



# Predicting pre-service teachers' computational thinking skills using machine learning classifiers

Hao-Yue Jin<sup>1</sup> · Maria Cutumisu<sup>1</sup> 

Received: 30 November 2022 / Accepted: 31 January 2023 / Published online: 20 February 2023  
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Computational thinking (CT) skills of pre-service teachers have been explored extensively, but the effectiveness of CT training has yielded mixed results in previous studies. Thus, it is necessary to identify patterns in the relationships between predictors of CT and CT skills to further support CT development. This study developed an online CT training environment as well as compared and contrasted the predictive capacity of four supervised machine learning algorithms in classifying the CT skills of pre-service teachers using log data and survey data. First, the results show that Decision Tree outperformed K-Nearest Neighbors, Logistic Regression, and Naive Bayes in predicting pre-service teachers' CT skills. Second, the participants' time spent on CT training, prior CT skills, and perceptions of difficulty regarding the learning content were the top three important predictors in this model.

**Keywords** Computational Thinking · Machine Learning · Classifier · Educational Data Mining · Pre-service Teachers · K-Nearest Neighbors · Logistic Regression · Naive Bayes · Decision Tree

## 1 Introduction

There is an acute need to develop students' computational thinking (CT) in K-12 education, as CT is deemed to be an essential skill for students to succeed in the digital age (Wing, 2006). Many nations have begun to place emphasis on the

---

✉ Maria Cutumisu  
cutumisu@ualberta.ca

Hao-Yue Jin  
hjin2@ualberta.ca

<sup>1</sup> Centre for Research in Applied Measurement and Evaluation, Department of Educational Psychology, Faculty of Education, University of Alberta, 6-102 Education Centre North, Edmonton T6G 2G5, Canada

development of CT skills and have committed to including CT training in K-12 curricula (Bocconi et al., 2016). One way to support students' access to CT is to facilitate teachers' understanding of CT and to help teachers integrate CT concepts into school curricula (Jaipal-Jamani & Angeli, 2017). Thus, several professional development (PD) programs aim to equip in-service and pre-service teachers with CT skills (Jaipal-Jamani & Angeli, 2017) or CT-related pedagogies (Kong et al., 2020). Nevertheless, PD programs have not always been successful in improving teachers' CT skills. For example, Bower et al. (2017) indicated that teachers did not develop an awareness of CT concepts and perspectives after a one-week workshop introducing key ideas of CT. Identifying at-risk learners at an early stage and providing them with immediate support could be an effective way to ensure that every learner achieves a high level of CT skills after training.

Previous studies have explored several influential factors leading to CT training effectiveness. First, training time is considered to be crucial in improving learners' CT skills. Qu and Fok (2021) found that the more time the learners spent on robotics training, the more they developed their skills. Second, learners' prior knowledge can impact their CT skills after the intervention (Hooshyar et al., 2021; Kwon et al., 2021). A study that integrated problem-based learning in an elementary classroom to improve students' CT skills and attitudes revealed that the high prior-knowledge group outperformed the low prior-knowledge group on both CT skills and attitudes (Kwon et al., 2021). Third, learners' dispositions, such as self-efficacy and perceptions of difficulty, were also found to be important factors in CT education (Lee & Lee, 2021; Malmi et al., 2020).

Recently, researchers have applied data mining techniques in educational contexts to predict learning performance (Chen et al., 2021; Lu et al., 2021). Several studies paid particular attention to the prediction of academic achievement. For example, Riestra-González et al. (2021) analyzed log files from learning management systems using machine learning algorithms (e.g., decision trees) to predict students' final grades. Lemay and Doleck (2020) predicted students' assignment completion in massive open online courses (MOOCs) based on video viewing behaviors using several machine-learning models, including logistic regression, naive Bayes, and decision trees. Moreover, previous studies used machine learning algorithms mainly aimed to build a digital environment (e.g., a recommendation system) to develop CT or evaluate CT (e.g., via text mining; Arastoopour Irgens et al., 2020; Lin & Chen, 2020). Little attention has been paid to predicting learners' CT skills and investigating influencing factors in a CT training activity using machine learning approaches.

Therefore, this study aims to apply several machine learning algorithms to predict pre-service teachers' CT skills after an online CT training activity using features from both psychological measurement and log data of user-computer interactions (i.e., prior knowledge, learning behaviors, and perceptions of the learning content). The following research questions guide this research:

1. To what extent can pre-service teachers' CT skills be predicted using machine learning algorithms based on their prior knowledge, learning behaviors, and perceptions of the learning content?
2. What are the most influential predictors of pre-service teachers' CT skills?

## 2 Literature review

### 2.1 CT education for pre-service teachers

In the past few years, several CT studies have focused on pre-service teachers. Researchers examined the effects of programming activities on pre-service teachers' CT skills. For example, Jaipal-Jamani and Angeli (2017) developed a robotics course that required pre-service teachers to build and program three robotic models and the participants were asked to complete a total of four assessment programming tasks. By analyzing the completion of the programming tasks, the researchers found that the pre-service teachers' CT skills improved significantly after the robotics intervention. Mouza et al. (2017) investigated how to enable pre-service teachers to implement CT concepts in their courses. In a 15-week course, 21 pre-service teachers were trained using educational technologies (e.g., Scratch) and CT-supported concepts such as abstraction, problem decomposition, and algorithmic thinking. They also had opportunities to design and implement courses in their classrooms based on the newly acquired instructional tools and pedagogies. A pre-test and a post-test of CT concepts, computing tools, and practice were administered to these participants. The results suggested that this training course positively impacted pre-service teachers' CT concepts, tools, and practice. Similarly, Butler and Leahy (2021) introduced CT concepts and engaged pre-service teachers in CT practice with computational tools (e.g., Scratch) in constructionist learning environments and they found that pre-service teachers could connect constructionist principles with CT in meaningful ways. These pre-service teachers developed a deeper understanding of CT and of ways to embed CT into their practice after taking this course.

As summarized above, researchers have already noticed the significance of preparing pre-service teachers for CT education in the past five years, but most of them focused on deepening teachers' knowledge about CT concepts, CT practice, and CT-relevant pedagogies and transforming teachers' perceptions of CT education using in-person training. However, there is a need to deliver CT training to teachers remotely due to the constraints brought to the forefront by the Covid-19 pandemic. Despite this, research about online CT training for teachers is still sparse. Limited research has been conducted to identify the key factors affecting CT training programs. Therefore, it is necessary to develop an online CT training activity for pre-service teachers and evaluate the critical elements of such an activity.

