

Original Paper

Is Self-Rated Confidence a Predictor for Performance in Programming Comprehension Tasks?

Zubair Ahsan¹, Unaizah Obaidellah¹ and Mahmoud Danaee^{2*}

¹*Dept of Artificial Intelligence, Faculty of Computer Science and Information Technology, Universiti Malaya, 50603, Kuala Lumpur, Malaysia*

²*Dept of Social and Preventive Medicine, Faculty of Medicine, Universiti Malaya, 50603, Kuala Lumpur, Malaysia*

ABSTRACT

Studies on programming comprehension have focused largely on the type of reading strategies individuals employ. However, quite few programming comprehension studies have focused on the relationship between the self-rated confidence levels and the performance levels of the participants. In this study, our aim was to identify the effect of confidence levels among the participants as they attempt familiar programming questions. Our results indicate that due to familiarity, all participants generally show high confidence levels. High performers demonstrated self-rated high confidence levels as compared to low performers. However, the difference in confidence levels of high and low performers was found non significant. Furthermore, the confidence levels and the performance levels are weakly correlated indicating that confidence levels do not affect the performance levels of this set of participants on the types of questions tested. Moreover, the machine learning algorithms utilized to

*Corresponding author: Unaizah Obaidellah, unaizah@um.edu.my. This work was supported by the UM Research University Grant - Faculty Research (GPF095C-2020). The authors would like to thank the participants of this study for their participation and the reviewers for their recommendations and suggestions.

Received 30 July 2021; Revised 02 November 2021

ISSN 2048-7703; DOI 10.1561/116.00000041

© 2022 Z. Ahsan, U. Obaidellah and M. Danaee

classify the participants in this study showed potential based on their performance and confidence levels.

Keywords: Machine learning, expertise, confidence, programming comprehension, computer education.

1 Introduction

The world we know depends more on computer programming by each passing day. This makes understanding computer programming crucial to individual and societal development. Ultimately, it becomes even more important to understand how programmers learn and understand computer programs [26] in order to develop learning instructions and tools to address any issues that arise in these studies. Most program comprehension studies have focused on the reading strategies of the students to outline how high and low performers differ from each other in program comprehension tasks [26, 25, 23, 9, 24, 19, 2, 20, 5].

Experienced programmers' ability to write successful computer codes is dependent on their previous knowledge. Findings from studies on programming comprehension suggest that more organized prior knowledge and stronger concepts are possessed by these experienced programmers [9, 8, 6, 21]. Studies [5, 21] also suggest that during program comprehension the related schemas are evoked. These schemas [7] are structured mental representation of concepts that are developed over time through experience. Each time an individual is presented with a new problem, these schemas are used to assist problem solving or decision making.

It has hence been established that experienced (experts) and non-experienced or less experienced (novices) programmers' underlying mental schemas differ as more experienced programmers process and perform the tasks faster than their counterparts. However, identifying the confidence levels of programmers as they perform program comprehension tasks is still an area to be explored, especially, with the incorporation of the effect of mental schemas as they perform program comprehension tasks. In other words, in the context of success in coding it is yet unknown to what extent prior knowledge influences the levels of self-rated confidence exhibited by a programmer of varying degree of expertise. In this case, prior knowledge is identified as the amount of experience, existing and relevant knowledge an individual holds for a specific domain. The identification of this effect would provide deeper understanding about its role in key decision making and problem-solving processes including motivation, communication, challenges, and being in the state of flow.

In the literature, self-confidence termed as self-efficacy, perceived ability and perceived competence describes "an individual's perceived ability or degree of

belief one possesses to accomplish a certain level of performance” [11]. However, Bandura [3] differentiates between self-confidence and self-efficacy by defining the former as “firmness or strength of belief without a specific direction”, while the latter as a goal set together with a belief. A wide range of definitions and concepts on self-confidence studies revealed constant importance to investigate its effects on factors such as motivation and performance.

Self-rated confidence can also be termed as the degree of optimism an individual identifies related to his/her performance upon completing a specific task. The notion of believe about one’s competence is consistent with [10] as a cognitive mechanism mediating motivation, thought patterns, emotional reactions, and behavior [3]. According to Bandura [3], performance monitoring on success offers greater encouragement and confidence, than if the focus is given on failures an individual experience. Furthermore, diverse studies [15, 22] collectively provided evidence that people’s perception of their performance capability significantly influences their motivational behavior [3]. In the past, Kruger and Dunning [14] pointed out how unskilled people overestimate their competence without realizing it and that focusing on improving their skills and making them realize their incompetence can help them. Hence, the identification of self-rated confidence along with performance on programming tasks is important as it will help educators understand early learners’ confidence in relation to their performance. This is so that appropriate persuasive techniques and goal settings can be offered to positively influence the potential of every learner’s confidence, motivation, and behavior. Consequently, better teaching and learning strategies can be designed to aid these early learners. This includes applying techniques and findings gained from this study in designing and developing personalised learning systems.

The aim of this research is to analyze the performance and the self-rated confidence levels of novice programmers on program comprehension tasks. The proposed work attempts to identify (RQ1) the relationship between confidence levels and performance levels of the participants. In this work, self-rated confidence level refers to the degree of belief one identifies themselves when judging their own ability and capacity in solving a programming problem, while performance levels is identified as the total score an individual attains after the completion of all programming tasks. Furthermore, given the varying degree of experience these novice programmers exhibit (as there will be individuals classified as high performers vs low performers – see Section Methodology: Analysis), (RQ2) what groups can be classified from the confidence levels and the performance levels of the participants? Finally, (RQ3) how well can selected machine learning algorithms classify these participants based on their performance? The classification using machine learning is included to see if participants can be classified into high and low performers’ groups along with their confidence levels. To date, these kind of predictions for a programming comprehension study have not been carried out, hence, this research aims to fulfill this research gap.

In this study, as we target early learners who have been assessed in the introductory programming course in their university, we hypothesize that high performers will have higher confidence levels than low performers as they attempt the programming questions. This is because we expect them (high performers) to have a general idea of their levels of performance and expect their confidence levels to reflect such judgement. It is further hypothesized that the machine learning algorithms will achieve a considerably good accuracy in classifying these performers. Taken together, these analyses will inform about the relationship between the confidence levels and the performance levels of the participants on programming comprehension tasks.

2 Related Work

Programming comprehension studies have always been influenced by psychological aspects as we see Soloway *et al.* [26] using the term “cognitive fit” to elaborate how subjects have a certain preference for a program comprehension strategy. This was extended by Soloway and Ehrlich [27] as they used the terms plan-like and unplan-like for computer programs referring to the structure of the programs as they are generally taught (plan-like) and other correct implementations of the same programs (unplan-like). These studies structured the questions in a way to evaluate whether programming problems are associated with the concepts that lie in individuals’ minds (or the underlying mental schemas). Rist [23] contributed on the strategies that subjects use during program comprehension and suggested that a bottom-up and backward strategy is employed while learning programming (creating mental schemas), and a top-down and forward strategy is employed while retrieving (using mental schemas). Détienne and Soloway [9] asserted that often one comprehension strategy is not enough during program comprehension and subjects often must rely on more than one strategy to perform the program comprehension tasks.

