Journal of Information Technology Education: Research

An Official Publication of the Informing Science Institute InformingScience.org

JITEResearch.org

Volume 21, 2022

EXPLORING THE INFLUENCE OF STUDENTS' MODES OF BEHAVIORAL ENGAGEMENT IN AN ONLINE PROGRAMMING COURSE USING THE PARTIAL LEAST SQUARES STRUCTURAL EQUATION MODELING APPROACH

Nien-Lin Hsueh	Department of Information Engineering and Computer Science, Feng Chia University, Taichung, 407 Taiwan	nlhsueh@fcu.edu.tw
Bilegjargal Daramsenge*	Department of Information Engineering and Computer Science, Feng Chia University, Taichung, 407 Taiwan	P0731809@o365.fcu.edu.tw
Lien-Chi Lai	Department of Information Engineering and Computer Science, Feng Chia University, Taichung, 407 Taiwan	P1000433@mail.fcu.edu.tw

* Corresponding author

ABSTRACT

Aim/Purpose	The goal of this study was twofold: first, to examine how learners' behavioral engagement types affect their final grades in an online programming course; and second, to explore which factors most strongly affect student performance in an online programming course and their connection to the types of cognitive engagement.
Background	During the COVID-19 pandemic situation, information technology educational methods and teaching have been transforming rapidly into online or blended. In this situation, students learn course content through digital learning management systems (LMSs), and the behavioral data derived from students' interactions with these digital systems is important for instructors and researchers. However, LMSs have some limitations. For computer science students, the traditional learning management system is not enough because the coding behavior cannot be analyzed. Through the OpenEdu platform, we collected log data

Accepting Editor Aaron M. Glassman │Received: May 17, 2022│ Revised: July 17, 2022 │ Accepted: August 4, 2022.

Cite as: Hsueh, N-L., Daramsenge, B., & Lai, L-C. (2022). Exploring the influence of students' modes of behavioral engagement in an online programming course using the partial least squares structural equation modeling approach. *Journal of Information Technology Education: Research, 21,* 403-423. https://doi.org/10.28945/5010

from 217 undergraduates enrolled in a Python programming course offered by Feng Chia University in Taiwan in the spring semester of 2021.

Methodology We applied the evaluation framework of learning behavioral engagement conducted on a massive open online course (MOOC) platform and integrated it with the partial least squares structural equation modeling (PLS-SEM) approach. PLS-SEM is widely used in academic research and is appropriate for causal models and small sample sizes. Therefore, this kind of analysis is consistent with the purpose of our study.

Contribution In today's fast-paced world of information technology, online learning is becoming an important form of learning around the world. Especially in computer science, programming courses teach many skills, such as problem-solving, teamwork, and creative thinking. Our study contributes to the understanding of how behavioral engagements in distance programming learning affect student achievement directly and through cognitive engagement. The results can serve as a reference for practitioners of distance programming education.

Findings Our results demonstrate that: (1) online time and video-watching constructs had significant effects on the self-assessment construct, self-assessment and video-watching constructs had significant effects on the final grade construct, and online document reading was not a significant factor in both self-assessments and final grades; (2) video watching had a most significant effect than other behavioral constructs in an online programming course; (3) cognitive engagement types are inextricably linked to the development of a behavioral engagement framework for online programming learning. The mediation analysis and the importance-performance map analysis supported the importance of cognitive engagement.

Recommendations for Practitioners (1) Online education platform developers and university policymakers should pay close attention to the development of self-assessment systems and design such systems based on students' cognitive skills. (2) Instructors are advised to put substantial effort into the creation of videos for each course session and to actively promote students' interest in the course material.

Recommendations for Researchers The empirical results reported in this study allow a better understanding of the connection between behavioral engagement and final achievement. However, there are still great challenges in trying to explore more kinds of engagement, like emotional or social engagement. It would be interesting to deepen the results obtained by integrating programming behavior like debugging and testing.

Impact on Society Online programming courses allow students to improve their coding skills and computer science background. Students' behavioral engagement strongly affects their academic achievement, their ability to complete a course successfully, and the quality of the learning process. Our work can encourage more people who are different majors in society to learn coding in an online environment even not only computer science students. Moreover, the findings of this study can be recommendations for understanding students' learning behavior and the development of distance programming learning.

Future Research We suggest for future studies: (1) include a wider range of participants, such as students enrolled in MOOCs environments; (2) include more log data items that can express various students' behavior, depending on the reliability and validity of the research model; and (3) conduct more detailed studies of the effects of emotional engagement as well as additional aspects of students' social engagement to elucidate the factors affecting students' behavioral participation and performance more thoroughly.

Keywords behavioral engagement, cognitive engagement, programming education, distance learning, PLS-SEM

INTRODUCTION

With the development of information technology and the renewal of modern educational methods, education and teaching have been transformed (Shi & Weng, 2021). The environments in which students learn are gradually changing from traditional classroom environments to online or blended environments (Hu et al., 2019). Most students' first exposure to course material occurs outside the class. Students are more frequently experiencing course content through digital learning management systems (LMSs), and the collection of behavioral data derived from students' interactions with these digital systems is simple (Battestilli et al., 2020). LMSs provide multiple advantages for teaching and learning. First, LMSs support teachers in conducting a variety of in- and after-class activities, such as review of learning materials, asynchronous discussion, quizzes, and self-assessment or peer assessment. Second, each student can freely select their course materials and control their learning path and pace (Li & Tsai, 2017).

However, LMSs have some limitations, such as for computer science students, the traditional learning management system is not enough because the coding behavior cannot be analyzed. Students often use online judge (OJ) systems during their out-of-class time. Most OJ systems can only determine whether a student is obtaining low scores and cannot determine why, which can cause students to lose confidence in programming (Wu et al., 2021). OJ is an automatic programming assignment grading system. The judge also allows us to archive past problems, solutions, and student submissions easily. As well as the effective reuse of past problems and solutions, analysis of students' solutions can help in making the course materials more effective (Kurnia et al., 2001).

Exploring students' learning behavior by using log data can provide definite results. Student activity logs are a key resource for gaining insights into student behavior in online environments. Observing students' behavior patterns is necessary for detecting their learning styles (Estacio & Raga, 2017). The partial least squares structural equation modeling (PLS-SEM) approach is widely used in academic research and is appropriate for causal models and small sample sizes (Hair et al., 2011). Therefore, we proposed to explore the following questions:

- What is the direct effect of students' engagement types on their final grades in an online programming course?
- What is the mediating role of cognitive engagement on the relationship between behavioral engagement types and final grades in an online programming course?
- Which factors most strongly affect students' performance in an online programming course?

We applied the evaluation framework of learning behavioral engagement based on the MOOC platform proposed by G. Sun and Bin (2018) and integrated it with the PLS-SEM approach. Our analysis is conducted from a Python programming course offered by Feng Chia University in Taiwan in the spring semester of 2021.

The remainder of this paper is organized as follows. The next section provides a discussion of previous studies investigating students' behavioral and cognitive engagements in E-learning. The following two sections present the research background and methodology. Next, the results of our data analysis, relevant discussions, and implications of our findings are presented. Finally, we describe the study's limitations, recommendations for future research, and conclusions.

LITERATURE REVIEW

In this section, we will introduce some general research on learning engagements from the aspect of video watching behavior, learning materials, and self-assessment. Some research work specifically related to our hypotheses will be introduced in the hypothesis section.

Video watching behavior: Cummins et al. (2015) described the construction and instrumentation of an Interactive Video Lecture Platform to measure student engagement with in-video quizzes. The results from this investigation demonstrated that in-video quizzes were successful in creating an engaging and interactive mode of content delivery and student engagement with in-video questions was consistently high (71%-86%). Picardo et al. (2021) investigated the common lecture recording viewing behaviors of students and the relationship between lecture recording viewing and academic performance in a first-year programming course. A significant positive correlation between lecture recording views and final grades was identified. Students who repeated the course after failing it once, achieved, on average, higher grades if they had more lecture recording views in their second attempt.

Learning materials: Othman et al. (2013) noted that students prefer to work in small groups to enhance their understanding of programming and they also prefer to search for learning materials from various Internet resources such as e-learning portals. Isomöttönen and Tirronen (2016) concluded that independent study can be and was facilitated by adding quizzes, feedback tools, and embedded coding boxes in the materials and by improving learning materials overall. Learning materials consisted of web resources, specifically particular relevant e-books.