## 2.2 CT and machine learning

Over the past several years, researchers have applied machine learning in the research and educational practices of CT for different purposes, such as developing CT training platforms. For example, Lin and Chen (2020) designed an augmented reality system for programming with deep learning recommendations, finding that the university students who used this system achieved better learning outcomes. More recently, Hooshyar et al. (2021) created an adaptive digital game for CT development based on the Naive Bayes algorithm. After conducting a teaching experiment with 79 primary-school students, they found that this educational game significantly improved students' CT concepts and skills. Additionally, machine learning techniques have been applied to monitor and predict students' CT learning. For instance, Akram et al. (2020) trained five supervised learning algorithms (e.g., simple linear regression, ridge regression, lasso regression, support vector regression, and Gaussian process regression) to predict five CT skills of 69 middle-school students based on their program snapshots. The best-performing algorithm was Gaussian process regression, yielding the least prediction errors and the best overall performance (mean squared error = 1.71; adjusted  $R^2 = 0.94$ ). Figueiredo et al. (2019) presented a system based on multiple feed-forward neural networks to predict the pass or failure final grade of 85 college students in a programming course. The model achieved a satisfying prediction performance (e.g., an accuracy of 94.12%). Some applications of machine learning have evaluated students' CT skills based on textual data. Arastoopour Irgens et al. (2020) used natural language processing techniques (n-gram and topic modeling algorithms) to extract the CT concept reflections of 41 students from 1,766 responses to open-ended questions. The inter-rater reliability between raters and algorithms yielded *rho* values less than 0.05. Ke et al. (2020) combined a support vector machine (SVM) model with natural language processing techniques to analyze the intervention transcripts of autistic children and obtained an accuracy of 0.761. The approach could link children's certain verbal utterances with predefined CT concepts. An automatic scoring system designed by Li et al. (2018) could transform students' programming submissions into a distance matrix based on TF-IDF and Cosine similarity. Then, a hierarchical clustering algorithm was used to score new submissions, yielding a 75.42% prediction accuracy. Finally, Min et al. (2019) developed an educational game to draw inferences about the CT skills of 182 middle-school students based on deep learning algorithms.

The related literature reveals that most researchers have applied machine learning algorithms in developing CT training platforms (e.g., adaptive CT educational games) or assessing CT skills (e.g., analyzing projects and text), whereas limited research has focused on predicting learners' CT skills using variables derived from prior knowledge, intervention, and perceptions of difficulty of the learning content. To fill this gap, there is a need to develop prediction models of CT skills in an educational context.

### 2.3 Learning performance and machine learning classifiers

Machine learning algorithms can be employed in educational contexts to predict learning performance using variables or features extracted from the data, including interaction data in e-learning platforms, learning behaviors in the classroom, and background information in large-scale assessments. Supervised machine learning techniques can solve two types of prediction problems: classification (i.e., the outcome variable is categorical) and regression (i.e., the outcome variable is continuous). The application of machine learning techniques in predicting learning achievement has gained significant ground in recent years with the development of learning in digital environments, such as MOOCs and learning management systems (LMSs).

In studies focused on the prediction of academic performance, researchers paid particular attention to identifying low-achievement or at-risk students, as shown in Table 1.

For example, Chui et al. (2020) proposed a reduced training vector-based support vector machine (SVM) model to predict whether students will fail a course. They evaluated the performance of 32,593 university students in seven courses. The proposed model achieved an accuracy of 92.2% and was able to predict 91.3% of at-risk students. In addition to the modification of the existing algorithms, researchers compared different frequently used machine learning techniques to identify the model with the best performance. Adnan et al. (2021) compared several machine-learning models, including random forest (RF), SVM, and decision tree (DT) on their accuracy and recall of detecting at-risk university students. The Open University Learning Analytics (OULA) data set was used to evaluate model performance. Predictors included students' demographics, Virtual Learning Environment interaction, assessments, course registration, and courses offered. In this study, random forest achieved the highest overall accuracy and recall. Riestra-González et al. (2021) applied Naive Bayes (NB), logistic regression, multi-layer perceptron (MLP), SVM, and DT to classify at-risk students using their interaction with the learning management system. The findings suggested that MLP obtained the best performance, with

**Table 1** Studies employing machine learning algorithms to predict at-risk students

Citation	LR	NB	SVM	DT	RF	KNN	MLP	NN	GLMNET
Marbouti et al., 2016	*	√	*	*		*	*		
Chen et al., 2021	*	√	*	*	*			*	
Chui et al., 2020			*						
Adnan et al., 2021			*	*	√				
Riestra-González et al., 2021	*	*	*	*			√		
Z. Yang et al., 2020					*				
Bertolini et al., 2021	*		*		*				√

\* Algorithms used, √ the algorithm with the highest accuracy, *LR* logistic regression, *NB* naive Bayes, *SVM* support vector machine, *DT* decision tree, *RF* random forest, *KNN* K-nearest neighbors, *MLP* multi-layer perceptron, *NN* neural network, *GLMNET* generalized linear model with elastic net

an overall accuracy of 80.1% when 10% of the course had been delivered, and 90.1% when half of it had been delivered. Z. Yang et al. (2020) used RF classification models for the early prediction of at-risk students in a physics classroom. Demographic information, home average, class attendance, test scores, and a pre-test score were included as predictors in this model. The model yielded an overall accuracy of 80%. However, the RF algorithm could only predict 37% of the at-risk students. Bertolini et al. (2021) applied LR, SVM, RF, and the generalized linear model with elastic net (GLMNET) to predict university students' academic performance in a Science course. The findings showed that GLMNET achieved a significantly higher accuracy compared with other algorithms, whereas LR achieved the poorest prediction.

According to the studies mentioned above, researchers used various machine learning algorithms to predict learning performance and obtained different optimal models. Additionally, few studies have combined self-reported survey data and log data to predict learning performance. Thus, the current study compares four machine learning classifiers to predict pre-service teachers' CT skills using learners' prior knowledge, learning behaviors (i.e., training time), and perceptions of the learning content.

### 3 Method

#### 3.1 Participants and procedure

Participants were 93 pre-service teachers (38 males, 48 females, 5 non-binary, and 2 not reported) from the Faculty's Research Participation Pool program. The consent form and surveys were administered using the Qualtrics online platform and were distributed through the SONA system according to the ethics protocol Pro00112720. The participants completed the online consent form and pre-survey for an average of 20.3 min and a median of 14.0 min. One week after the pre-survey, they completed the intervention and the post-survey for an average of 41.5 min to prevent fatigue. This delay was also added to temporally space out testing and allow any potential learning from the pre-survey to solidify.