The study by Obaidallah *et al.* [21] suggested that prior knowledge is indeed helpful to the subjects as they solve new problems. Several other studies [1, 2, 4, 13, 16, 19] and [20] have successfully incorporated program comprehension theories and drawn significant results. However, these studies do not include confidence levels of the subjects leaving this aspect to be explored.

The study by Sharif and Maletic [24] found that high levels of ability in UML design result in higher confidence, contributing a strong positive correlation between performance and confidence. However, a study by McChesney and Bond [18] drew a comparison between dyslexia programmers’ and typical programmers’ confidence levels, however, no significant difference between the groups was found. Another study, by Duan *et al.* [5] tested whether confidence is a good predictor of performance on model comprehension tasks. They [5] found that confidence can be a good predictor for the correctness of the tasks. A study by Doukakis *et al.* [10] focused on measuring students’

confidence in algorithms and programming. They [10] found that secondary students who intend to take a technological direction during tertiary education showed higher levels confidence to solve problems and design algorithms. It was also reported that tertiary level students who had taken relevant courses during their secondary education showed higher confidence levels than the tertiary level students who did not enroll in the relevant courses. This further indicates that prior knowledge (to some extent) causes them to have higher confidence levels. The dearth of studies relating confidence levels of the students with their performance in program comprehension tasks clarifies the gap that research in this area needs to be carried out. Based on the literature discussed so far, a possible new direction that can be taken is by obtaining the self-rated confidence level of the subjects as they perform familiar program comprehension tasks to see the effect of the mental schemas (prior knowledge) on their confidence.

Furthermore, we are interested to investigate if machine learning algorithms can classify participants based on their performance as well as confidence levels. Studies [1, 12, 17] on program comprehension used machine learning algorithms to predict programmer expertise and task difficulty. However, these studies do not integrate the mental schema aspect, nor do they integrate the confidence levels of the participants. This introduces another gap to be filled. As discussed, it is important to understand the effect of mental schemas on individuals' confidence levels and its relation to their performance. Consequently, the proposed research integrates mental schema theory as its theoretical framework to carry out a study on the confidence levels of novice programmers in relation to their performance as well as using machine learning algorithms for classification.

3 Methodology

This section provides a review about the methods adopted for the proposed study including an overview about the participants, materials for the tasks, procedure taken for data collection and analysis methods applied in the research.

3.1 Participants

First year undergraduate students majoring in computer science were chosen from a public university in Asia. The only pre-requisite for participants in the current study for the students was to have undertaken the Fundamentals of Programming course. This was to ensure that they are familiar with the concepts and the questions that they are going to be presented with in the study. A total of 60 students took part in the study, male and female included, between 18 and 25 years of age ($M_{\text{age}} = 19.3$ years, $SD_{\text{age}} = 0.57$). All participants were beginners, and hence, considered as novices.

Q: PRINT INTEGERS 1 TO 10. CHOOSE ALL CORRECT ANSWERS.

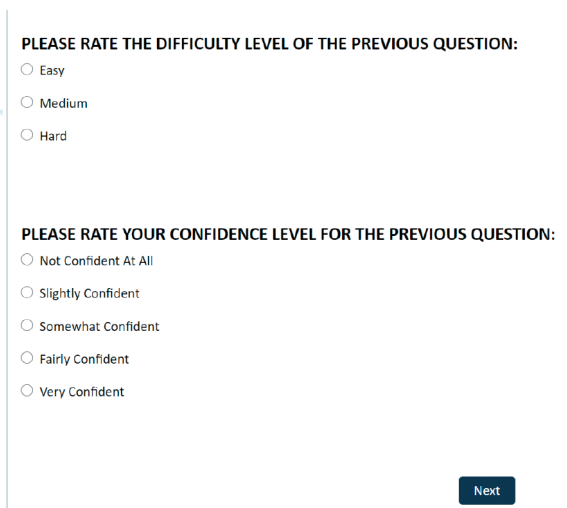
<input type="checkbox"/>	<pre>for(int i=1; i<10; i++) { System.out.println(i); }</pre>	<input type="checkbox"/>	<pre>for(int i=1; i<11; i++) { System.out.println(i); }</pre>
<input type="checkbox"/>	<pre>for(int i=1; i<=10; i++) { System.out.println(i); }</pre>	<input type="checkbox"/>	<pre>for(int i=0; i<9; i++) { System.out.println(i); }</pre>

[Next](#)

Figure 1: Example of a question. The bottom left is the plan-like correct option and the top right option is the unplan-like correct option.

3.2 Materials

There were two types of questions; Selection and Iteration, each type included two questions of each level of difficulty; easy, medium, and hard – making a total of twelve questions. The easy questions were short codes with only one statement containing the functionality for selection and iteration type questions. The medium questions included either two selection statements or while loops for increased difficulty. The hard questions had increased sophistication using nested selection and loop statements. The content of these questions was inspired from the textbooks and notes used to teach the students weeks before the data was collected. In a way, the students were familiar with the tasks presented to them. The questions had a problem statement and four code snippets as options with at least two correct choices where one was plan-like and the other unplan-like, see Figure 1. In Figure 1, the bottom left is the plan-like correct option usually seen in the textbooks to satisfy the requirements in the problem statement, whereas the top right option is the unplan-like correct option that also achieves the desired result but is not seen in the textbooks and/or taught by the instructors. This was to ensure that the plan-like code snippet can evoke mental schemas and to see whether these schemas can help them identify the correct implementation in the unplan-like code snippets provided to them. A locally hosted web-application was developed to present the questions to the participants. A post-survey questionnaire was included at the end of a session to obtain the demographic details of the students.



PLEASE RATE THE DIFFICULTY LEVEL OF THE PREVIOUS QUESTION:

- Easy
- Medium
- Hard

PLEASE RATE YOUR CONFIDENCE LEVEL FOR THE PREVIOUS QUESTION:

- Not Confident At All
- Slightly Confident
- Somewhat Confident
- Fairly Confident
- Very Confident

Next

Figure 2: Rate difficulty and confidence level.

3.3 Procedure

The participants were individually scheduled to come to a dedicated lab and attempt all questions in one session each. The participants would first read the instructions, watch an example video about the tasks, and perform an example task before attempting the main questions. The questions were presented in a randomized order to avoid order effect. The participants were asked to rate their confidence and difficulty level at the end of each question, see Figure 2. The questions had no time-limit and the students were told to take as much time as they need. However, most participants completed their sessions in less than 25 minutes. After answering all twelve questions they were asked to provide information regarding demographic details. Each participant received course credit for their participation.

3.4 Analysis

Statistical tests will be carried out to understand the relationship between the confidence levels and the performance of the students. Supervised machine learning algorithms are considered to classify the participants into groups of performance (high vs low). These machine learning algorithms will be validated using weighted F -accuracy. The data was exported from the database of locally hosted web-application used in the experiment.