Self-assessments: Ngai et al. (2010) presented how self-assessment practices can be successfully integrated into an introductory remedial programming course to improve the learning experience of students. Results showed that, given the proper direction and feedback, students can assess themselves fairly and objectively. Students had an excellent overall experience in the course through the self-assessment. Lepp et al. (2017) described an experience in the creation of quizzes, programming exercises, and tests for automated feedback, self-assessment questions, and troubleshooters. The results indicated that self-assessment questions and explanations of self-assessment answers are of importance and using self-assessment tools reduces the number of letters/questions to instructors.

BEHAVIORAL ENGAGEMENT ANALYSIS IN E-LEARNING

Several definitions of behavioral engagement have been proposed. Deng et al. (2019) noted that behavioral engagement is traditionally conceptualized as students' participation in classroom learning and academic activities, such as video use and in-video interactions. Bergdahl et al. (2020) defined behavioral engagement as students' active participation, involvement, and persistence in a learning activity. T. K. Chiu (2022) defined behavioral engagement as the degree of involvement of students in learning activities in terms of attention, participation, effort, intensity, and persistence.

Some previous studies have investigated the role of behavioral engagement in online and blended programming courses. Watson et al. (2013) presented a new approach for predicting students' performance in a programming course based on the analysis of directly logged data, which reflects various aspects of the students' typical programming behavior. Programming behavior was directly logged by using an extension for the BlueJ IDE. An evaluation of log data from a sample of 45 programming students revealed that a student's analytical approach was an excellent early predictor of their performance, explaining 42.49% of the variance in students' coursework scores.

Wang (2017) explored how online behavior engagement affects achievement in flipped classrooms with a problem-centered learning flow consisting of activation, demonstration, application, and integration. All the courses were conducted on the Moodle platform, and a total of 488 undergraduate students enrolled at a university in Taiwan from 2010 to 2015 participated in the courses. The findings demonstrated that engagement in problem-solving activities exerts a significant effect on achievement. Furthermore, engagement in self-assessment and self-reflection activities exerts a significant direct effect on engagement in online classes and social interaction.

Xie et al. (2021) conducted learning behavior analysis and student performance prediction based on data from students' behavior logs from three consecutive years of a college-level Java Language Programming course conducted in a blended format. Researchers revealed that, according to the fine-grained results of feature selection and correlation analysis, the learning features that have a more significant impact on the students' final grades are selected from their learning behaviors for analysis, which is conducive to teachers' individualized teaching.

COURSE DESIGN AND RESEARCH MODEL

COURSE DESCRIPTION AND PARTICIPANTS

We analyzed data from a Python programming course offered by the College of Information and Electrical Engineering at Feng Chia University in Taiwan. The goal of the course was to teach students basic Python programming, including computation, logic, collection objects, functions, and basic panda-based analysis methods. The course was conducted in a blended format, which comprised 2 hours of online study anytime and anywhere and 1 hour of in-class discussion each week of the semester. To make online learning more interactive, we designed several types of learning materials, such as quiz-in-video activities that enabled the students to quickly refresh their learning, Trinket online workshops, Google Slide flashcards, and Game-test assessment. The course won an Outstanding MOOC Award from Taiwan's Ministry of Education for its well-designed curriculum and application of technology.

Initially, 315 students were registered. After inactive students without behavioral log data on the system were removed, we analyzed the log data of 217 students. Most (177; 82%) of the students were Information Engineering and Computer Science majors; the others (40; 18%) were majoring in Materials Science, Applied Mathematics, Automatic Control Engineering, Chemical Engineering, Business Administration, Aerospace and Systems Engineering, Civil Engineering, Statistics, and Electrical Engineering.

IDENTIFYING CONSTRUCTS

A structural equation model with latent constructs has two components. The first component is measurement models that include the unidirectional predictive relationships between each latent construct and its associated observed indicators. The second component is a structural model which illustrates the relationships (paths) between latent constructs (Hair et al., 2011).

In this study, we assessed students' engagement types based on learning behavior indicators derived from LMS log data. Herein, we propose a model for measuring students' behavioral engagement in online programming courses and discuss its connections to types of cognitive engagement, drawing on the evaluation framework proposed by G. Sun and Bin (2018). Within this framework, behavioral engagement is evaluated in terms of different dimensions, learning activities, and logs. Most importantly, the logs included in this framework are as follows: (1) learning time, (2) learning interval and regularity, (3) depth and length of learning notes, (4) after-school test scores, (5) number of questions, answers, and topics recommended by learners, (6) interactive display times, (7) timely assessment of time and time score, and (8) courseware video-on-demand (micro video and interactive electronic teaching material) score. Some researchers have used this framework as a reference; for example, Y. Sun & Chai (2020) established a multidimensional active engagement model for online learning and used the framework to measure the degree of the active participation and interaction of learners in an online learning environment. Figure 1 presents our proposed research model.

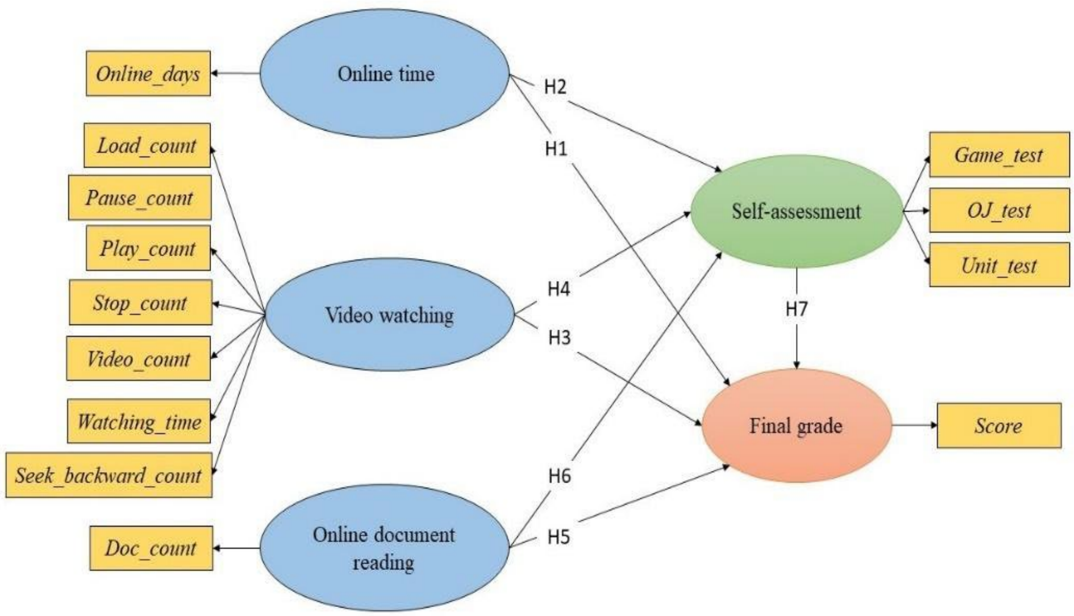


Figure 1. Research model

Note: H1–H7 are the proposed hypotheses; self-assessment is a mediator construct (variable)

Our research model consists of 13 indicators and five constructs, connected through seven hypothetical paths. The online time construct includes the single indicator *Online_days,* which represents the frequency with which a student logs into the system. The video-watching construct includes seven indicators regarding different video-watching behaviors: *Load_count, Play_count, Pause_count, Seek_backward_count, Stop_count, Video_count,* and *Watching_time.* The online document reading construct includes the single indicator *Doc_count,* which represents the frequency with which a student reads course documents. The self-assessment mediator construct represents cognitive engagement and includes three different indicators/tasks on the OJ system: *Game_test* (a more challenging assessment with a gamified format), *OJ_test* (a practical programming test on the OJ system), and *Unit_test* (a multiple-choice style test). Online Judge (OJ) is an automatic programming assignment grading system. Finally, the final grade target construct includes the single indicator *Score,* which is used to evaluate a student's final achievement (Figure 1). The details of each indicator are introduced in the Hypotheses section.

In this study, we employed a quantitative statistical analysis approach. Hypotheses 1, 3, 5, and 7 were developed in response to our first research question; hypotheses 2, 4, and 6 were developed in response to our second research question. The third research question was investigated through importance-performance map analysis. We used SmartPLS software (version 3.3.6) to analyze the collected log data. The PLS-SEM method is appealing to many researchers because it enables the development of complex models with many constructs, indicator variables, and structural paths without making distributional assumptions about the data. Moreover, PLS-SEM is suitable when the sample size is restricted by a small population (Hair et al., 2019).