#### 3.2 Online CT training

During the CT intervention, introduction sentences were displayed on the screen guiding the participants to work on the code training tasks: "In this section, we will let you know whether you get the answer right or wrong. You will also be able to read the explanation for the answers."

Next, students were provided with four code training tasks (see Fig. 1) consisting of multiple-choice, sorting, and ranking items. After completing each task, participants were asked to briefly explain their responses. Participants received instant feedback when they submitted their answers during the CT training. Then,

Consider this code:

```


when clicked
say Hello! for 2 secs
set number to 0
set number_of_times to 3
say number for 2 secs
repeat number_of_times
  set number to number + number_of_times
  say number for 2 secs
say Goodbye! for 2 secs
say number for 2 secs
say number_of_times for 2 secs
    
```

What will appear on the screen, after the green flag is clicked?

- (A) Hello! 0; 0; Goodbye!; 0; 3
- (B) Hello!; 0; 3; 6; 9; Goodbye!; 9; 3
- (C) Hello!; 0; 3; Goodbye!; 3; 3
- (D) Hello!; 0; 3; 6; 9; Goodbye!; 9; 9


Please explain briefly why you selected this option:

Fig. 1 Screenshot of the CT training



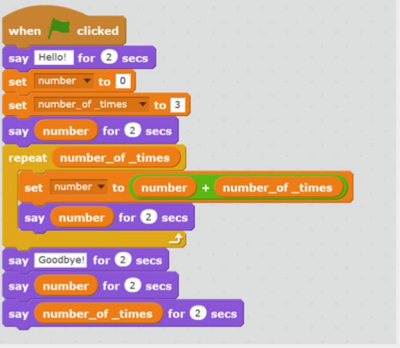
No, that is not the solution.  
Would you like to see how we solved this problem?

No, I would like to move on.      Yes, I would like to see how to solve this problem.



That's correct.  
Would you like to see how we solved this problem?

No, I would like to move on.      Yes, I would like to see how to solve this problem.

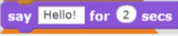


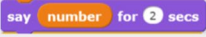
```


when clicked
say Hello! for 2 secs
set number to 0
set number_of_times to 3
say number for 2 secs
repeat number_of_times
  set number to number + number_of_times
  say number for 2 secs
say Goodbye! for 2 secs
say number for 2 secs
say number_of_times for 2 secs
    
```

The correct answer is:

(B) Hello!; 0; 3; 6; 9; Goodbye!; 9; 3

At first, "Hello" is displayed on the screen after  is executed.

The initial value of the variable `number` is 0 and of the variable `number_of_times` is 3. So 0 is displayed on the screen after  is executed.

Because the variable `number_of_times` is 3, this code, , called a loop, is equivalent to this code:

```
repeat 3
```

Therefore, the loop will execute the code inside it 3 times. In the loop, the code, `number = number + 3`, will increase the value of the variable `number` by 3, because 3 is the value of the variable `number_of_times`.

In the first execution of the loop, `number = 0 + 3 = 3`.  
 In the second execution of the loop, `number = 3 + 3 = 6`.  
 In the third execution of the loop, `number = 6 + 3 = 9`.  
 Thus, the final value of the variable `number` is 9, but the value of the variable `number_of_times` remains unchanged (it still equals 3).

Fig. 2 Screenshot of the feedback and explanations during the CT training

participants had opportunities to read the explanations of the task provided by the researchers (see Fig. 2). The time spent on the training tasks and the time spent on reading the explanations were both recorded. When the participants finished the CT training, they could choose to rest before continuing to complete the post-survey.

### 3.3 Data collection tools

**Computational thinking skill assessment** The CT assessment used in this study was adapted from the Callysto Computational Thinking test (CCTt, Adams et al., 2019; Cutumisu et al., 2021). The original assessment takes approximately 45 min to complete. One worked example and four multiple-choice spatial CT items that measured CT skills were selected for this study.

**Perceptions of the CT training** At the end of the intervention, participants were asked “From 0 to 10, how do you think you did in the training session?” and “From 0 to 10, how difficult did you find this training session?” to measure their self-efficacy and perception of CT training difficulty, respectively.

**Demographic information** This form was designed to collect the following information from the participants: gender, age, grade level, program, and programming experience.

### 3.4 Measures

#### 3.4.1 Outcome variable

In this study, participants whose post-CT skill scores fell one standard deviation below the mean were identified as low-CT pre-service teachers (“1”), whereas the rest of the participants were identified as high-CT pre-service teachers (“0”). Thus, there were 15 low-CT pre-service teachers in the total sample of 93 participants.

#### 3.4.2 Feature selection

Table 2 shows the features derived from the data collected from the Qualtrics platform. In this study, a Spearman correlation analysis of all the features with the post CT skills scores was conducted because the variables in the data set were not normally distributed. Only the features that were significantly correlated with the post CT skills were then selected. Thus, as shown in Table 3, only 7 features were included in the data analysis.



**Table 2** The description of the features included in the models

Feature	Description	Type
Prior knowledge	Programming experience	Continuous
Learning behaviors	Prior CT skills	Continuous
	Resting time	Time spent on resting between the CT training and the CT assessment
	Training time	Time spent on the learning tasks, excluding the time spent on reading explanations
	Explanation reading time	Time spent on reading explanations
	Correct answers	The number of correct answers in CT training
Perceptions of the learning content	Number of explanations viewed	Continuous
	Self-efficacy	The participants' perceptions of their own CT training performance
	Perceptions of difficulty	The perceptions of CT training difficulty

**Table 3** The descriptive analysis and Spearman correlation results of the included variables

Feature		Correlation	<i>p</i>	<i>M</i> ( <i>SD</i> )
Prior knowledge	Prior CT skills	.427***	< 0.001	0.62 (0.26)
Learning behaviors	Training time	.469***	< 0.001	1322.79 (979.35)
	Explanation reading time	.364***	< 0.001	47.71 (43.42)
	Number of correct answers	.289**	0.005	1.19 (1.17)
	Number of read explanations	.290*	0.005	2.37 (1.37)
Perceptions of the learning content	Self-efficacy	.302**	0.003	3.18 (3.18)
	Perceptions of difficulty	-.300**	0.003	6.35 (3.34)

\* $p < 0.05$ , \*\* $p < 0.01$ , \*\*\* $p < 0.001$

### 3.5 Data analysis plan

#### 3.5.1 Balancing the data set

There were 15 low-CT pre-service teachers in the total sample of 93 participants. Thus, the original data set was imbalanced, which could bias the machine learning classifiers towards learning to always predict the majority class. To overcome this problem, the synthetic minority oversampling technique (SMOTE) was applied to balance the training data set. This approach selects a minority class data point ‘a’ at random and locates its closest minority class neighbors. Then, a synthetic data point is generated by randomly selecting one of the nearest *k* neighbors ‘b’ and connecting it to the original data point ‘a’ to form a line segment.