Table 1: Participants' performance.

No. Students	Correct answers	Percentage of 12	Grade	Category
2	5	42%	D+	FAIL
9	6	50%	C	PASS
5	7	58%	C+	
10	8	66%	B	GOOD
16	9	75%	A-	DISTINCTION
11	10	83%	A	
6	11	91%	A	
1	12	100%	A	

Note: Percentage of 12 in (column 3) is calculated by dividing the number of correct answers (column 2) by 12 (total number of questions).

3.4.1 Data Preparation

The number of correct answers were tallied to acquire the total score of the participants. Only fully correct answers are considered in this study as correct answers. This means that all chosen answers must match those defined by the experimenter. The confidence levels of the student were also exported and tallied with all five confidence levels. These tallies will be used for visualization, see Figures 3–14. Furthermore, a mean of all participants for each question was calculated by converting the confidence levels into numbers (1: Not Confident, 5: Very Confident). For the purpose of machine learning classification, mean confidence levels for each participant across all questions were separated along with their expertise levels (high and low), see Defining High and Low Performers.

3.4.2 Defining High and Low Performers

In order to determine the high and low performers, the grading criteria from the university where the data was collected was utilized. Table 1 shows that 34 students received an A- grade and above which is 75% or more correct answers out of 12 questions. These are labeled as “High Performers”. The rest of the students have 8 or less correct answers out of 12 questions with a grade B and below, and these students are labeled as “Low Performers”.

3.4.3 Statistical Tests

A Pearson correlation coefficient test will be carried out using the confidence levels of each student for each question, their overall score, and their levels

of expertise (high and low) as established in Defining High and Low Performers section. This test will offer insight into the relationship between the performance levels and the confidence levels of the participants.

3.4.4 Supervised Machine Learning

The dataset contained the mean confidence levels of the students for each question, as described in the Data Preparation section, and their levels of expertise (high and low) as established in Defining High and Low Performers section. This data set was divided into training and testing set with 60% (data of 36 participants) and 40% (data of 24 participants) of the total data respectively. The training set was labeled manually into four groups; Less Confident Low Performer (LC), More Confident Low Performer (ML), Less Confident High Performer (LH), and More Confident High Performer (MH). As there were five confidence levels, for the purpose of machine learning classification, they were transformed into two by calculating the mean across all questions for each participant. More confident students had a mean confidence level more than 3 on all questions (accumulated), whereas less confident students had a mean confidence of less than 3 on all questions (accumulated). The training set was put into MATLAB's classification learner with 5-fold cross validation and the model was generated. Three classifiers were used in the training; Tree (Fine), Support Vector Machine (Quadratic), and KNN (fine). The obvious choice here was Tree (Fine) as it is considered to be the most appropriate for this kind of dataset due to its ability to classify using branches, however, the other two classifiers also showed high accuracies during prediction and they are included for a comparison. For validation, weighted F1-accuracy will be calculated. Other metrics including weighted precision and weighted recall will be presented.

4 Results

As previously described in Section Methodology: Analysis, the results will be presented in this section. Firstly, visualizations of metrics including confidence levels, performance, and question-dependent performance of the participants will be presented. Secondly, accumulated confidence levels for all questions of high and low performers will be presented for each question. This will be followed by Pearson correlation coefficient test results. Lastly, supervised machine learning training model, prediction, and F-score validation will be presented.

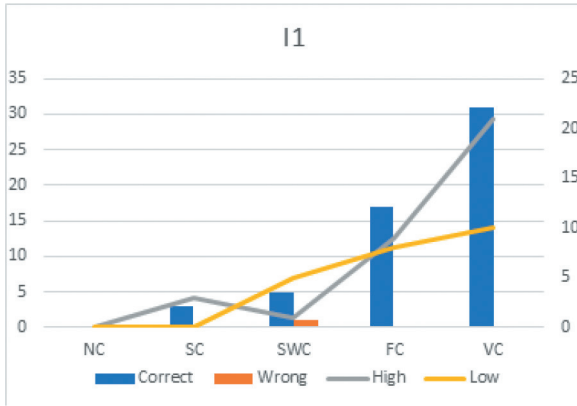


Figure 3: Iteration 1 (I1) – Talled participants for each confidence level (NC – Not Confident, SC – Slightly Confident, SWC – Somewhat Confident, FC – Fairly Confident, VC – Very Confident). Correct and wrong refers to the number of participants who scored correctly or wrongly.

4.1 Confidence Levels and Performance

In this section, three metrics are visualized for each question; Selection 1–6 (S1–S6) and Iteration 1–6 (I1–I6). The first metric is the number of participants who answered correctly or wrongly for each question, and this is represented in blue and orange bars respectively. The second metric is the confidence level selected by the participants for each question. This metric contains all participants who answered correctly or wrongly for each question. The third metric is the number of participants in high and low performers group (overall performance on all questions) in each confidence category for each question. High performers are represented by grey lines and low performers are represented by yellow lines. Together, the visualizations represent the number of high and low performers in each confidence category for each question and shows question-dependent performance of the same participants.

Figure 3 shows that both high and low performers (respectively indicated by gray and yellow lines) were generally confident for this question. This could possibly be because of its ease as 59 out of 60 participants were able to score correctly. The participant who did not score correctly for this question rated their confidence as somewhat confident.

Figure 4 shows that both sets of high and low performers particularly showed a high confidence but were unable to answer correctly. This could possibly be because of the unplan-like options managing to successfully trick them as this question was generally perceived as an easy question. Participants who scored correctly generally chose either somewhat confident or fairly confident rating.



Figure 4: Iteration 2 (I2) – Talled participants for each confidence level (NC – Not Confident, SC – Slightly Confident, SWC – Somewhat Confident, FC – Fairly Confident, VC – Very Confident). Correct and wrong refers to the number of participants who scored correctly or wrongly.

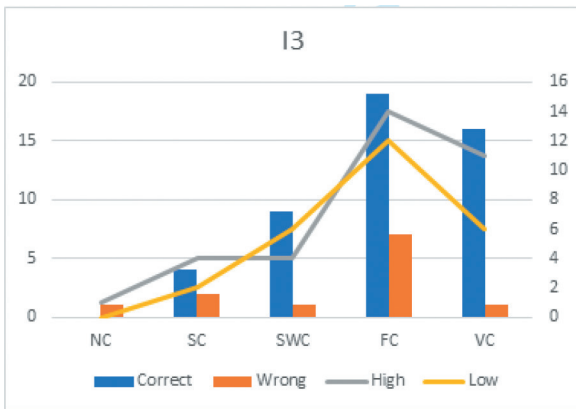


Figure 5: Iteration 3 (I3) – Talled participants for each confidence level (NC – Not Confident, SC – Slightly Confident, SWC – Somewhat Confident, FC – Fairly Confident, VC – Very Confident). Correct and wrong refers to the number of participants who scored correctly or wrongly.