LOG DATA PROCESSING

The data were collected from various platforms. Figure 2 illustrates the overall log data processing flow. The LMS used was provided by Feng Chia University and had basic course management functions. We adopted OpenEdu, an extension of Open edX (Ruiz et al., 2014), as our MOOC platform because it records various student engagement events, including video interaction, course navigation, and problem-solving interaction events, and could therefore serve as an abundant data source for our analysis. Massive Open Online Courses (MOOCs) allow students to study anytime, anywhere via the internet. OpenEdu is a MOOC platform based on Open edX, is well known in Taiwan, and provides learning activity log files as well as detailed explanations of each column in the log (Y.-C. Chiu et al., 2018). We also set up the OJ system for self-practice and the StarTrec lite game for study review. The OJ system recorded the numbers of correct and incorrect answers to a question for each student, which were used as indicators of the student's learning performance.

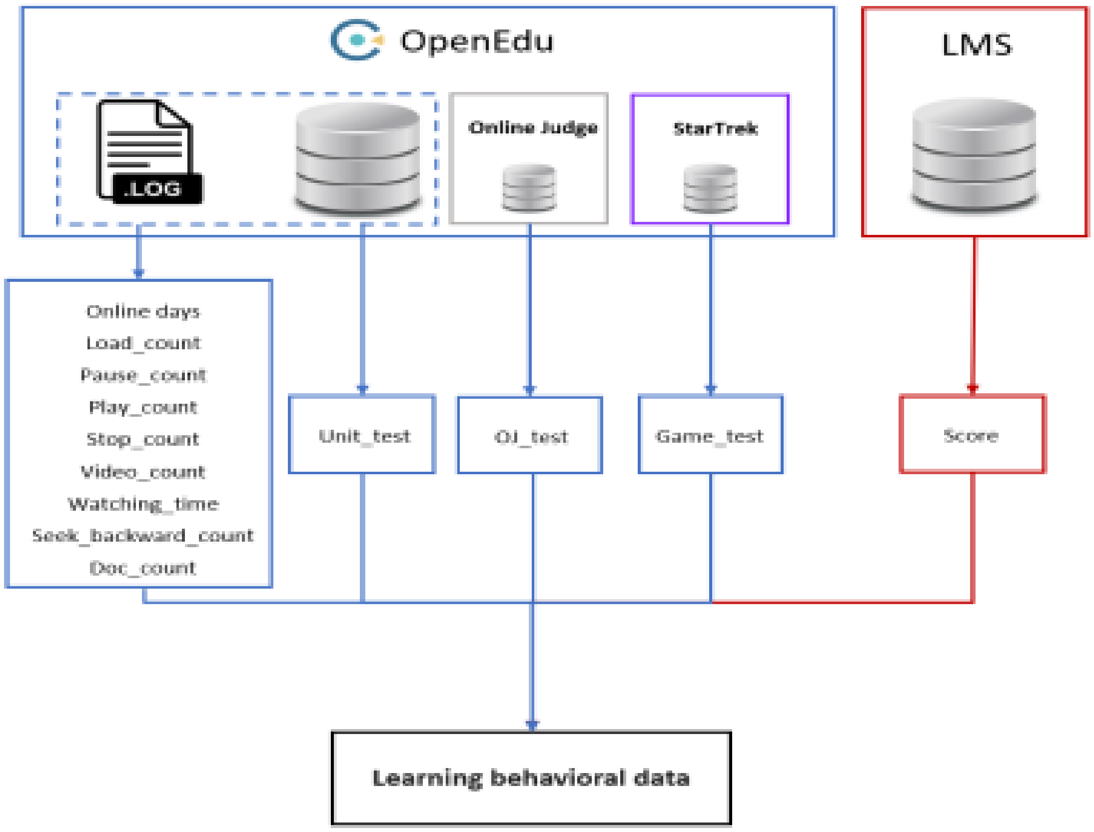


Figure 2. Log data processing

In our student activity calculation, we used data regarding the following three types of events: answering questions, jumping to certain pages, and watching videos. By analyzing events involving jumping to certain pages, we calculated the number of times each student entered the flashcard page and the page through which the textbook was accessed. We used video-watching events to analyze the students' learning behaviors by calculating the number of course videos each student watched and the number of times each learning behavior was performed, as well as the total time each student spent watching videos.

Hypotheses

Online time (behavioral engagement)

The online time construct consists of the *Online_days* indicator, which represents the number of days spent by a student on completing activities on the OpenEdu platform. Some researchers have identified online time as an important variable in online learning. For instance, Kuh (2003) suggested that thorough and accurate information regarding student engagement, including the time and energy devoted by students to educational activities inside and outside of the classroom, is required to assess the quality of undergraduate education at an institution. Trowler (2010) defined student engagement as the commitment of time, effort, and other resources to learning.

LMSs provide various student log data. Using these data, we determined how many days the students were active on the LMS during the programming course by counting the number of days for which the students had logs. Moreover, we assumed that if a student spent a longer time on learning activities than the other students, they would achieve higher scores than the other students. Accordingly, we propose the first and second hypotheses:

H1. Online time is significantly associated with final grades in an online programming course.

H2. Online time is significantly associated with self-assessment in an online programming course.

Video watching (behavioral engagement)

Online video lectures are widely used in e-learning environments. Many previous studies have reported the importance of video-viewing behavior. McGowan and Hanna (2015) described that the analysis of video-viewing behavior provides an abundant and accessible outcome that can be used to improve lecture quality and enhance lecturer and learner performance in programming education. Li (2019) compared prior knowledge with learning performance and discovered that learners with a high prior knowledge level used viewing strategies more frequently, had a more positive attitude toward watching video lectures, and exhibited higher learning performance than those with a low prior knowledge level in a programming course. Yoon et al. (2021) investigated the importance of behavioral patterns and learner clusters in video-based online learning. In addition, the researchers examined the construction of learning behavior patterns through a principal component analysis of behavior log data.

OpenEdu is an extension of Open edX that records numerous video-watching action events. We analyzed the action events and tried to figure out their effects on students' final grades. *Load_count* is the number of *video_load* actions performed by each student and represents the frequency of logging into the digital classroom; *Pause_count* is the number of *video_pause* actions performed and may reflect the students' thinking time required to understand the content; *Play_count* is the number of the *video_play* actions performed and represents the studying frequency for a student; *Stop_count* is the number of *video_stop* actions performed and represents the frequency with which a student stopped watching the videos; *Video_count* is the number of videos watched by a student; *Watching_time* is the total time spent by a student in watching videos; *Seek_backward_count* is the number of *seek_backward* actions performed and represents how often a student actively seeks specific content. Accordingly, we developed the following third and fourth hypotheses:

H3. Video watching is significantly associated with final grades in an online programming course.

H4. Video watching is significantly associated with self-assessment in an online programming course.

Online document reading (behavioral engagement)

In addition to watching lecture videos, reading is crucial when learning to program. In this course, we built an online document where we provided examples of code and their explanation. When the student clicks the "read document" link, we saved the behavior into our log for analysis. We have decomposed the document into chapters and sections which include example codes and their explanations. Students navigate each chapter/section to watch the video, answer the question, and read the document. Even if students download the document to read when offline from the course, they still must click and trigger the events in the system. Therefore, we can perform an analysis based on the clicking behavior.