#### 3.5.2 Machine learning models

This research used four machine learning algorithms: logistic regression, naive Bayes, k-nearest neighbor, and decision tree.

**Logistic Regression (LR)** Logistic regression is a supervised learning algorithm commonly used in educational settings (e.g., Braunstein et al., 2008; Eckles & Stradley, 2012). It is a probabilistic model, calculating the probability of a categorical variable (e.g., letter grade, pass/no-pass) from a number of predictor variables. In the training phase, the coefficients are estimated based on the training data. Forward-selection logistic regression was employed in this study because it is commonly used in educational settings.

**Naive Bayes (NB)** Naive Bayes is also a probabilistic model like logistic regression. It calculates a conditional probability distribution over the output of a function based on applying Bayes’ theorem with the (naive) assumption of independence between the predictive variables (Russell & Norvig, 1995). Although this assumption is often violated (e.g., correct answers and explanations reading time are not independent),

the algorithm yields good results in practice, being widely used in several applications (e.g., natural language processing tasks).

**K-Nearest Neighbor (KNN)** The K-Nearest Neighbor algorithm is a non-linear, non-parametric, instance-based classifier that can be used for both regression and classification tasks. KNN classifies an object (e.g., a student) using a majority vote of its K nearest neighbors (Friedman et al., 1977). It calculates the distance between the objects using a similarity metric of the data instances. Here, the five-nearest neighbors algorithm was used to identify low-CT pre-service teachers. The Euclidean distance metric between two instances was calculated to find the nearest neighbors.

**Decision Tree (DT)** Like KNN, the decision tree is a versatile non-linear, non-parametric supervised learning algorithm that can be used for both regression and classification prediction tasks. In each step, DT partitions the data based on one variable (e.g., training time) until all data in each node have only one category label (e.g., low or high) or all variables have been used. Partitioning is done by defining a score function that calculates the purity of all possible nodes and selects the variable that generates the purest nodes. The prediction of the class of an instance in a classification tree is obtained by traversing the tree until that instance reaches a leaf and taking the majority vote of the classes of all the instances landing at that leaf.

### 3.5.3 Model evaluation

Using a validation-set model evaluation approach, the data set was randomly split into a training set (80%) and a test set (20%) with the *createDataPartition* function in R that aims to balance the class distribution within the splits. Then, the model was trained on the balanced training set (using leave-one-out cross-validation to choose the best model) and evaluated on both the training set (in-sample or training error) and the test set (test error). Several performance criteria were employed: accuracy (Eq. 1), precision (Eq. 2), recall (Eq. 3), specificity (Eq. 4), and F1-score (Eq. 5). *Accuracy* was used to calculate the total number of cases correctly classified. *Precision* was used to calculate the ratio of correctly detected low-CT participants to the total predicted low-CT pre-service teachers. *Recall* was used to calculate the ratio of correctly detected low-CT participants to the total number of actual low-CT pre-service teachers. *Specificity* was used to calculate the ratio of correctly detected high-CT participants to the total number of actual high-CT pre-service teachers. *F1-score* combines the precision and recall of a classifier into a single metric by taking their harmonic mean and it is used to assess the performance of a classifier when the data is not balanced because it provides a fairer measure of performance than precision or recall in this case.

Additionally, receiver operating characteristics (ROC) curves and area under the ROC curve (AUC) were also used as criteria to evaluate the model performance. The ROC curve shows the relationship between the true positive rate (TPR) and the

false-positive rate (FPR). It also reflects the relationship between sensitivity and specificity (i.e., when the sensitivity rises, specificity declines and vice versa). If the curve is closer to the top left corner of the graph, then the algorithm shows high accuracy. Otherwise, the curve approaches the diagonal, which shows a performance accuracy that is no better than guessing. Values of AUC exceeding 0.9 indicate excellent results; values between 0.8–0.9 are considered good; values between 0.7–0.8 are considered fair; and values smaller than 0.6 are considered poor.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total number of cases}} \quad (1)$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (2)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (3)$$

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} \quad (4)$$

$$F1 - \text{score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

## 4 Results

### 4.1 CT performance prediction

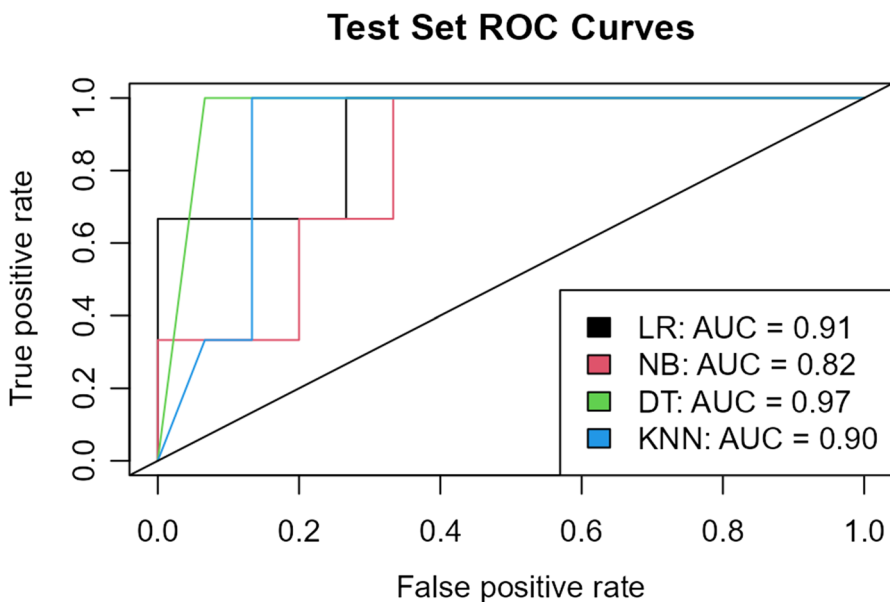
The first research question aims to explore the extent to which pre-service teachers' CT skills can be predicted using machine learning classifiers. The results of testing four algorithms using the selected features are reported in Table 4. The