Figure 5 shows that the majority of high and low performers who showed a higher confidence were able to answer correctly. Whereas most participants who answered wrongly chose fairly confident as their confidence level. There were more low performers who chose somewhat confident than high performers and almost all the participants who chose somewhat confident answered correctly.

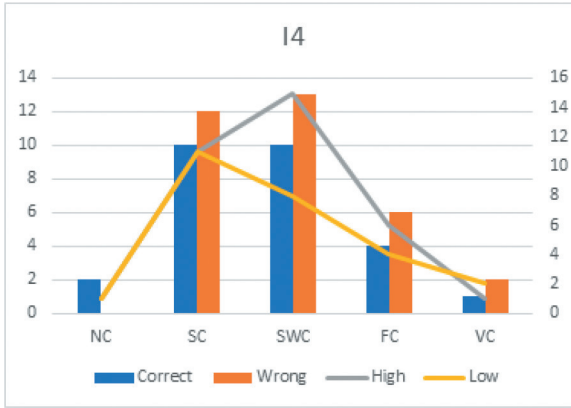


Figure 6: Iteration 4 (I4) – Tallied participants for each confidence level (NC – Not Confident, SC – Slightly Confident, SWC – Somewhat Confident, FC – Fairly Confident, VC – Very Confident). Correct and wrong refers to the number of participants who scored correctly or wrongly.

Figure 6 shows that the majority of the high and low performers chose either a confidence level of slightly confident or somewhat confident and more than half of them answered wrongly. This could possibly be because of the perceived general difficulty of this question. A considerable number of them chose fairly confident as well and more than half of those also answered wrongly. It is worth noting that two students (one high and one low performer) who chose not confident as their confidence level both answered correctly. More low performers than high performers chose very confident.

Figure 7 shows that the majority of high and low performers who showed a higher confidence were able to answer correctly. Whereas the participants who answered wrongly chose either slightly confident, somewhat confident or fairly confident as their confidence level.

Figure 8 shows that the majority of high and low performers chose somewhat confident as their confidence level and more than half of them scored wrongly. Participants who chose slightly confident or not confident, the majority of them too answered wrongly and the majority of these participants were high performers. Participants who chose fairly confident, a majority of them answered correctly and were high performers whereas the only participant that chose very confident answered wrongly and was a high performer. This was a hard question and participants were expected to be challenged by the question.

A commonality found in all iteration type questions for the majority of low performers is the choice of somewhat confident as their confidence in all questions as compared to the high performers except for Iteration 4 question.

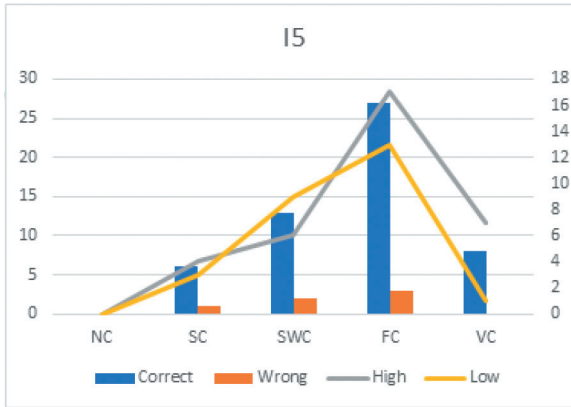


Figure 7: Iteration 5 (I5) – Talled participants for each confidence level (NC – Not Confident, SC – Slightly Confident, SWC – Somewhat Confident, FC – Fairly Confident, VC – Very Confident). Correct and wrong refers to the number of participants who scored correctly or wrongly.

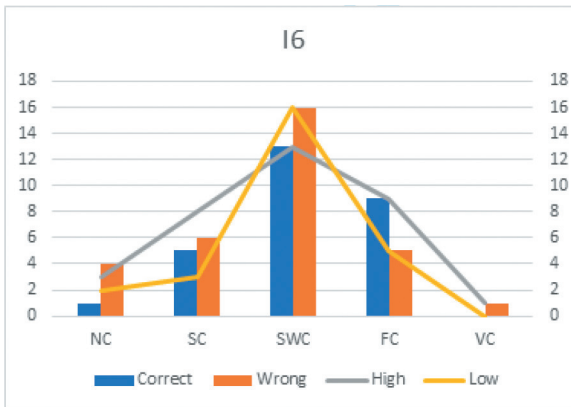


Figure 8: Iteration 6 (I6) – Talled participants for each confidence level (NC – Not Confident, SC – Slightly Confident, SWC – Somewhat Confident, FC – Fairly Confident, VC – Very Confident). Correct and wrong refers to the number of participants who scored correctly or wrongly.

This was possibly due to the perceived general difficulty of the question where all participants generally exhibited a lower level of confidence. The high performers tended to choose either fairly confident or very confident and it was reflected in their performance as well. Furthermore, the questions where a number of the low performers showed a high confidence is also reflected in correct answers to those questions. This is consistent for all iteration type



Figure 9: Selection 1 (S1) – Talled participants for each confidence level (NC – Not Confident, SC – Slightly Confident, SWC – Somewhat Confident, FC – Fairly Confident, VC – Very Confident). Correct and wrong refers to the number of participants who scored correctly or wrongly.

questions except for Iteration 2, as the unplan-like answer managed to trick the participants causing high and low performers to perform poorly even with its perceived general ease.

Figure 9 shows that the majority of high and low performers chose either fairly confident or very confident and most of them answered correctly. A few of them chose somewhat confident but most of them answered wrongly. Three out of five who chose slightly confident answered correctly. This was an easy question and participants were expected to perform well.

Figure 10 shows that the majority of high and low performers chose either fairly confident or very confident and most of them answered correctly. A few of them chose somewhat confident and all of them answered correctly as well. Three out of five who chose slightly confident answered correctly. One participant who chose not confident was a high performer and answered wrongly. This was also an easy question and participants were expected to perform well.

Figure 11 shows that the majority of high and low performers chose fairly confident and only half of them answered correctly. A few of them chose very confident and most of them answered correctly. It is worth noting that more low performers chose very confident for this question making this particular question a special case. A few of the high and low performers also chose somewhat confident and more than half answered correctly. Three out of five who chose slightly confident answered correctly.

Figure 12 shows that the majority of high and low performers chose fairly confident and a vast majority of them answered correctly. Most high performers chose very confident whereas very few low performers chose this confidence level and only one of them answered wrongly. A few of them chose somewhat



Figure 10: Selection 2 (S2) – Talled participants for each confidence level (NC – Not Confident, SC – Slightly Confident, SWC – Somewhat Confident, FC – Fairly Confident, VC – Very Confident). Correct and wrong refers to the number of participants who scored correctly or wrongly.