The online document reading construct consists of the *Doc_count* indicator, which represents the frequency with which a student reads online documents. Our online document was written by the course instructor, which is similar to a kind of online textbook, and used Google Docs. Leppänen et al. (2017) explored students' usage of online learning material as a predictor of academic success in the context of an introductory programming course. Their results indicated that the time spent with the online learning material is a moderate predictor of student success. But in a broader context, course material usage can be used to predict academic success, and such data can be collected *in-situ* with minimal interference to students' learning process. Accordingly, we developed the following fifth and sixth hypotheses:

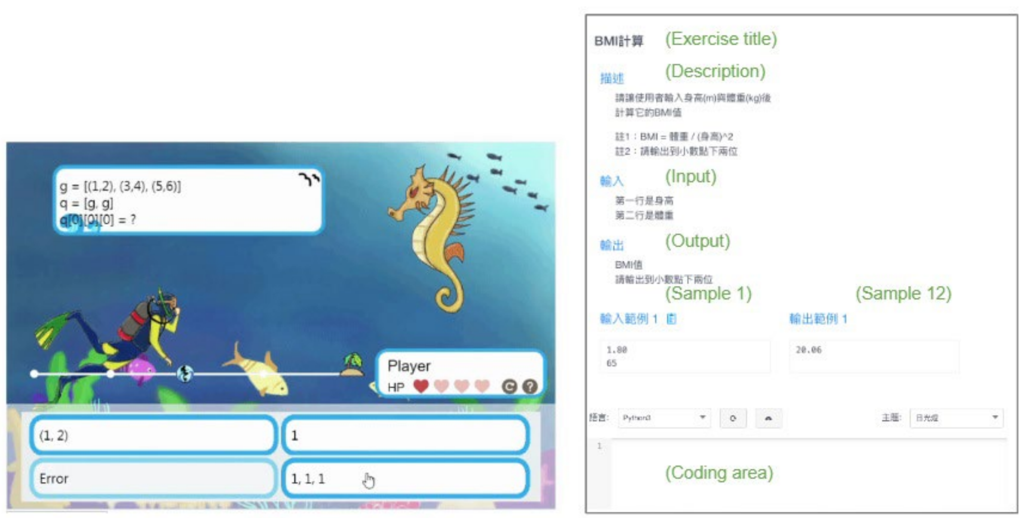
H5. Online document use is significantly associated with final grades in an online programming course.

H6. Online document use is significantly associated with self-assessment in an online programming course.

Self-assessment (cognitive engagement)

Cognitive engagement denotes students' willingness to exert the effort necessary to comprehend complex ideas, master difficult skills, and strengthen their learning and performance (Deng et al., 2019). In the current study, students can participate in cognitive engagements through the OJ system included in the OpenEdu system. OpenEdu is an extension of Open edX that records various learning behavior action events. Usually, instructors teach programming by lecturing and coding exercises in the traditional programming course, but we designed the OJ system, which is like a teacher assistant, to offer such an environment for student practice on the interactive. Using this system, students can get instant responses about their coding results. Therefore, we extended our research model through the inclusion of three self-assessment activities as different types of cognitive engagement.

The first assessment, *Unit_test*, was a multiple-choice style test for assessing a student's understanding of a unit concept. The second assessment, *Game_test*, was a more challenging assessment with a gamified format (Figure 3a). The students were required to acquire 4 HP (health points) by choosing the correct answers to advance to the next level; if they failed, they were required to restart the game. Such a test can evaluate whether a student has the motivation and ability to seek additional information about a unit concept. Hints regarding the answer to each question could be found in the online document or on the Internet. We also encouraged the students to discuss the challenge with each other. The third assessment, *OJ_test*, was a practical programming test that required the students to write executable code in our environment. The OJ system automatically checked the correctness of each student's response by using predefined test cases. The *OJ_test* was the most difficult assessment for beginners because they were required to write code and design a program rather than just understand the underlying concepts and syntax. Figure 3b presents a snapshot of our OJ system. The value of *OJ_test* represents the score a student gets on the OJ assessment.



a b

Figure 3. Plug-in modules: (a) game test, and (b) OJ system

Kanaparan et al. (2019) and Lee (2021) have determined that students' programming self-efficacy and cognitive engagement exerted the strongest positive effects on programming performance. Hu et al. (2021) reported that peer assessment can effectively improve students' programming abilities and, in turn, encourage students to participate in learning. Ortiz-Rojas et al. (2019) observed that gamification significantly improved students' performance in a programming course. Accordingly, we developed the seventh hypothesis as follows:

> *H7. Self-assessment is significantly associated with final grades in an online programming course.*

FINAL GRADE (DEPENDENT VARIABLE)

We selected the final grade construct, which includes the *Score* indicator (a student's final score in the course), as the dependent variable. The students' final grades in their term project, which included a small project on data science (e.g., analyzing traffic data from the past decade or the birth rate in Taiwan since 1900), were considered in the analysis. Our online course provided a tutorial for accessing government open data. Because the data were relevant to the students' lives, they could understand the domain and were interested in the application of the data. Therefore, the students could focus on programming. By evaluating the final project, the teachers were able to determine whether each student could solve problems by using Python. In other words, the students' final projects were reflective of their programming performance.

Kanaparan et al. (2019) defined programming performance as an objective measure of understanding how well a student performs in introductory programming. Shaw (2012) used final examination scores as a dependent variable and measured a learning performance in a programming course. Lau and Yuen (2009) measured programming performance as a dependent variable and discovered that a student's ability positively affects their programming performance; that is, higher-ability students exhibit higher performance. Moreover, students' learning styles significantly affected their programming performance.

RESULTS

ASSESSMENT OF THE MEASUREMENT MODEL

We will describe the structural model by the graphical representation of the PLS path model, as shown in Figure 6 in the Discussion and Implications section. Assessment of the measurement model, including calculating the item reliability, internal consistency, and convergent validity of the research model, is the first step.

The measurement model was assessed based on cross-loading (CL), Cronbach's alpha (α), composite reliability (CR), average variance extracted (AVE), and Fornell-Larcker criterion values. Higher values generally indicate higher reliability; for example, values between 0.60 and 0.70 are considered "acceptable in exploratory research," whereas values between 0.70 and 0.90 range from "satisfactory to good" (Hair et al., 2019). The assessment results are presented in Table 1 and Table 2.

In summary, our analysis results indicated that *Game_test, OJ_test,* and *Unit_test* scores were significant indicators of the self-assessment construct, with CLs of 0.866, 0.787, and 0.926, respectively. *Load_count, Pause_count, Play_count, Seek_backward_count, Stop_count, Video_count,* and *Watching_time* were significant indicators of the Video watching construct, with CLs between 0.655 and 0.873. Finally, the *Score, Doc_count,* and *Online_days* indicators were associated with the final grade, online document reading, and online time constructs, each of which value of 1.000. These results indicate that the model exhibited high reliability and internal consistency.

Another method of assessing a reflective measurement model involves determining its discriminant validity through a comprehensive evaluation of its results (Hair et al., 2013). The AVE of each latent construct should be higher than the construct's highest squared correlation with any other latent construct or its Fornell–Larcker criterion (Hair et al., 2011). The results of the analysis are presented in Table 2.

Table 1. Reliability and convergent validity of the measurement model

Constructs	Items	CL	α	CR	AVE
Final grade	*Score*	single item construct			
Online document reading	*Doc_count*	single item construct			
Online time	*Online_days*	single item construct			
Self-assessment	*Game_test*	0.866	0.824	0.896	0.743
	OJ_test	0.787			
	Unit_test	0.926			
Video watching	*Load_count*	0.873	0.900	0.922	0.629
	Pause_count	0.836			
	Play_count	0.792			
	Seek_backward_count	0.670			
	Stop_count	0.655			
	Video_count	0.738			
	Watching_time	0.857			

Notes: CL, cross-loading; a, Cronbach's alpha; CR, composite reliability; AVE, average variance extracted.

Table 2. Discriminant validity (Fornell-Larcker criteria) of the model

	Final grade	Online document reading	Online time	Self-assessment	Video watching
Final grade	1.000				
Online document reading	0.159	1.000			
Online time	0.409	0.290	1.000		
Self-assessment	0.869	0.165	0.385	0.862	
Video watching	0.380	0.255	0.458	0.305	0.793

Assessment of The Structural Model

We will describe the structural model by the graphical representation of the PLS path model, as shown in Figure 6 in the discussion and implication section. The path coefficient was estimated using a bootstrapping method with 5000 subsamples, and a two-tailed test was conducted to assess its significance. The results of the structural model analysis are presented in Table 3.

In this section, we discuss our first research question: *What is the direct effect of a student's engagement type on their final grade in an online programming course?* P-values were used to assess the significance of each model path. As indicated in Table 3, four of the seven path coefficients of the internal model were statistically significant. Our results indicated that the online time construct had a significant effect on the self-assessment construct (p = 0.000**). In addition, the self-assessment construct had a significant effect on the final grade construct (p = 0.000**), and the video-watching construct had a significant effect on the final grade and self-assessment constructs (p = 0.003** and 0.015*, respectively). Therefore, H2, H3, H4, and H7 were supported, whereas H1, H5, and H6 were not.