**Table 4** The model performance of the four classifiers

Model	Prediction Error	Accuracy	Specificity	Precision	Recall	F1-Score
NB	Test error	0.722	0.733	0.333	0.667	0.444
	Train error	0.897	1.000	1.000	0.794	0.885
LR	Test error	0.778	0.800	0.400	0.667	0.500
	Train error	0.849	0.889	0.879	0.810	0.843
KNN	Test error	0.889	0.867	0.600	1.000	0.750
	Train error	0.818	0.937	0.917	0.698	0.793
DT	Test error	0.889	0.867	0.600	1.000	0.750
	Train error	0.897	0.921	0.917	0.873	0.894

test error represents the performance obtained after training the model on the balanced data set and testing it on a random holdout set taken from the original imbalanced data set. The train error represents the in-sample error (training and testing the model on the balanced data set). The cross-validation procedure yielded an optimal  $k$  value for KNN of 5 and an optimal  $cp$  value for DT of 0.06. All algorithms (LR, NB, KNN, and DT) achieved high accuracy scores ranging from 0.722 to 0.889. Among these four algorithms, the best model for overall accuracy and F1-score was Decision Tree (DT), which correctly identified 88.9% of the pre-service teachers' CT skills. As for predicting high-CT pre-service teachers, the DT yielded the highest accuracy because it could correctly identify 86.7% of high-CT participants in all actual high-CT cases. This model is also the best in predicting the low-CT class, which is important for the early detection of struggling learners. According to its recall value, DT could correctly detect all low-CT participants in the test set.

As shown in Fig. 3, all ROC curves are situated in the top left corner because all prediction models produced fairly accurate results. Figure 3 also displays the AUC values for all four classifiers, showing that the DT has the highest AUC value, followed by LR and KNN. The NB model has the lowest AUC value, suggesting the poorest prediction of pre-service teachers' CT skills. Thus, overall, the DT model performed best based on the above criteria in identifying both low-CT and high-CT pre-service teachers.

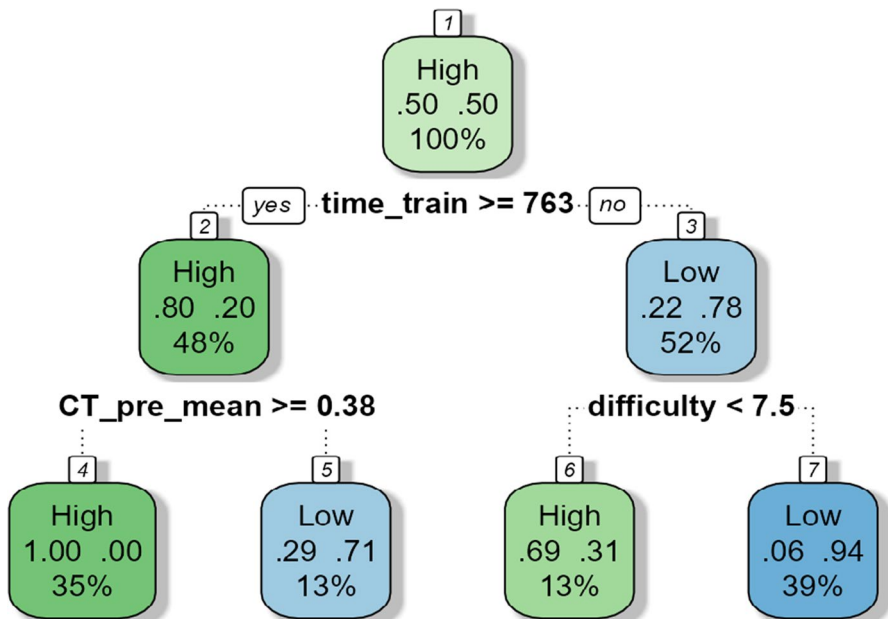


**Fig. 3** The ROC curves of the four classifiers. *Note.* The diagonal solid line represents the chance classifier and it has an AUC value of 0.5

## 4.2 Influential factors of CT performance

The second research question aims to identify the most influential factors affecting pre-service teachers' CT skills. Figure 4 shows the final decision tree. Oval nodes indicate class segmentation variables. The information included on the dashed line shows the segmentation score of the sample from the variable tested in the previous node. For example, for the root node (*time\_train*), the main sample is divided into two sub-samples, one on the left with the instances with training time over 763 s (12.7 min), and one on the right with the instances with the training time below 763 s. The value in the rectangle indicates the percentage of cases of the subsample included in the previous node progressing through that branch. Thus, when the participant's training time is less than approximately 13 min and the perception of the training difficulty exceeds 7.5, the participant is more likely to be classified as a low-CT pre-service teacher. In addition, a participant is more likely to be classified as a low-CT learner if the participant's pre-CT skills score is less than 0.38, even though the learner's training time exceeds 13 min. In contrast, if a participant's training time exceeds 13 min and the pre-CT skills score is above 0.38, the participant is more likely to be assigned a high-CT class. Moreover, a participant could be classified as a high-CT pre-service teacher if the participant's perception of training difficulty is below 7.5, regardless of the actual training time.

As shown in Table 5, the top three important features affecting pre-service teachers' CT skills are training time, pre-CT skills score, and perceptions of training



**Fig. 4** The graphical representation of the decision tree. Note. *time\_train*: time spent on the learning tasks, excluding the time spent on reading explanations; *difficulty*: the perceptions of CT training difficulty; *CT\_pre\_mean*: pre-scores of CT skills

**Table 5** The feature importance of the DT model

Features	Values
Training time	100.00
Prior CT skills	61.56
Perceptions of difficulty	48.39
Explanation reading time	7.60
Number of correct answers	7.18
Number of read explanations	2.85
Self-efficacy	0.00

difficulty. Findings suggest that the participants with low pre-CT skills scores are more likely to be classified as low-CT pre-service teachers. Additionally, spending less time on CT training as well as perceiving the training tasks as being difficult leads to a low-CT skill classification. However, contrary to our expectation, the time spent reading explanations and the number of times the participant chose to read explanations seemed to be less important in the DT model, which indicates that reading explanations may affect pre-service teachers' CT skills to a lesser extent.

## 5 Discussion

The main objective of this research was to compare and find the best prediction model among four different machine learning classifiers in an online CT training environment and to identify the most important factors affecting pre-service teachers' CT skills. After comparing different models based on accuracy, specificity, recall, precision, and F1-score, the best model for predicting low-CT pre-service teachers was Decision Tree (DT). Moreover, the top three influencing factors in this DT model were training time, prior CT skills, and perceptions of difficulty.