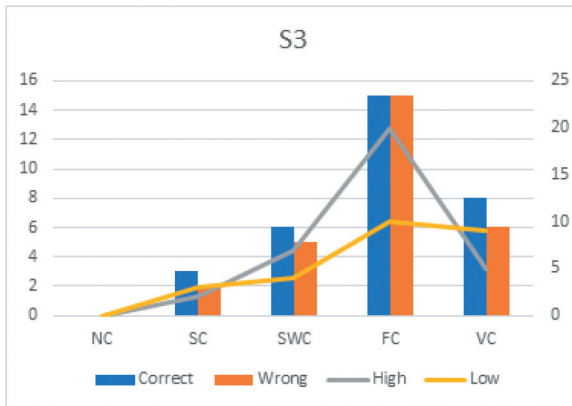


Figure 11: Selection 3 (S3) – Talled participants for each confidence level (NC – Not Confident, SC – Slightly Confident, SWC – Somewhat Confident, FC – Fairly Confident, VC – Very Confident). Correct and wrong refers to the number of participants who scored correctly or wrongly.

confident and more than half answered correctly. Six participants who chose slightly confident all answered correctly.

Figure 13 shows that the majority of high and low performers chose fairly confident and a vast majority of them answered correctly. Most high performers chose very confident whereas several low performers chose very confident and only one answered wrongly. A few of them chose somewhat confident and half

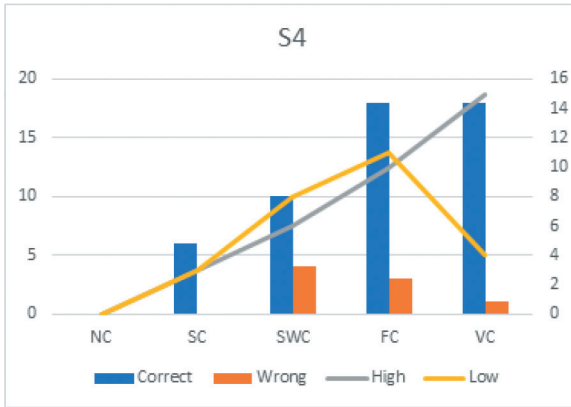


Figure 12: Selection 4 (S4) – Talled participants for each confidence level (NC – Not Confident, SC – Slightly Confident, SWC – Somewhat Confident, FC – Fairly Confident, VC – Very Confident). Correct and wrong refers to the number of participants who scored correctly or wrongly.

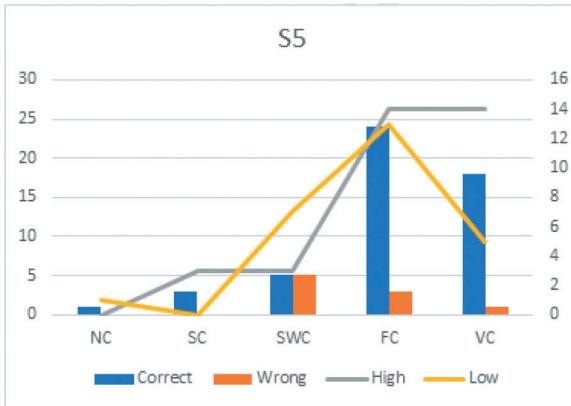


Figure 13: Selection 5 (S5) – Talled participants for each confidence level (NC – Not Confident, SC – Slightly Confident, SWC – Somewhat Confident, FC – Fairly Confident, VC – Very Confident). Correct and wrong refers to the number of participants who scored correctly or wrongly.

of them answered correctly. Four participants who chose slightly confident were high performers and all of them answered correctly whereas only one participant who chose not confident was a low performer and answered correctly.

Figure 14 shows that the majority of high performers and several low performers chose very confident and all of them answered correctly. A few of them chose fairly confident and all of them answered correctly except one. Ten out of eleven participants (high and low performers alike) who chose somewhat

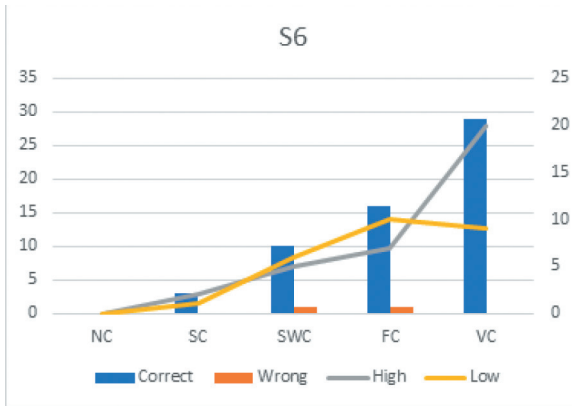


Figure 14: Selection 6 (S6) – Talled participants for each confidence level (NC – Not Confident, SC – Slightly Confident, SWC – Somewhat Confident, FC – Fairly Confident, VC – Very Confident). Correct and wrong refers to the number of participants who scored correctly or wrongly.

confident answered correctly. Three participants (two high performers and one low performer) chose slightly confident and all answered correctly.

What all selection type questions have in common is that high performers are generally all more confident as they self-rated their confidence as “very confident”; for S1 there are 16 high performers and 6 low performers, for S2 there are 17 high performers and 10 low performers, for S3 there are 5 high performers and 10 low performers, for S4 there are 15 high performers and 4 low performers, for S5 there are 14 high performers and 5 low performers, and for S6 there are 20 high performers and 9 low performers. Conversely, majority of the low performers generally indicated higher confidence for selection type questions as compared to iteration type questions. As more low performers chose fairly confident in all selection type questions and in Selection 3 the number of low performers who chose very confident was higher than the number of high performers.

4.2 Accumulated Confidence Levels (Mean) For Each Question

In this subsection, the confidence levels of all participants were converted into numbers (1: Not Confident, 5: Very Confident) and an accumulated mean was calculated for high and low performers, as established in Methodology section, for each question. Figure 15 shows that all participants generally maintained a high confidence level except for I4 (Iteration 4) and I6 (Iteration 6) where the mean confidence levels for both high and low performers are less than 3 (3 represents Somewhat Confident). It is also worth noting that the participants generally showed a higher level of confidence for selection type

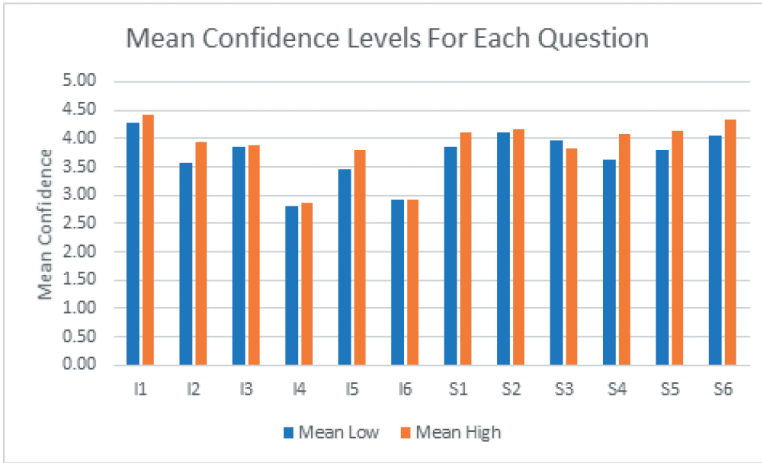


Figure 15: Accumulated confidence (mean) for each question.

questions (S1–S6) than the iteration type questions (I1–I6). This could possibly be due to the overall ease and simplicity of the selection type questions.