Table 3. Significance of direct effects (p ≤ 0.01, *p ≤ 0.05)**

H	Relation	Original Sample (O)	Sample Mean (M)	Standard Deviation	T Statistics (|O/STDEV|)	P Values
H1	Online time -> Final grade	0.047	0.046	0.034	1.394	0.163 (NS)
H2	Online time -> Self-assessment	0.302	0.303	0.056	5.430	**0.000** (S)**
H3	Video watching -> Final grade	0.113	0.113	0.039	2.937	**0.003** (S)**
H4	Video watching -> Self-assessment	0.157	0.164	0.064	2.429	**0.015* (S)**
H5	Online document reading -> Final grade	-0.018	-0.019	0.032	0.582	0.561 (NS)
H6	Online document reading-> Self-assessment	0.037	0.036	0.057	0.658	0.511 (NS)
H7	Self-assessment -> Final grade	0.819	0.819	0.028	29.505	**0.000** (S)**

Note: Bold values indicate significant effects. H, hypothesis; NS, not supported; S, supported.

The next important step in the structural model assessment was evaluating the explained variance (R^2). R^2 values indicate variance, which is explained in each endogenous construct and, therefore, is a measure of the model's explanatory power. The R^2 value is also referred to as in-sample predictive power (Hair et al., 2019). R^2 values of 0.75, 0.50, or 0.25 for endogenous latent variables in a structural model denote high, moderate, or low variance, respectively (Hair et al., 2011), as indicated in Table 4.

Table 4. Explanatory power (R^2)

Dependent variable	R^2	R^2 Adjusted	Decision
Final grade	0.771	0.767	Substantial

Mediation Analysis

Researchers should routinely report total effects (i.e., the sum of direct and indirect effects between two constructs) to not only provide comprehensive insights into the role of the mediating constructs but also provide practitioners with actionable results regarding cause-effect relationships. Moreover, formal mediation analysis using bootstrapping methods is valuable when corresponding hypotheses have been developed (Hair et al., 2013). The results of the analysis of the mediating effects are presented in Table 5.

Table 5. Specific indirect effects (p ≤ 0.01, *p ≤ 0.05)**

Relation	Original Sample (O)	Sample Mean (M)	Standard Deviation	T Statistics (|O/STDEV|)	P Values
Online time -> Self-assessment -> Final grade	0.248	0.248	0.046	5.352	**0.000****
Video watching -> Self-assessment -> Final grade	0.128	0.134	0.054	2.389	**0.017***
Online document reading -> Self-assessment -> Final grade	0.031	0.029	0.047	0.656	0.512

Note: Bold values indicate significant effects.

In this section, we discuss our second research question: *What is the mediating role of cognitive engagement on behavioral engagement and final grade in an online programming course?* As indicated in Table 5, two of the three path coefficients of the mediating effects were statistically significant. The self-assessment construct had significant indirect effects on "Online time and Final grade" ($t = 5.352$, $p = 0.000$**) and "Video watching and Final grade" ($t = 2.389$, $p = 0.017$*). This finding is crucial because if students' online time and video-watching behaviors are supported by self-assessment activities (such as *Game_test, OJ_test,* and *Unit_test*) in an online programming course, their final grades will be significantly affected.

Importance Performance Map Analysis

Finally, in this section, we discuss our third research question: *Which factors most strongly affect students and their performance in an online programming course?* We conducted IPMA, which is one of the advanced PLS-SEM techniques and used the final grade construct as the target variable. IPMA enables researchers to enrich their PLS-SEM analysis to obtain additional results and insights; expanding PLS-SEM analysis to the indicator level facilitates the identification of the most important areas of specific actions (Ringle & Sarstedt, 2016). More specifically, for a specific endogenous construct representing a key target construct in the analysis, IPMA contrasts the structural model total effects (importance, x-axis) and the average values of the latent variable scores (performance, y-axis) to highlight significant areas in which a student's learning activities can be improved. The total effect of a relationship between two constructs is the sum of all the direct and indirect effects in the structural model. When all the total effects (importance values) are more than 0.10, the variables are significant at the $p \leq 0.10$ level. Next, to make the results comparable across different scales, a performance scale of 0 to 100 is used, wherein 0 and 100 represent the lowest and highest performance, respectively (Ahmad & Afthanorhan, 2014; Hair et al., 2012).

As indicated in Figure 4 and Table 6, according to the IPMA results for the constructs, the self-assessment construct had the highest importance and performance ($I = 0.82$; $p = 62.43$) in terms of the students' final grades in an online programming course.

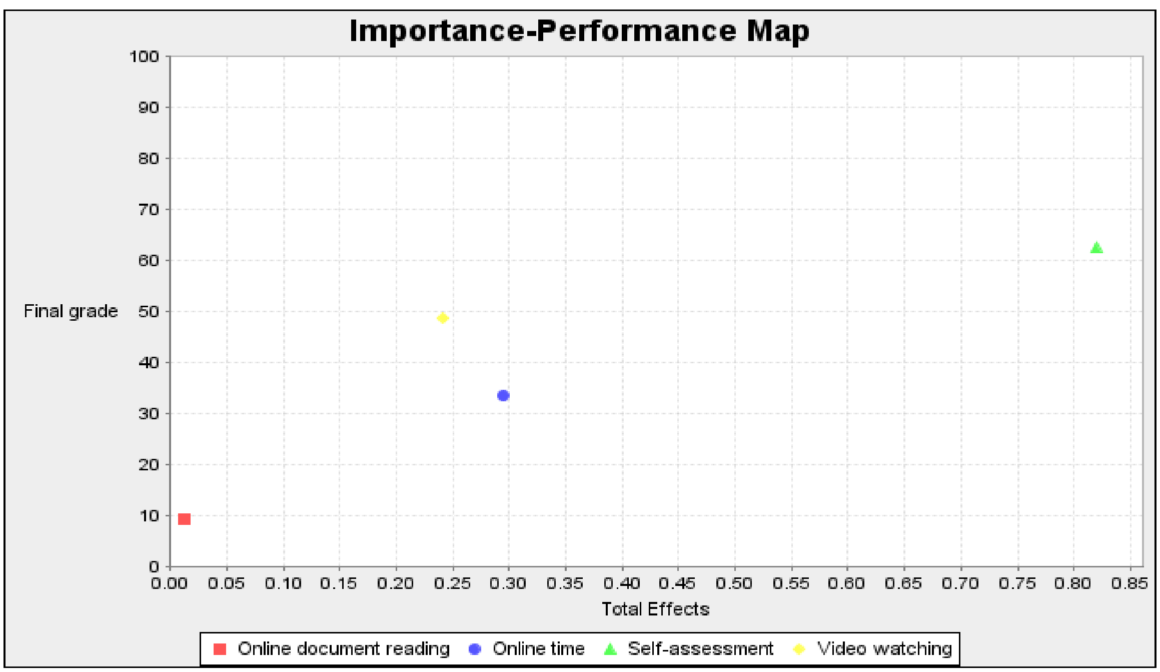


Figure 4. IPMA results for constructs

Table 6. Importance-performance map – standardized effects of constructs on final grade

Engagement types	Constructs	Total Effects/ Importance	Performances
Behavioral engagements	Online document reading	0.01	9.19
	Online time	**0.29**	33.43
	Video watching	**0.24**	48.70
Cognitive engagements	Self-assessment	**0.82**	**62.43**

Note: All total effects >0.10 are significant at the a ≤0.10 level. Bold values indicate significant total effects/importance values and significant performance values.

Online time had the second-highest importance (0.29), but with low performance (33.43), whereas the video watching construct had the third-highest importance (0.24), but with low performance (48.70). Finally, the online document reading construct, which consisted of a single item (*Doc_count*), had the lowest performance and importance (I = 0.01; p = 9.19), indicating that teachers of online programming courses should encourage students to read online documents more frequently.

As indicated in Figure 5 and Table 7, according to the IPMA results for specific indicators, *Unit_test* had both the highest importance and performance (I = 0.36; p = 85.97) in terms of students' final grades in the online programming course. *Unit_test* was a multiple-choice test for assessing students' understanding of a unit concept in the programming course. *Game_test* had the second-both highest importance (0.28) and high performance (77.00). *Game_test* was a more challenging assessment with a gamified format, and it evaluated each student's ability to seek additional information on a unit concept. *OJ_test* and *Online_days* had importance (0.31 and 0.29, respectively) but low performance (47.88 and 33.43, respectively). By contrast, *Video_count* had low importance (0.06) but high performance (62.52). In summary, these results suggest that cognitive learning engagement is crucial to students' achievement in programming distance learning programs because it promotes students' behavioral engagement. Additionally, the amount of time a student spends studying and actively engaging with the online platform is crucial to their learning performance.