### 5.1 CT performance prediction

One objective of this study was to investigate whether machine learning algorithms could accurately identify at-risk learners in the proposed online CT activity. Compared to studies using a small data set, the F1-score of DT in our research is acceptable. For instance, Urkude and Gupta (2019) used students' contextual data to predict whether students will graduate or not. They obtained an F1-score for DT of 0.655 for a sample size of 100. The F1-score for DT increased to 0.750 when the data set size increased to 300 records.

Of the two probabilistic classifiers (i.e., LR and NB), LR slightly outperformed NB in terms of all evaluation metrics in this data set. Other researchers found that LR worked better than NB in their classification tasks (Aborisade & Anwar, 2018). However, Pundlik (2016) pointed out that the accuracy of NB is often better than that of LR when the training size is small. For example, Marbouti et al. (2016) used

both LR and NB to predict at-risk university students. They trained their models on a data set of 780 students and found that NB outperformed LR with an accuracy of 86.2% in identifying students who failed to pass the course.

Of the two non-probabilistic classifiers (i.e., KNN and DT), DT generally outperformed KNN, which is consistent with previous studies (e.g., Pathak & Pathak, 2020; Wiyono & Abidin, 2019). For instance, Wiyono and Abidin (2019) applied KNN and DT to predict whether students were active or not as determined based on their performance. The results showed that DT yielded an accuracy of 93%, whereas KNN yielded an accuracy of 92%. One potential reason might be that the log data generated in the learning platform usually included mixed or nonlinear predictor relationships and DT has an advantage when working with nonlinear data (Rizvi et al., 2019).

Based on the present findings, the DT model is able to identify low-CT pre-service teachers well. Its high specificity indicates that very few actual high-CT participants were predicted incorrectly. Its high recall indicates that the DT model identifies low-CT learners well. Overall, the F1-score indicates a good overall DT model.

## 5.2 Influential factors of CT performance

In terms of factors affecting CT skills, we had anticipated that the explanation reading time and number of explanations viewed would be important factors in the prediction model because the given explanations were a critical part of our training activity. However, the results suggested that training time, prior CT skills, and perceptions of difficulty were more predictive of the response variable.

### 5.2.1 Training time

The time spent completing the training tasks seems to be important, because learners who choose to spend more time in the training session might be more engaged in the learning process. Many studies have emphasized the positive relationship between engagement and learning performance (e.g., Pi et al., 2021; Qureshi et al., 2021). Learners were likely more engaged when solving the items of the training session than when reading explanations. This is supported by the higher average time spent on the training tasks (approximately 20 min) compared to the lower average time spent on reading explanations (47.7 s). However, in this CT training activity, the given explanations were the only materials that could help participants learn about programming concepts. Perhaps other teaching strategies (e.g., self-explanation or think alouds) could be applied in the online CT training activity to improve learners' engagement.

### 5.2.2 Prior CT knowledge

This finding could be partly explained by the fact that prior knowledge levels can impact learning behaviors (Kitsantas, 2013; Yang & Quadir, 2018). Yang et al. (2018) conducted a study about self-regulated learning and prior knowledge. They



found that learners with higher or lower levels of prior knowledge behaved differently during the performance phase. Specifically, learners with more prior knowledge tend to solve more complex problems, while learners with less prior knowledge tend to solve simpler problems. Also, learners with less prior knowledge tend to apply multiple problem-solving strategies (e.g., self-reflection strategies), while learners with more prior knowledge tend to correct mistakes by only checking answers. In the present study, participants could only get access to the correct answers and to the explanations of the answers. Providing more support or scaffolding for pre-service teachers with lower initial CT skills could be a useful direction for future research.

### 5.2.3 Perceptions of difficulty

Perception of difficulty is another crucial factor influencing participants' CT performance. Participants who felt that the learning content was difficult were more likely to achieve low CT performance. Previous studies have identified the relationship between learning difficulty and learning performance. For example, Hung et al. (2013) designed a digital educational game with different difficulty levels. They found that students using learning materials of moderate difficulty achieved a significantly higher learning performance than those using learning materials of higher difficulty. In addition, Cheng et al. (2021) found that there was a positive correlation between the perceived difficulty level of online courses and students' preference of teacher-led strategies, indicating that online learning materials that are too difficult might lead to a decrease in self-directed learning. Adjusting difficulty levels of learning content for learners with different perceptions of difficulty might be a solution for improving learning performance.

## 5.3 Limitations and future work

One limitation of this research is the short duration of the training session (around one hour). This could have influenced the model performance and several key features. For example, the number of explanations viewed and the time spent on reading explanations are likely to be more important factors if the intervention duration were longer (e.g., an entire semester). Another limitation is that the SMOTE approach used to balance the data set could have caused overfitting and skewness in the class distribution. The Tomek Links technique used for undersampling could be an alternative approach for large sample sizes (Zeng et al., 2016). Also, there is room for improvement in classification accuracy. Future research can consider using ensemble methods to increase the model performance.

In future research, it might be possible to devise teaching strategies to facilitate pre-service teachers' engagement in completing learning tasks and reading the explanations provided. For example, using self-explanation strategies could help learners reflect on their solutions and correct their answers, becoming aware of their weaknesses.

## 6 Conclusion

Four machine learning algorithms were compared and contrasted to predict pre-service teachers' CT skills after an online CT training activity. The decision tree model achieved the highest prediction performance. For online learning activities with fewer than 100 participants and fewer than 20% low-performance learners among these participants, it might be possible to use DT to identify low achievers with high accuracy. Another contribution of this work is the identification of the key features employed in an online CT learning activity. This study shows that although training activities contribute to CT performance, prior knowledge is an important factor that can facilitate learners' CT skills. Thus, this study suggests that, before or at the beginning of the CT training activities, the instructor could initiate warm-up activities to help learners (especially those with lower prior knowledge levels) acquire or retrieve knowledge and skills that will better prepare them for upcoming challenges. Then, throughout the CT activities, several learning strategies (e.g., think alouds or self-explanation) can be integrated into the learning environments to engage learners in CT learning. In addition, this study makes it possible to predict at-risk learners at an early stage of learning, which allows instructors to offer classes, help, additional resources, or other assistance tailored to these learners.