A Pearson correlation coefficient test was carried out by splitting the data in two groups; high and low performers, where their mean confidence was tested against their total number of correct answers in all questions, against their total number of correct answers in Iteration-type questions, and against their total number of correct answers in Selection-type questions. The Pearson correlation coefficient between mean confidence of low performers and total number of correct answers in all questions ($r(24) = 0.137, p = 0.503$) indicates a weak positive correlation. Moreover, a negative correlation ($r(24) = -0.075, p = 0.717$) was found between mean confidence of low performers and total number of correct answers in Iteration-type questions. Furthermore, a weak positive correlation ($r(24) = 0.212, p = 0.298$) was found between mean confidence of low performers and total number of correct answers in Selection-type questions. The Pearson correlation coefficient between mean confidence of high performers and total number of correct answers in all questions ($r(32) = 0.168, p = 0.343$) indicates a weak positive correlation. Moreover, between mean confidence of high performers and total number of correct answers in Iteration-type questions, a weak positive correlation ($r(32) = 0.193, p = 0.275$) was found. Furthermore, a negative correlation ($r(32) = -0.010, p = 0.957$) was found between mean confidence of high performers and total number of correct answers in Selection-type questions. These results indicate that for these set of questions and for these set of participants, confidence and performance are not correlated. This means that students' confidence is not an indicator of how they actually perform as the performance

Table 2: Accuracies of the machine learning classifiers.

Classifier	Accuracy
Tree (Fine)	97.2%
Support vector machine (Quadratic)	91.7%
KNN (Fine)	91.7%

Table 3: F1-accuracy, precision and recall of the machine learning classifiers.

Classifier	Weighted F1-accuracy	Weighted precision	Weighted recall
Tree (Fine)	96.9%	97.3%	97.2%
SVM (Quadratic)	90.8%	92.6%	91.7%
KNN (Fine)	91.3%	91.7%	91.6%

of the participants is somewhat independent of their self-rated confidence levels.

4.3 Supervised Learning

After determining the high and low performers, as mentioned in Methodology section, confidence levels were also converted into two by calculating the mean across all questions for each student. As there were five confidence levels collected (1: Not Confident, 2: Slightly Confident, 3: Somewhat Confident, 4: Fairly Confident, 5: Very Confident), it was decided that students who show a mean of confidence of more than 3 (somewhat confident) will be considered as more confident students for the purpose of this analysis. Categorizing the participants into high and low performers as well as more and less confident performers makes it simplistic and easier to label. Accordingly, there will be four groups; More Confident High Performer (MH), Less Confident High Performer (LH), More Confident Low Performer (ML), and Less Confident Low Performer (LL). The classifiers used in the training were Tree (Fine), Support Vector Machine (Quadratic), KNN (Fine) as mentioned in the Methodology section. Table 2 shows the accuracies of the machine learning classifiers on the training set.

Weighted F1-accuracy was utilized for the purposes of validation as the classes in the data were not equally distributed. Table 3 presents the weighted F1-accuracy, weighted precision, and weighted recall for each of the three classifiers used on the training set.

Consequently, Tree (Fine) was chosen, as it showed the highest accuracy in Table 2 as well as highest Weighted F1-accuracy, Weighted Precision, and

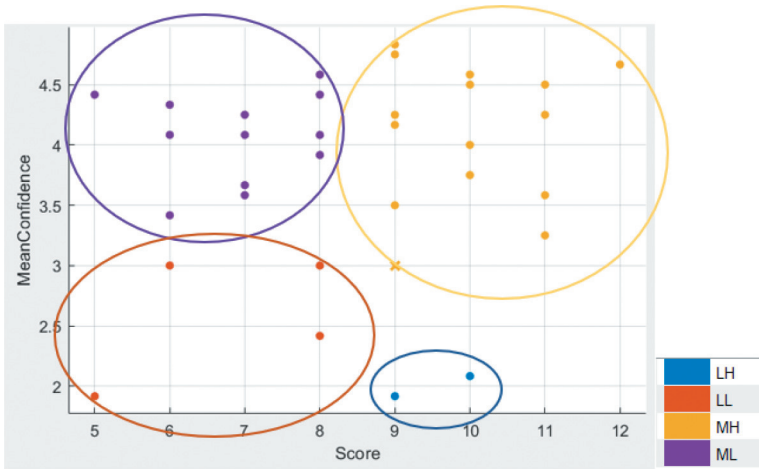


Figure 16: Training model (Less confident-high performer: 2, Less confident-low performer: 4, More confident-high performer: 15, More confident-low performer: 13).

Weighted Recall. Only two predictors were used; Mean Confidence Level and participant’s overall Score out of 12. Figure 16 shows the training model.

The prediction can be seen in Figure 17. The prediction was not 100% accurate as one participant who is predicted in Less Confident Low Performer (LL) group has a mean confidence of just above 3, whereas anyone with a mean confidence higher than 3 should be a more confident performer. Another wrong prediction was a participant classified into Less Confident High Performer (LH) group, whose mean confidence level is just above 3 as well.

5 Discussion

5.1 Relation between Confidence Levels and Performance

As research question 1 is to identify the relation between the confidence levels and the performance of the students, the results found in Confidence Levels and Performance in the Results section suggest that generally the participants (regardless of performance levels) exhibited high confidence throughout all twelve questions. This is potentially due to the reason that the participants were familiar with the questions presented to them during data collection. Familiarity, in this case, gave them more confidence to perform well on the tasks, which, as the results suggest, did happen. As out of 60 students, 34 were high performers and that did not include the 10 students who scored a B grade in this experiment. This possibly indicates that most of the students had robust mental schemas indicating effective previous knowledge on the topic of interest.

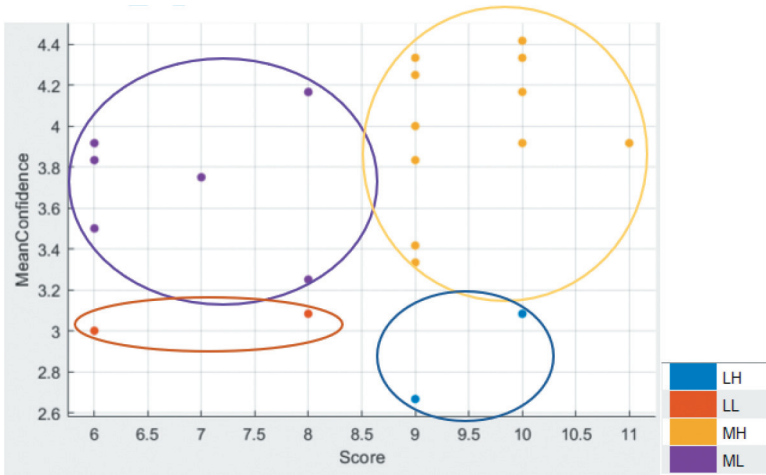


Figure 17: Prediction on testing set.