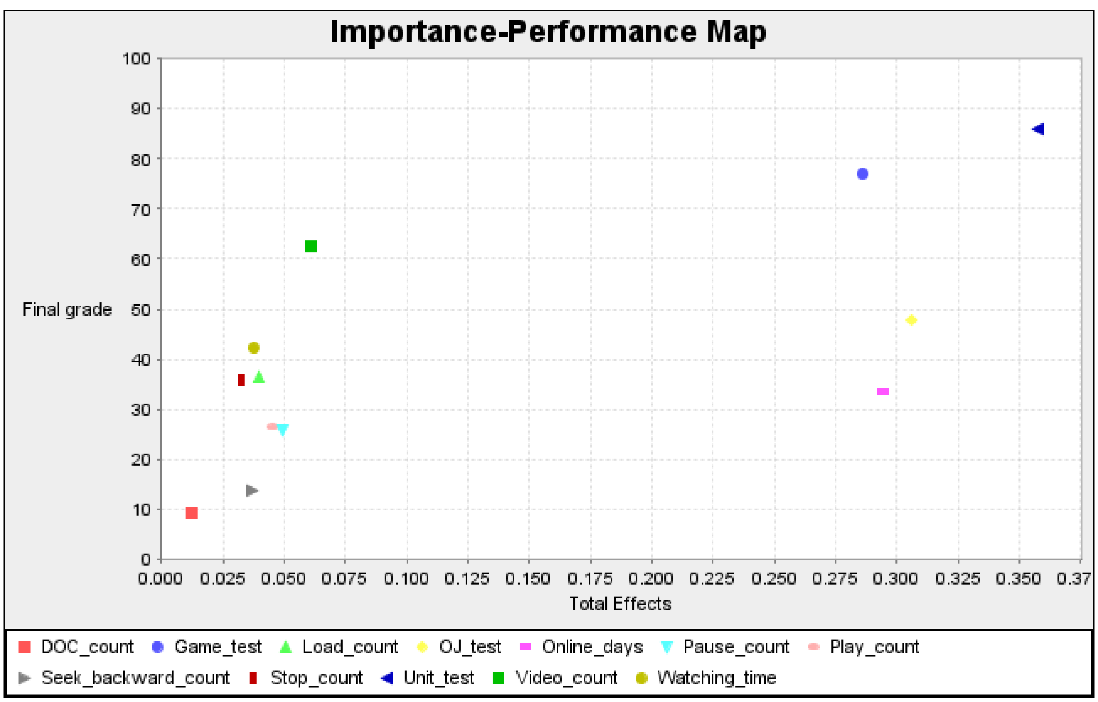


Figure 5. IPMA results for indicators

Table 7. Importance-performance map: standardized effects of indicators on final grade

Engagement types	Items	Total Effects/Importance	Performances
Behavioral engagements	*Doc_count*	0.01	9.19
	Online_days	**0.29**	33.43
	Pause_count	0.05	25.78
	Play_count	0.05	26.47
	Seek_backward_count	0.04	13.73
	Stop_count	0.03	35.88
	Video_count	0.06	**62.52**
	Watching_time	0.04	42.16
	Load_count	0.04	36.32
Cognitive engagements	*OJ_test*	**0.31**	47.88
	Game_test	**0.28**	**77.00**
	Unit_test	**0.36**	**85.97**

Note: All total effects >0.10 are significant at the a ≤0.10 level. Bold values indicate significant total effects/importance values and significant performance values.

DISCUSSION AND IMPLICATIONS

In this study, we applied the behavioral engagement framework and PLS-SEM approach to analyze student log data. Our study contributes to the understanding of how behavioral engagement in distance programming learning affects student achievement directly and through cognitive engagement. The results can serve as a reference for practitioners of distance programming education. Our findings and their practical implications can be divided into two parts according to whether they concern the developers of online educational platforms and university policymakers or instructors and students.

DEVELOPERS OF ONLINE EDUCATIONAL PLATFORMS AND UNIVERSITY POLICYMAKERS

Our results regarding "Online time and Self-assessment" (p = 0.000**) and "Self-assessment and Final grade" (p = 0.000**) imply that the use of self-assessment systems on distance learning platforms is significantly associated with students' active time and video-watching activities (see Figure 6). In other words, universities should develop active self-assessment systems like OJ. This will increase participation in the course and have a positive effect on the final achievement. These results are in line with the result of Wang (2019), which shows that out-of-class problem solving may play a role in consolidating what is learned and important learning activity. In another research, Ngai et al. (2010) noted that students can be taught to assess through a self-assessment system accurately and objectively themselves, which results in a more enjoyable learning experience. Therefore, online education platform developers and university policymakers should pay close attention to the development of self-assessment systems, such as *Game_test, OJ_test,* and *Unit_test*, and design such systems based on students' cognitive skills. This is also supported by the IPMA results. Furthermore, these results were verified by two of the three paths of our mediation analysis (see Table 5). The results of the present study demonstrated that cognitive engagement types are inextricably linked to the development of a behavioral engagement framework in distance programming learning.

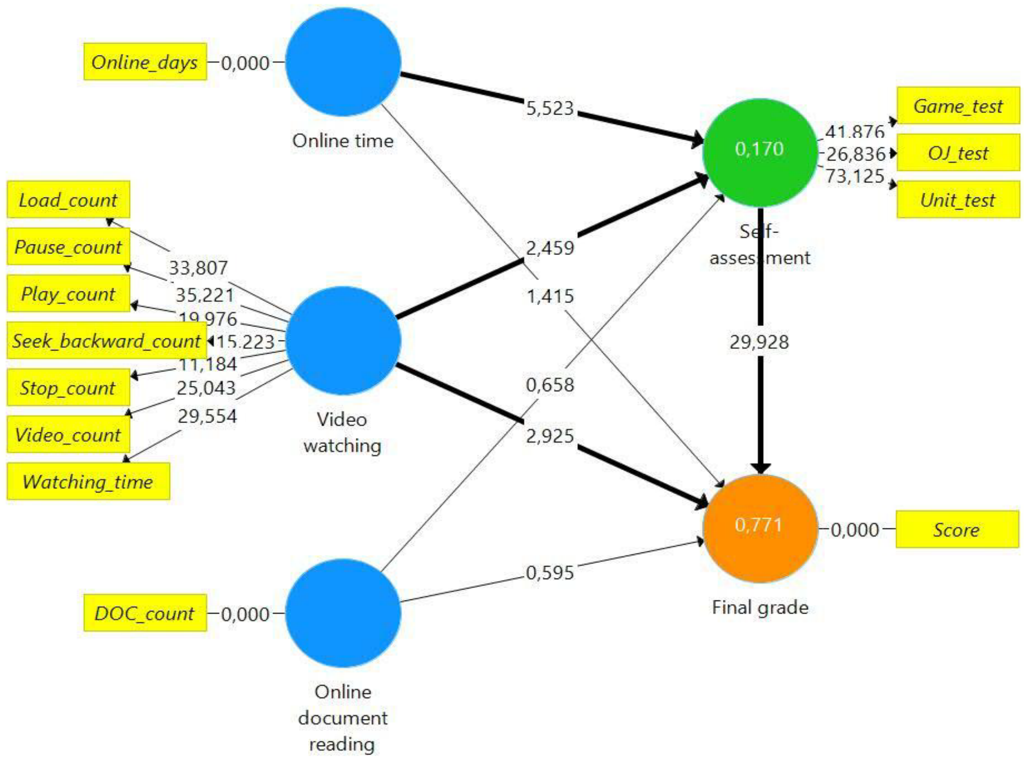


Figure 6. PLS path modeling results

Note: Bold lines indicate significant paths

INSTRUCTORS AND STUDENTS

Our results regarding "Video watching and Final grade" (p = 0.003**) and "Video watching and Self-assessment" (p = 0.015*) imply that watching the instructor's pre-recorded videos and testing themselves with each unit's tests contributed significantly to the students' final achievement (see Figure 6). In the IPMA, *Video count*, the number of videos watched by a student, had a high performance (62.52). In other words, self-learning activities are the most crucial component of distance learning, especially in an online programming course. These results are in line with the creation of short videos that can be a positive way to get students to engage with the material before coming to class. Students indicated the videos helped them learn the material (Carlisle, 2010). In another research (Ronchetti, 2010), the creative use of recorded digital videos can be helpful in promoting a more interesting and interactive teaching style; for example, video watching can (a) help working students by bridging the gap given by their absence during regular lectures, (b) support regular students by giving them the opportunity to recover lectures lost due to forced or elective absence, (c) assist students having difficulties with the lecture's spoken language, and (d) give students a means to review critical sections and check their notes. Therefore, instructors are advised to put substantial effort into the creation of videos for each course session and to actively promote students' interest in the course material.