**Acknowledgements** We would like to thank the editor, the anonymous reviewers, and the study participants. We are also grateful to the Social Sciences and Humanities Research Council of Canada—Insight Development Grant (SSHRC IDG) RES0034954, the Social Sciences and Humanities Research Council of Canada—Insight Grant (SSHRC IG) RES0048110, the Natural Sciences and Engineering Research Council Discovery Grant (NSERC DG) RES0043209, and the CanCode – Cybera Inc. (Callysto) Grant RES0059331 for supporting this research.

**Data Availability** The data file is available upon request.

## Declarations

**Conflict of interest** None.

## References

- Aborisade, O., & Anwar, M. (2018). Classification for authorship of tweets by comparing logistic regression and naive bayes classifiers. *IEEE International Conference on Information Reuse and Integration (IRI), 2018*, 269–276.
- Adams, C., Cutumisu, M., Yuen, C., Hackman, L., Lu, C., & Samuel, M. (2019). *Callysto Computational Thinking Test (CCTr) Teacher Version [Measurement instrument]*. Retrieved on February 9, 2023. <https://www.callysto.ca/>
- Adnan, M., Habib, A., Ashraf, J., Mussadiq, S., Raza, A. A., Abid, M., Bashir, M., & Khan, S. U. (2021). Predicting at-risk students at different percentages of course length for early intervention using machine learning models. *IEEE Access*, 9, 7519–7539. <https://doi.org/10.1109/ACCESS.2021.3049446>
- Akram, B., Azizoltani, H., Min, W., Wiebe, E., Mott, B., Navied, A., Boyer, K. E., & Lester, J. (2020). Automated assessment of computer science competencies from student programs with Gaussian process regression. In *Proceedings of the 13th International Conference on Educational Data Mining (EDM 2020)*, pp. 555–560.

- Arastoopour Irgens, G., Dabholkar, S., Bain, C., Woods, P., Hall, K., Swanson, H., Horn, M., & Wilensky, U. (2020). Modeling and measuring high school students' computational thinking practices in science. *Journal of Science Education and Technology*, 29(1), 137–161. <https://doi.org/10.1007/s10956-020-09811-1>
- Bertolini, R., Finch, S. J., & Nehm, R. H. (2021). Testing the impact of novel assessment sources and machine learning methods on predictive outcome modeling in undergraduate biology. *Journal of Science Education and Technology*, 30(2), 193–209. <https://doi.org/10.1007/s10956-020-09888-8>
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education - implications for policy and practice*. In P. Kamylyis, & Y. Punie (Eds.). Publications Office of the European Union, Luxembourg. <https://doi.org/10.2791/792158>
- Bower, M., Wood, L. N., Lai, J. W., Highfield, K., Veal, J., Howe, C., Lister, R., & Mason, R. (2017). Improving the computational thinking pedagogical capabilities of school teachers. *Australian Journal of Teacher Education*, 42(3), 53–72. <https://doi.org/10.14221/ajte.2017v42n3.4>
- Braunstein, A. W., Lesser, M. N., & Pescatrice, D. R. (2008). The impact of a program for the disadvantaged on student retention. *College Student Journal*, 42(1), 36–40.
- Butler, D., & Leahy, M. (2021). Developing preservice teachers' understanding of computational thinking: A constructionist approach. *British Journal of Educational Technology*, 52(3), 1060–1077. <https://doi.org/10.1111/bjjet.13090>
- Chen, C.-H., Yang, S. J., Weng, J.-X., Ogata, H., & Su, C.-Y. (2021). Predicting at-risk university students based on their e-book reading behaviours by using machine learning classifiers. *Australasian Journal of Educational Technology*, 37(4), 130–144. <https://doi.org/10.14742/ajet.6116>
- Cheng, X., Ma, X.-Y., Luo, C., Chen, J., Wei, W., & Yang, X. (2021). Examining the relationships between medical students' preferred online instructional strategies, course difficulty level, learning performance, and effectiveness. *Advances in Physiology Education*, 45(4), 661–669. <https://doi.org/10.1152/advan.00234.2020>
- Chui, K. T., Fung, D. C. L., Lytras, M. D., & Lam, T. M. (2020). Predicting at-risk university students in a virtual learning environment via a machine learning algorithm. *Computers in Human Behavior*, 107, 105584. <https://doi.org/10.1016/j.chb.2018.06.032>
- Cutumisu, M., Adams, C., Glanfield, F., Yuen, C., & Lu, C. (2021). Using structural equation modeling to examine the relationship between pre-service teachers' computational thinking attitudes and skills. *IEEE Transactions on Education*, 1–7. <https://doi.org/10.1109/TE.2021.3105938>
- Eckles, J. E., & Stradley, E. G. (2012). A social network analysis of student retention using archival data. *Social Psychology of Education*, 15(2), 165–180. <https://doi.org/10.1007/s11218-011-9173-z>
- Figueiredo, J., Lopes, N., & García-Peñalvo, F. J. (2019). Predicting student failure in an introductory programming course with multiple back-propagation. *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, 44–49. <https://doi.org/10.1145/3362789.3362925>
- Friedman, J. H., Bentley, J. L., & Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3), 209–226. <https://doi.org/10.1145/355744.355745>
- Hooshyar, D., Pedaste, M., Yang, Y., Malva, L., Hwang, G.-J., Wang, M., Lim, H., & Delev, D. (2021). From Gaming to Computational Thinking: An Adaptive Educational Computer Game-Based Learning Approach. *Journal of Educational Computing Research*, 59(3), 383–409. <https://doi.org/10.1177/0735633120965919>
- Hung, C.-Y., Kuo, F.-O., Sun, J.C.-Y., & Yu, P.-T. (2013). An interactive game approach for improving students' learning performance in multi-touch game-based learning. *IEEE Transactions on Learning Technologies*, 7(1), 31–37. <https://doi.org/10.1109/TLT.2013.2294806>
- Jaipal-Jamani, K., & Angeli, C. (2017). Effect of robotics on elementary preservice teachers' self-efficacy, science learning, and computational thinking. *Journal of Science Education and Technology*, 26(2), 175–192. <https://doi.org/10.1007/s10956-016-9663-z>
- Ke, F., Moon, J., & Sokolikj, Z. (2020). Tracking Representational Flexibility Development through Speech Data Mining. *IEEE Frontiers in Education Conference (FIE), 2020*, 1–4. <https://doi.org/10.1109/FIE44824.2020.9273818.he>
- Kitsantas, A. (2013). Fostering college students' self-regulated learning with learning technologies. *Hellenic Journal of Psychology*, 10(3), 235–252.

