broisin, J. (2024). Design and evaluation of a web-based distributed pair programming tool for novice programmers. *Proceedings of the 2024 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)* (pp. 527–533). doi: 10.1145/3649217.3653571

De Witte, K., & Chenier, M.-A. (2023). Learning analytics in education for the twenty-first century. In E. Bertoni, M. Fontana, L. Gabrielli, S. Signorelli, & M. Vespe (Eds.), *Handbook of Computational Social Science for Policy* (pp. 305–326). Springer International Publishing. doi: 10.1007/978-3-031-16624-2_16

Demir, Ö., & Seferoglu, S. S. (2021). The effect of determining pair programming groups according to various individual difference variables on group compatibility, flow, and coding performance. *Journal of Educational Computing Research, 59*(1), 41–70. doi: 10.1177/0735633120949787

Duvall, S., Spurlock, S., Hutchings, D. R., & Duvall, R. C. (2021). Improving content learning and student perceptions in CS1 with Scrumage. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education,* 474–480. doi: 10.1145/3408877.3432415

Faja, S. (2014). Evaluating effectiveness of pair programming as a teaching tool in programming courses. *Information Systems Education Journal, 12*(6), 39–49.

Fan, G., Liu, D., Zhang, R., & Pan, L. (2025). The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance: A comparative study with traditional pair programming and individual approaches. *International Journal of STEM Education, 12*(1), 16. doi: 10.1186/s40594-025-00537-3

Figueiredo, J., & García-Peñalvo, F. J. (2024). Design science research applied to difficulties of teaching and learning initial programming. *Universal Access in the Information Society, 23*(3), 1151–1161. doi: 10.1007/s10209-022-00941-4

Graßl, I., & Fraser, G. (2024). Equitable student collaboration in pair programming. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education,* 274–285. doi: 10.1145/3639474.3640086

Heng Lim, E., Sangodiah, A., Muniandy, M., Isawasan, P., Koo Yuen, P., & Al Nallusamy, S. (2019). Integration of teaching taxonomy into personalized learning management system using Felder-Silverman model. *Proceedings of the 2019 IEEE International Conference on Cybernetics and Intelligent Systems,* 29–33. doi: 10.1109/CSUDET47057.2019.9214765

Ifenthaler, D., & Yau, J. Y.-K. (2020). Utilising learning analytics to support study success in higher education: A systematic review. *Educational Technology Research and Development, 68*(4), 1961–1990. doi: 10.1007/s11423-020-09788-z

Izhikevich, K., Ong, K., & Alvarado, C. (2022). Exploring group dynamics in a group-structured computing undergraduate research experience. *Proceedings of the 2022 ACM Conference on International Computing Education Research (ICER),* 135–148. doi: 10.1145/3501385.3543959

Joshi, Y., Mallibhat, K., & M., V. (2022). Students' performance prediction using multimodal machine learning. *Proceedings of the 2022 IEEE IFEES World Engineering Education Forum – Global Engineering Deans Council (WEEF-GEDC),* 1–5. doi: 10.1109/WEEF-GEDC54384.2022.9996212

Karimov, A., Saarela, M., Aliyev, S., & Baker, R. S. (2025). Ethical considerations and student perceptions of engagement data in learning analytics. *Proceedings of the Annual Hawaii International Conference on System Sciences (HICSS),* 4757–4766. doi: 10.24251/hicss.2025.572

Leow, W. M. F., & Huang, W. (2021). Exploring the effectiveness of pair programming in developing students' computational thinking skills through Scratch. *Proceedings of the 2021 International Conference on Computer Education and Instructional Technology,* 2–7.

Noh, Z. M., Yatim, N. M., Saad, W. H. M., Hamzah, R. A., Radzi, S. A., & Aziz, N. H. A. (2024). Analysis of student engagement towards learning materials in learning management system: A case study. *Proceedings of the 2024 International Conference on Technology and Education (ICTeD)* (pp. 86–91). IEEE. doi: 10.1109/ICTeD62334.2024.10844667

Paulsen, L., & Lindsay, E. (2024). Learning analytics dashboards are increasingly becoming about learning and not just analytics—A systematic review. *Education and Information Technologies*, *29*(11), 14279–14308. doi: 10.1007/s10639-023-12401-4

Pecuchova, J., & Drlik, M. (2024). Enhancing the early student dropout prediction model through clustering analysis of students' digital traces. *IEEE Access, 12,* 159336–159367. doi: 10.1109/ACCESS.2024.3486762

Roque Hernández, R. V., Guerra Moya, S. A., & Rico, F. C. C. (2021). Acceptance and assessment in student pair-programming: A case study. *International Journal of Emerging Technologies in Learning (iJET), 16*(9), 4–18. doi: 10.3991/ijet.v16i09.18693