Interestingly, online document reading (*Doc_count*), which represents the frequency with which a student reads online documents, was not very significantly associated with the constructs of self-assessment (p = 0.511) and the final grade (p = 0.561) (see Figure 6). This finding was further confirmed by our mediation (p = 0.512). In the IPMA, the online document reading construct had the lowest performance and importance (performance = 9.19). This engagement is one of the important learning activities in an online programming course. However, our results show that students appear to be more willing to practice programming than to read and study course materials. Therefore, our results suggest that instructors should pay attention to this issue and improve the learning materials during the course process.

Limitations and Directions for Future Work

Although the findings of this study contribute considerably to the body of literature on online programming learning, the study has several limitations. First, the study participants only included undergraduate students from a single university course. Therefore, we recommend that future studies include a wider range of participants, such as students enrolled in MOOCs.

Second, we initially incorporated 28 log data indicators into our research model, but later we excluded 15 indicators because their CL values were below standard based on the theory of PLS-SEM analysis (Hair et al., 2019). To maintain the meaning and stability of the results, we retained only two single-item constructs with standard values, namely the "Online time" and "Online document reading" constructs. Because PLS-SEM allows for the unrestricted use of single-item constructs, numerous models integrate such constructs. In the case of single-item measures, the final numerical result is equal to 1.00 because the latent variable is measured using only one observed variable (Ringle et al., 2012). Therefore, although this is a common occurrence in PLS-SEM analysis, we suggest that future studies attempt to include more log data items, depending on the number of participants.

Third, the most prevalent conceptualization in the relevant literature is that engagement consists of three distinct yet interrelated dimensions: behavioral, emotional/affective, and cognitive (Fredricks et al., 2016). However, in the present study, we did not account for emotional engagement because it comprises students' affective reactions to their classmates, teachers, learning activities, and school (T. K. Chiu, 2021), which cannot be evaluated precisely in distance learning environments. In the future, researchers can conduct more detailed studies of the effects of emotional engagement as well as additional aspects of students' social engagement to elucidate the factors affecting students' behavioral participation and performance more thoroughly.

Conclusion

Online programming courses allow students to improve their programming concepts and skills. The advantage of log data is that it allows for a realistic analysis by accounting for students' actions in various situations that may arise during a course. PLS-SEM is suitable for applications where strong assumptions cannot be fully met, and it is often referred to as a distribution-free soft modeling approach (Hair et al., 2012).

This study examined the relationships among students' behavioral and cognitive engagement types and performance in an online programming course. The research participants were 217 undergraduate students from Feng Chia University in Taiwan. According to our findings, "Video watching" had the most significant effect than other behavioral constructs in an online programming course. The most significant path relationships in the research model were "Online time and Self-assessment" ($p = 0.000$**) and "Self-assessment and Final grade" ($p = 0.000$**), followed by "Video watching and Final grade" ($p = 0.003$**) and "Video watching and Self-assessment" ($p = 0.015$*). Interestingly, "Online document reading" was not a significant factor in students' final achievement. These findings were supported by the results of both the mediation analysis and IPMA. In summary, self-assessment and video-watching behavior should be considered in the analysis and evaluation of technologies designed to enhance the learning experiences of students in online programming courses.

Acknowledgments

This work was supported by the Ministry of Science and Technology, Taiwan, under grant MOST 110-2221-E-035-031-MY2.

REFERENCES

Ahmad, S., & Afthanorhan, W. M. A. B. W. (2014). The importance-performance matrix analysis in partial least square structural equation modeling (PLS-SEM) with SmartPLS 2.0 M3. *International Journal of Mathematical Research*, *3*(1), 1-14. https://doi.org/10.18488/journal.24/2014.3.1/24.1.1.14

Battestilli, L., Domínguez, I. X., & Thyagarajan, M. (2020). Toward finding online activity patterns in a flipped programming course. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 1345-1345. https://doi.org/10.1145/3328778.3372626

Bergdahl, N., Nouri, J., & Fors, U. (2020). Disengagement, engagement, and digital skills in technology-enhanced learning. *Education Information Technologies*, *25*(2), 957-983. https://doi.org/10.1007/s10639-019-09998-w

Carlisle, M. C. (2010). Using YouTube to enhance student class preparation in an introductory Java course. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, 470-474. https://doi.org/10.1145/1734263.1734419

Chiu, T. K. (2021). Digital support for student engagement in blended learning based on self-determination theory. *Computers in Human Behavior*, *124*, 106909. https://doi.org/10.1016/j.chb.2021.106909

Chiu, T. K. (2022). Applying the self-determination theory (SDT) to explain student engagement in online learning during the COVID-19 pandemic. *Journal of Research on Technology in Education*, *54*(sup1), S14-S30. https://doi.org/10.1080/15391523.2021.1891998

Chiu, Y.-C., Hsu, H.-J., Wu, J., & Yang, D.-L. (2018). Predicting student performance in MOOCs using learning activity data. *Journal of Information Science and Engineering, 34*(5), 1223-1235. https://doi.org/10.6688/JISE.201809_34(5).0007

Cummins, S., Beresford, A. R., & Rice, A. (2015). Investigating engagement with in-video quiz questions in a programming course. *IEEE Transactions on Learning Technologies*, *9*(1), 57-66. https://doi.org/10.1109/TLT.2015.2444374

Deng, R., Benckendorff, P., & Gannaway, D. (2019). Progress and new directions for teaching and learning in MOOCs. *Computers and Education*, *129*, 48-60. https://doi.org/10.1016/j.compedu.2018.10.019

Estacio, R. R., & Raga, R. C., Jr. (2017). Analyzing students' online learning behavior in blended courses using Moodle. *The Asian Association of Open Universities Journal*, *12*, 52-68. https://doi.org/10.1108/AAOUJ-01-2017-0016

Fredricks, J. A., Wang, M.-T., Linn, J. S., Hofkens, T. L., Sung, H., Parr, A., & Allerton, J. (2016). Using qualitative methods to develop a survey measure of math and science engagement. *Learning and Instruction*, *43*, 5-15. https://doi.org/10.1016/j.learninstruc.2016.01.009

Hair, J. F., Ringle, C. M., & Sarstedt, M. (2011). PLS-SEM: Indeed, a silver bullet. *Journal of Marketing Theory Practice*, *19*(2), 139-152. https://doi.org/10.2753/MTP1069-6679190202

Hair, J. F., Ringle, C. M., & Sarstedt, M. (2013). Partial least squares structural equation modeling: Rigorous applications, better results, and higher acceptance. *Long Range Planning*, *46*(1-2), 1-12. https://doi.org/10.1016/j.lrp.2013.01.001

Hair, J. F., Risher, J. J., Sarstedt, M., & Ringle, C. M. (2019). When to use and how to report the results of PLS-SEM. *European Business Review*, *31*, 2-24. https://doi.org/10.1108/EBR-11-2018-0203

Hair, J. F., Sarstedt, M., Ringle, C. M., & Mena, J. A. (2012). An assessment of the use of partial least squares structural equation modeling in marketing research. *Journal of the Academy of Marketing Science*, *40*(3), 414-433. https://doi.org/10.1007/s11747-011-0261-6

Hu, Q., Huang, Y., & Deng, L. (2019). A multivariate learning evaluation model for programming course in online learning environment. *Proceedings of the 2019 14th International Conference on Computer Science & Education* (pp. 737-740). IEEE. https://doi.org/10.1109/ICCSE.2019.8845392

Hu, Q., Huang, Y., & Deng, L. (2021). A framework to evaluate student engagement of programming course in blend learning environment. *Proceedings of the 2021 16th International Conference on Computer Science & Education* (pp. 915-919). IEEE. https://doi.org/10.1109/ICCSE51940.2021.9569553

Isomöttönen, V., & Tirronen, V. (2016). Flipping and blending – An action research project on improving a functional programming course. *ACM Transactions on Computing Education*, *17*(1), 1-35. https://doi.org/10.1145/2934697