- Kong, S.-C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers & Education, 151*, 103872. <https://doi.org/10.1016/j.compedu.2020.103872>
- Kwon, K., Ottenbreit-Leftwich, A. T., Brush, T. A., Jeon, M., & Yan, G. (2021). Integration of problem-based learning in elementary computer science education: Effects on computational thinking and attitudes. *Educational Technology Research and Development, 69*(5), 2761–2787. <https://doi.org/10.1007/s11423-021-10034-3>
- Lee, M., & Lee, J. (2021). Enhancing computational thinking skills in informatics in secondary education: The case of South Korea. *Educational Technology Research and Development, 69*(5), 2869–2893. <https://doi.org/10.1007/s11423-021-10035-2>
- Lemay, D. J., & Doleck, T. (2020). Predicting completion of massive open online course (MOOC) assignments from video viewing behavior. *Interactive Learning Environments, 1*–12. <https://doi.org/10.1080/10494820.2020.1746673>
- Li, Y., Pan, Y., Liu, W., & Zhang, X. (2018). An automated evaluation system for app inventor apps. In *Proceedings of the 2018 IEEE 16th International Conference on Dependable, Autonomic, and Secure Computing, the 16th International Conference on Pervasive Intelligence and Computing, and the 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)* (pp. 230–235). IEEE. <https://doi.org/10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00048>
- Lin, P.-H., & Chen, S.-Y. (2020). Design and evaluation of a deep learning recommendation based augmented reality system for teaching programming and computational thinking. *IEEE Access, 8*, 45689–45699. <https://doi.org/10.1109/access.2020.2977679>
- Lu, O. H., Huang, A. Y., & Yang, S. J. (2021). Impact of teachers' grading policy on the identification of at-risk students in learning analytics. *Computers & Education, 163*, 104109. <https://doi.org/10.1016/j.compedu.2020.104109>
- Malmi, L., Sheard, J., Kinnunen, P., & Sinclair, J. (2020). Theories and models of emotions, attitudes, and self-efficacy in the context of programming education. *Proceedings of the 2020 ACM Conference on International Computing Education Research, 36*–47. <https://doi.org/10.1145/3372782.3406279>
- Marbouti, F., Diefes-Dux, H. A., & Madhavan, K. (2016). Models for early prediction of at-risk students in a course using standards-based grading. *Computers & Education, 103*, 1–15. <https://doi.org/10.1016/j.compedu.2016.09.005>
- Min, W., Frankosky, M. H., Mott, B. W., Rowe, J. P., Smith, A., Wiebe, E., Boyer, K. E., & Lester, J. C. (2019). DeepStealth: Game-based learning stealth assessment with deep neural networks. *IEEE Transactions on Learning Technologies, 13*(2), 312–325. <https://doi.org/10.1109/ltl.2019.2922356>
- Mouza, C., Yang, H., Pan, Y.-C., Ozden, S. Y., & Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK). *Australasian Journal of Educational Technology, 33*(3). <https://doi.org/10.14742/ajet.3521>
- Pathak, A., & Pathak, S. (2020). Study on decision tree and KNN algorithm for intrusion detection system. *International Journal of Engineering Research & Technology, 9*(5), 376–381.
- Pi, Z., Zhang, Y., Zhou, W., Xu, K., Chen, Y., Yang, J., & Zhao, Q. (2021). Learning by explaining to oneself and a peer enhances learners' theta and alpha oscillations while watching video lectures. *British Journal of Educational Technology, 52*(2), 659–679. <https://doi.org/10.1111/bjet.13048>
- Pundlik, R. (2016). Comparison of Sensitivity for Consumer Loan Data Using Gaussian Naïve Bayes (GNB) and Logistic Regression (LR). *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, 120–124. <https://doi.org/10.1109/ISMS.2016.57>
- Qu, J. R., & Fok, P. K. (2021). Cultivating students' computational thinking through student-robot interactions in robotics education. *International Journal of Technology and Design Education, 1*–20. <https://doi.org/10.1007/s10798-021-09677-3>
- Qureshi, M. A., Khaskheli, A., Qureshi, J. A., Raza, S. A., & Yousufi, S. Q. (2021). Factors affecting students' learning performance through collaborative learning and engagement. *Interactive Learning Environments, 1*–21. <https://doi.org/10.1080/10494820.2021.1884886>
- Riestra-González, M., del Puerto Paule-Ruiz, M., & Ortin, F. (2021). Massive LMS log data analysis for the early prediction of course-agnostic student performance. *Computers & Education, 163*, 104108. <https://doi.org/10.1016/j.compedu.2020.104108>

- Rizvi, S., Rienties, B., & Khoja, S. A. (2019). The role of demographics in online learning: A decision tree based approach. *Computers & Education, 137*, 32–47. <https://doi.org/10.1016/j.compedu.2019.04.001>
- Russell, S., & Norvig, P. (1995). A modern, agent-oriented approach to introductory artificial intelligence. *Acm Sigart Bulletin, 6*(2), 24–26. <https://doi.org/10.1145/201977.201989>
- Urkude, S., & Gupta, K. (2019). Student intervention system using machine learning techniques. *International Journal of Engineering and Advanced Technology, 8*(6), 21–29. <https://doi.org/10.35940/ijeat.F1392.0986S319>
- Wing, J. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- Wiyono, S., & Abidin, T. (2019). Comparative study of machine learning knn, svm, and decision tree algorithm to predict students performance. *International Journal of Research-Granthaalayah, 7*(1), 190–196. <https://doi.org/10.29121/granthaalayah.v7.i1.2019.1048>
- Yang, J. C., & Quadir, B. (2018). Effects of prior knowledge on learning performance and anxiety in an English learning online role-playing game. *Journal of Educational Technology & Society, 21*(3), 174–185. <https://www.jstor.org/stable/26458516>
- Yang, T.-C., Chen, M. C., & Chen, S. Y. (2018). The influences of self-regulated learning support and prior knowledge on improving learning performance. *Computers & Education, 126*, 37–52. <https://doi.org/10.1016/j.compedu.2018.06.025>
- Yang, Z., Yang, J., Rice, K., Hung, J.-L., & Du, X. (2020). Using convolutional neural network to recognize learning images for early warning of at-risk students. *IEEE Transactions on Learning Technologies, 13*(3), 617–630. <https://doi.org/10.1109/tlt.2020.2988253>
- Zeng, M., Zou, B., Wei, F., Liu, X., & Wang, L. (2016). Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data. *IEEE International Conference of Online Analysis and Computing Science (ICOACS), 2016*, 225–228. <https://doi.org/10.1109/ICOACS.2016.7563084>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.