Sai Sharvesh, R., Suresh Kumar, K., Kumar, D. D., & Sabthagiri, P. (2023). Course recommendation system using Python and Streamlit. *Proceedings of the International Conference on Recent Trends and Advances in Computing (ICRTAC)* (pp. 548–552). doi: 10.1109/ICRTAC59277.2023.10480866

Salleh, N., Mendes, E., & Grundy, J. (2011). Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review. *IEEE Transactions on Software Engineering, 37*(4), 509–525. doi: 10.1109/TSE.2010.59

Satratzemi, M., Stelios, X., & Tsompanoudi, D. (2023). Distributed Pair Programming in Higher Education: A Systematic Literature Review. *Journal of Educational Computing Research*, *61*(3), 546–577. doi: 10.1177/07356331221122884

Simon, Luxton-Reilly, A., Ajanovski, V., Fouh, E., Gonsalvez, C., Leinonen, J., Parkinson, J., Poole, M., & Thota, N. (2019). Pass rates in STEM disciplines including computing. *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education,* 267–268. doi: 10.1145/3304221.3325532

Sun, D., Ouyang, F., Li, Y., & Chen, H. (2021). Three contrasting pairs' collaborative programming processes in China's secondary education. *Journal of Educational Computing Research, 59*(4), 740–762. doi: 10.1177/0735633120973430

Tan, J., Wu, L., & Ma, S. (2024). Collaborative dialogue patterns of pair programming and their impact on programming self-efficacy and coding performance. *British Journal of Educational Technology, 55*(3), 1060–1081. doi: 10.1111/bjet.13412

Tuveri, M., Zurru, A., Fadda, D., & Saba, M. (2022). Online learning mediated by social teaching platforms: An experience from a flipped undergraduate physics course in renewable energies. *European Journal of Physics, 43*(5), 055703. doi: 10.1088/1361-6404/ac78a6

Tzimas, D. E., & Demetriadis, S. N. (2024). Impact of Learning Analytics Guidance on Student Self-Regulated Learning Skills, Performance, and Satisfaction: A Mixed Methods Study. *Education Sciences*, *14*(1), 92. doi: 10.3390/educsci14010092

Vetrivel, S. C., Vidhyapriya, P., & Arun, V. P. (2025). Data science for learning analytics: Understanding and improving learning processes. In *Driving Quality Education Through AI and Data Science* (pp. 409–432). IGI Global. doi: 10.4018/979-8-3693-8292-9.ch018

Vinueza-Morales, M., Rodas-Silva, J., & Vidal-Silva, C. (2025). Teaching programming in higher education: Analyzing trends, technologies, and pedagogical approaches through a bibliometric lens. *International Journal of Advanced Computer Science and Applications, 16*(1), 61–68. doi: 10.14569/IJACSA.2025.0160106

Wang, Q., & Mousavi, A. (2023). Which log variables significantly predict academic achievement? A systematic review and meta-analysis. *British Journal of Educational Technology, 54*(1), 142–191. doi: 10.1111/bjet.13282

Wang, Y., & Zhang, R. (2024). Research on the pair programming partner recommendation method based on personalized learning features. In J. Gan, Y. Pan, J. Zhou, D. Liu, X. Song, & Z. Lu (Eds.), *Computer Science and Educational Informatization* (Vol. 1900, pp. 342–357). Springer Nature Singapore. doi: 10.1007/978-981-99-9492-2_29

Watson, C., & Li, F. W. B. (2014). Failure rates in introductory programming revisited. *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education (ITiCSE '14),* 39–44. doi: 10.1145/2591708.2591749

WEF. (2020, June). *Future of jobs 2020.* UNESCO-UNEVOC. https://unevoc.unesco.org/pub/2020%20June%20IAG.pdf

Yin, S. X., Liu, Z., Goh, D. H.-L., Quek, C. L., & Chen, N. F. (2025). Scaling up collaborative dialogue analysis: An AI-driven approach to understanding dialogue patterns in computational thinking education. *Proceedings of the 15th International Learning Analytics and Knowledge Conference,* 47–57. doi: 10.1145/3706468.3706474

Zhan, C., Joksimovic, S., Ladjal, D., Rakotoarivelo, T., Marshall, R., & Pardo, A. (2024). Preserving both privacy and utility in learning analytics. *IEEE Transactions on Learning Technologies, 17,* 1655–1667. doi: 10.1109/TLT.2024.3393766

Zhu, M., & Ergulec, F. (2023). A review of collaborative assessment strategies in online learning. *Distance Education, 44*(3), 522–543. doi: 10.1080/01587919.2022.2150127

Zima, H. P. (2007). From FORTRAN 77 to locality-aware high productivity languages for peta-scale computing. *Scientific Programming, 15*(1), 45–65. doi: 10.1155/2007/219061