Kanaparan, G., Cullen, R., & Mason, D. (2019). The effect of self-efficacy and emotional engagement on introductory programming students. *Australasian Journal of Information Systems*, *23*. https://doi.org/10.3127/ajis.v23i0.1825

Kuh, G. D. (2003). What we're learning about student engagement from NSSE: Benchmarks for effective educational practices. *Change: The Magazine of Higher Learning*, *35*(2), 24-32. https://doi.org/10.1080/00091380309604090

Kurnia, A., Lim, A., & Cheang, B. J. C. (2001). Online judge. *Computers and Education*, *36*(4), 299-315. https://doi.org/10.1016/S0360-1315(01)00018-5

Lau, W. W., & Yuen, A. H. (2009). Exploring the effects of gender and learning styles on computer programming performance: Implications for programming pedagogy. *British Journal of Educational Technology*, *40*(4), 696-712. https://doi.org/10.1111/j.1467-8535.2008.00847.x

Lee, K.-S. (2021). Analysis of the moderating effects of programming languages and the relationship between learners' learning characteristics and achievements. *Journal of Digital Convergence*, *19*(4), 49-55. https://doi.org/10.14400/JDC.2021.19.4.049

Lepp, M., Luik, P., Palts, T., Papli, K., Suviste, R., Säde, M., Hollo, K., Vaherpuu, V., & Tõnisson, E. (2017). Self- and automated assessment in programming MOOCs. In D. Joosten-ten Brinke & M. Laanpere (Eds.), *Technology enhanced assessment, communications in computer and information science: Vol. 653* (pp. 72-85). Springer. https://doi.org/10.1007/978-3-319-57744-9_7

Leppänen, L., Leinonen, J., Ihantola, P., & Hellas, A. (2017). Predicting academic success based on learning material usage. *Proceedings of the 18th Annual Conference on Information Technology Education*, 13-18. https://doi.org/10.1145/3125659.3125695

Li, L.-Y. (2019). The effect of prior knowledge on attitudes, behavior, and learning performance in video lecture viewing. *International Journal of Human-Computer Interaction*, *35*(4-5), 415-426. https://doi.org/10.1080/10447318.2018.1543086

Li, L.-Y., & Tsai, C.-C. (2017). Accessing online learning material: Quantitative behavior patterns and their effects on motivation and learning performance. *Computers Education*, *114*, 286-297. https://doi.org/10.1016/j.compedu.2017.07.007

McGowan, A., & Hanna, P. (2015, November). How video lecture capture affects student engagement in a higher education computer programming course: A study of attendance, video viewing behavior, and student attitude. *Proceedings of the eChallenges e-2015 Conference, Vilnius, Lithuania,* 1-8. https://doi.org/10.1109/eCHALLENGES.2015.7440966

Ngai, G., Lau, W. W., Chan, S. C., & Leong, H.-v. (2010). On the implementation of self-assessment in an introductory programming course. *ACM SIGCSE Bulletin*, *41*(4), 85-89. https://doi.org/10.1145/1709424.1709453

Ortiz-Rojas, M., Chiluiza, K., & Valcke, M. (2019). Gamification through leaderboards: An empirical study in engineering education. *Computer Applications in Engineering Education*, *27*(4), 777-788. https://doi.org/10.1002/cae.12116

Othman, M., Othman, M., & Hussain, F. M. (2013). Designing prototype model of an online collaborative learning system for introductory computer programming course. *Procedia – Social Behavioral Sciences*, *90*, 293-302. https://doi.org/10.1016/j.sbspro.2013.07.094

Picardo, V., Denny, P., & Luxton-Reilly, A. (2021). Lecture recordings, viewing habits, and performance in an introductory programming course. *Proceedings of the Australasian Computing Education Conference*, 73-79. https://doi.org/10.1145/3441636.3442307

Ringle, C. M., & Sarstedt, M. (2016). Gain more insight from your PLS-SEM results: The importance-performance map analysis. *Industrial Management Data Systems, 116*(9), 1865-1886. https://doi.org/10.1108/IMDS-10-2015-0449

Ringle, C. M., Sarstedt, M., & Straub, D. W. (2012). Editor's comments: A critical look at the use of PLS-SEM in MIS Quarterly. *MIS Quarterly*, *36*(1), iii-xiv. https://doi.org/10.2307/41410402

Ronchetti, M. (2010). Using video lectures to make teaching more interactive. *International Journal of Emerging Technologies in Learning*, *5*(2), 45-48. https://doi.org/10.3991/ijet.v5i2.1156

Ruiz, J. S., Díaz, H. J. P., Ruipérez-Valiente, J. A., Muñoz-Merino, P. J., & Kloos, C. D. (2014). Toward the development of a learning analytics extension in open edX. *Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality,* 299-306. https://doi.org/10.1145/2669711.2669914

Shaw, R.-S. (2012). A study of the relationships among learning styles, participation types, and performance in programming language learning supported by online forums. *Computers and Education*, *58*(1), 111-120. https://doi.org/10.1016/j.compedu.2011.08.013

Shi, L., & Weng, Z. (2021). Application of flipped classroom teaching model based on online open shared course platform – Take the "Python Language Programming" course as an example. *International Journal of Science*, *8*(6), 67-74. http://www.ijscience.org/download/IJS-8-6-67-74.pdf

Sun, G., & Bin, S. (2018). Construction of learning behavioral engagement model for MOOC platform based on data analysis. *Educational Sciences: Theory & Practice*, *18*(5), 2206-2216. https://doi.org/10.12738/estp.2018.5.120

Sun, Y., & Chai, R. (2020). An early-warning model for online learners based on user portrait. *Ingénierie des Systèmes d'Information*, *25*(4), 535-541. https://doi.org/10.18280/isi.250418

Trowler, V. (2010). Student engagement literature review. *The Higher Education Academy*, *11*(1), 1-15.

Wang, F. H. (2017). An exploration of online behavior engagement and achievement in the flipped classroom supported by a learning management system. *Computers and Education*, *114*, 79-91. https://doi.org/10.1016/j.compedu.2017.06.012

Wang, F. H. (2019). On the relationships between behaviors and achievement in technology-mediated flipped classrooms: A two-phase online behavioral PLS-SEM model. *Computers and Education*, *142*, 103653. https://doi.org/10.1016/j.compedu.2019.103653

Watson, C., Li, F. W., & Godwin, J. L. (2013). Predicting performance in an introductory programming course by logging and analyzing student programming behavior. *Proceedings of the 2013 IEEE 13th International Conference on Advanced Learning Technologies*, 319-323. https://doi.org/10.1109/ICALT.2013.99

Wu, J.-S., Chien, T.-H., Chien, L.-R., & Yang, C.-Y. (2021). Using artificial intelligence to predict class loyalty and plagiarism in students in an online blended programming course during the COVID-19 pandemic. *Electronics*, *10*(18), 2203. https://doi.org/10.3390/electronics10182203

Xie, S.-T., He, Z.-B., Chen, Q., Chen, R.-X., Kong, Q.-Z., & Song, C.-Y. (2021). Predicting learning behavior using log data in blended teaching. *Scientific Programming*, *2021,* 327896. https://doi.org/10.1155/2021/4327896

Yoon, M., Lee, J., & Jo, I.-H. (2021). Video learning analytics: Investigating behavioral patterns and learner clusters in video-based online learning. *The Internet and Higher Education Pedagogies*, *50*, 100806. https://doi.org/10.1016/j.iheduc.2021.100806

AUTHORS

Dr. Nien-Lin Hsueh received a B.S. degree in Computer Science from Soochow University, Taiwan, in 1989, and M.S. and Ph.D. degrees in Computer Science from National Central University, Taiwan in 1995 and 1999, respectively. He is currently a Professor in the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. His research interests include software engineering, software testing, education technology, and learning analytics.

Bilegjargal Daramsenge received a B.S. degree in Broadcasting Engineering from the School of Broadcasting and Media Arts, Mongolian National University of Arts and Culture, in 2012, and an M.S. degree from the Mongolian University of Science and Technology, Mongolia, in 2014. She is currently pursuing a Ph.D. degree in the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. Her research interests include education technology, learning analytics, distance learning, and software engineering.

Lien-Chi Lai received B.S and M.S. degrees in Computer Science from Feng Chia University, Taiwan, in 2018 and 2021, respectively. He is currently pursuing a Ph.D. degree in the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. His research interests include software engineering, education technology, and learning analytics.