Thus, the strategies they undertook along with the edge that their confidence provided ended up in a good level of performance. For some questions such as I2, I4, S3, and S6, where the majority of participants chose high confidence but answered wrongly, it could possibly be because of the unplan-like code options that were given to them, and at times, they failed to properly use their mental schemas to identify the correct code options. It is worth mentioning that high performers exhibited higher confidence in both types of questions (iteration and selection) and low performers showed higher confidence for selection type questions than they did for iteration type questions. The higher confidence of the low performers was also reflected in their ability to score better on selection type questions than they did on iteration type questions.

The results found in the Accumulated Confidence Levels (mean) for Each Question of the Results section also suggest that the mean confidence levels were higher on the selection type questions than the iteration type questions. This is possibly due to the general ease of the selection statements, as they are easier to follow due to their straightforward structure. This general ease could have affected their confidence positively even before they attempted the question and seemingly continued after they had attempted them. This is also supported by the fact that participants performed better on the selection type questions than the iteration type questions. However, through the findings by Pearson correlation coefficient test as reported in the Results section, only weak positive correlations and negative correlations were found for high and low performers between their respective mean confidence and total number of correct answers in all questions, mean confidence and total number of correct answers in Iteration-type questions, and mean confidence and total number of correct answers in

Selection-type questions. Furthermore, no significant difference is found in these results. This indicates that the performance of the students on programming questions is weakly related to the self-rated confidence levels by the students. This ultimately means that students should rather practice writing programming codes than trusting their confidence in their ability to write programming codes. Studies in the literature as reviewed earlier relate confidence with UML tasks in the study by Sharif and Maletic [24] and with model comprehension in the study by Duan *et al.* [5] both of whom reported a positive relation between performance and confidence. However, these studies cannot be compared due to the apparent differences in the type of questions presented to them. These contradicting findings warrant further investigation in future studies.

The findings for this research question are consistent with the findings by Doukakis *et al.* [10] in terms of students with prior knowledge showing higher levels of confidence. In this study, however, all participants were pursuing computer science at the undergraduate level, and hence, they all exhibited higher levels of confidence. Furthermore, due to general ease, higher confidence and higher performance were recorded for selection type questions than iteration type questions. However, higher confidence does not translate into higher performance as supported by the statistical results presented in this study. This finding is consistent with the findings by Kruger and Dunning [14], as overestimation of abilities is a problem with the incompetent.

5.2 *Classifying Participants into Groups*

As research question 2 is to classify the participants into groups, the results found in the Supervised Learning results section suggest that the Tree (Fine) classifier classified participants into four predefined groups. The results show twenty participants in More Confident Low Performer (ML) group, which means that their mean confidence was more than 3 and they had 8 or less than 8 correct answers out of 12. Six participants in Less Confident Low Performer (LL) group means that their mean confidence was equal to or less than 3 and they had 8 or less than 8 correct answers out of 12. Thirty participants in More Confident High Performer (MH) group means that their mean confidence was more than 3 and they had 9 or more than 9 correct answers out of 12. Four participants in Less Confident High Performer (LH) group means that their mean confidence was 3 or less than 3 and they had 9 or more than 9 correct answers out of 12. This indicates that although most participants showed high levels of confidence, thirty of them were high performers which makes up to half of the total number of participants. Twenty students who exhibited high confidence but were low performers seemed to have overestimated their ability to attempt the questions correctly. Six participants who were less confident and were low performers seemed to have correctly estimated their ability to attempt the questions. The remaining four students who exhibited

less confidence but were high performers seemed to have underestimated their ability to attempt the questions.

5.3 Performance of Machine Learning Classifiers

As research question 3 is about the performance of the selected machine learning classifiers used in this research, the results shown in Supervised Learning, Results section suggest that all three machine learning classifiers performed well. However, as Tree (Fine) showed the highest accuracy of 97.2% with 96.9% weighted F1-accuracy, 97.3% weighted precision, and 97.2% weighted recall, it was chosen to classify the participants in the testing set. During prediction, the Tree (Fine) classifier classified two participants wrongly due to very small margins as both of these participants had a mean confidence of 3.08 out of 5 and should have been classified as more confident performers, instead, they were classified as less confident performers. This problem occurs with machine learning classifiers when the dataset is as small as it were in this study. It is possible that this error could have been avoided had the dataset included data from over 200 participants.

6 Conclusion

6.1 Implications

Findings from this research establish that the prior knowledge seems to influence the confidence levels and the performance of the students. This is validated through visualizations of the chosen confidence levels for each question by all participants, mean confidence levels of all high and low performers for each question, and by the formation of four groups as devised and classified with the use of supervised machine learning. However, this research work indicates that higher self-rated confidence on programming questions does not translate into higher performance. This is supported and validated through the Pearson correlation coefficient test. Hence, this research suggests that prior knowledge causes the individuals to exhibit more confidence regardless of how robust their prior knowledge is.

This means that educators are encouraged to enforce comprehensive programming practice to the students so that they gain more experience and improve their programming ability. Furthermore, educators should also encourage their students to assess their confidence levels periodically so that if a low performing student has high confidence, they can be informed about the reality of their ability. This will in turn allow them to work harder and motivate them to practice and learn more. However, if a high performing student has low confidence, they can be informed about their ability to increase their self-confidence

but also guided about the perils of overestimation to keep them motivated and working hard. Furthermore, depending on the degree of confidence exhibited by the students, an instructor may adopt several options to enhance the students' self-confidence and performance. This includes personalised instructional strategies based on a series of progressive small sequences of activities; specific, challenging, and attainable goal-settings that can be divided into short- and long-term goals; and feedback that emphasizes on process-related (or learning on measures such as effort and strategies) goals over outcome-related (or performance on measures such as number of tasks completed or scored) goals.

This research also shows that on easier questions (selection-type questions) individuals performed well and showed high confidence. Therefore, this type of questions are beneficial to be given to novice learners in introductory computer programming courses so as to maintain learners' interest and motivation. Lastly, machine learning showed potential in classifying participants based on their confidence levels and performance.

6.2 Limitations and Future Work

Several aspects were not addressed in this research work. The data was collected from one university in one country from 60 participants. Therefore, the results and discussions are limited to that dataset. This can be observed from the findings by the Pearson correlation coefficient test as the participants' scores were normally distributed contributing to higher numbers of average performers and leaving out very few participants in very low performers and very high performers group. A larger dataset could help combat this issue as more participants can be found in groups based on these two extremes - very high and very low performers. This problem continues even in the use of supervised machine learning where a larger dataset could have improved the classification accuracy. In the future, a larger dataset can be targeted containing about 200+ participants to observe the difference in the performance of supervised machine learning. The questions provided to the students also limit this research work to them and a different set of questions can be used including more types of questions to investigate further differences between question types and performance. Additionally, in the future, a contrast between the male and female students can be drawn to see the difference in their confidence levels in terms of their performance.

Ethical Standards

The authors assert that all procedures contributing to this work comply with the ethical standards of the relevant national and institutional committees on human experimentation and with the Helsinki Declaration of 1975, as revised in 2008.

Biographies

Zubair Ahsan received his M.Sc. degree from the Department of Artificial Intelligence, Faculty of Computer Science and Information Technology, University of Malaya in 2020. He is currently a research assistant at the same institute. His research interests include computer education, artificial intelligence, machine learning, and deep learning.

Unaizah Obaidellah received her Ph.D. in Cognitive Science from the University of Sussex in 2012. She is currently a Senior Lecturer in the Department of Artificial Intelligence, Faculty of Computer Science and Information Technology, University of Malaya. Her research interests are in the field of computing education, program comprehension, artificial intelligence, and eye-tracking studies on cognitive processes evaluations during learning and problem-solving.

Mahmoud Danaee is a senior lecturer at the UM Faculty of Medicine's Department of Social Preventive Medicine. As a statistician, he has taught statistics, experimental design, advanced statistical methods, and research methodology at the undergraduate and postgraduate levels in a variety of fields during the previous 24 years. His research interests are in the field advance statistical modelling and development scoring methods.

References

- [1] Z. Ahsan and U. Obaidellah, "Predicting Expertise Among Novice Programmers with Prior Knowledge on Programming Tasks," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2020, 1008–16.
- [2] M. Andrzejewska and A. Stolińska, "Comparing the Difficulty of Tasks using Eye Tracking Combined with Subjective and Behavioural Criteria," *Journal of Eye Movement Research*, 9(3), 2016, 1–16, <https://doi.org/10.16910/jemr.9.3.3>.
- [3] A. Bandura, *Social Foundations of Thought and Action: A Social Cognitive Theory*, Englewood Cliffs, NJ: Prentice-Hall, Inc, 1986.
- [4] R. Bednarik, C. Schulte, L. Budde, B. Heinemann, and H. Vrzakova, "Eye-movement Modeling Examples in Source Code Comprehension: A Classroom Study," in *Proceedings of the 18th Koli Calling International Conference on Computing Education Research (Koli Calling '18)*, November 2018, 1–8, <https://doi.org/10.1145/3279720.3279722>.

- [5] M. Daun, J. Brings, P. A. Obe, and V. Stenkova, “Reliability of Self-rated Experience and Confidence as Predictors for Students’ Performance in Software Engineering: Results from Multiple Controlled Experiments on Model Comprehension with Graduate and Undergraduate Students,” *Empirical Software Engineering*, 26(4), 2021, <https://doi.org/10.1007/s10664-021-09972-6>.
- [6] F. Detienne, “Reasoning from a Schema and from An Analog in Software Code Reuse,” in *Dans Fourth Workshop on Empirical Studies of Programmers*, 1991, 1–16, <http://arxiv.org/abs/cs/0701200>.
- [7] F. Detienne, *Software Design - Cognitive Aspects*, Springer-Verlag London Ltd, 2002.
- [8] F. Detienne, “Expert Programming Knowledge: A Schema-Based Approach,” in, 1990, 1–18.
- [9] F. D etienne and E. Soloway, “An Empirically-Derived Control Structure for the Process of Program Understanding,” *International Journal of Man-Machine Studies*, 33(3), 1990, 1–20, [https://doi.org/10.1016/S0020-7373\(05\)80122-1](https://doi.org/10.1016/S0020-7373(05)80122-1).
- [10] S. Doukakis, M. N. Giannakos, C. Koilias, and P. Vlamos, “Measuring Students’ Acceptance and Confidence in Algorithms and Programming: The Impact of Engagement with CS on Greek secondary education,” *Informatics in Education*, 12(2), 2013, 1–13, <https://doi.org/10.15388/infedu.2013.14>.
- [11] D. Druckman and R. A. Bjork, *Learning, Remembering, Believing: Enhancing Human Performance*, National Academy Press, 1994.
- [12] T. Fritz, A. Begel, S. C. M uller, S. Yigit-elliott, M. Z uger, and S. C. Muller, “Using Psycho-Physiological Measures to Assess Task Difficulty in Software Development,” in *Proceedings of the 36th International Conference on Software Engineering*, Vol. 2, 2014, 1–12, <https://doi.org/10.1109/ICSE.2015.284>.
- [13] S. A. Jessup, S. M. Willis, M. A. Lee, and G. M. Alarcon, “Using Eye-Tracking Data to Compare Differences in Code Comprehension and Code Perceptions between Expert and Novice Programmers,” in *Proceedings of the 54th Hawaii International Conference on System Sciences*, 2021, 1–10.
- [14] J. Kruger and D. Dunning, “Unskilled and Unaware of It: How Difficulties in Recognizing One’s Own Incompetence Lead to Inflated Self-Assessments,” *Journal of Personality and Social Psychology*, 77, 1999, 1121–34, DOI: [10.1037//0022-3514.77.6.1121](https://doi.org/10.1037//0022-3514.77.6.1121).
- [15] B. Landrum, “Examining Students’ Confidence to Learn Online, Self-Regulation Skills and Perceptions of Satisfaction and Usefulness of Online Classes,” *Online Learning*, 24(3), 2020, 128–46.

- [16] T. D. Latoza, D. Garlan, J. D. Herbsleb, and B. A. Myers, "Program Comprehension as Fact Finding," in *6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE 2007*, 2007, 361–70, <https://doi.org/10.1145/1287624.1287675>.
- [17] S. Lee, D. Hooshyar, H. Ji, K. Nam, and H. Lim, "Mining Biometric Data to Predict Programmer Expertise and Task Difficulty," *Cluster Computing*, 2017, 1–11, <https://doi.org/10.1007/s10586-017-0746-2>.
- [18] I. McChesney and R. Bond, "Eye Tracking Analysis of Computer Program Comprehension in Programmers with Dyslexia," in *Empirical Software Engineering*, Vol. 24, 2019, <https://doi.org/10.1007/s10664-018-9649-y>.
- [19] A. S. Najar, A. Mitrovic, and K. Neshatian, "Eye Tracking and Studying Examples: How Novices and Advanced Learners Study SQL Examples," *Journal of Computing and Information Technology*, 23(2), 2015, 1–20, <https://doi.org/10.2498/cit.100262>.
- [20] U. Obaidellah and M. A. Haek, "Evaluating Gender Difference on Algorithmic Problems using Eye-Tracker," in *Eye Tracking Research and Applications Symposium (ETRA)*, 2018, 1–8, <https://doi.org/10.1145/3204493.3204537>.
- [21] U. Obaidellah, M. Raschke, and T. Blascheck, "Classification of Strategies for Solving Programming Problems using AoI," in *Eye Tracking Research and Applications (ETRA)*, 2019, 1–10, <https://doi.org/10.1145/3314111.3319825>.
- [22] J. Prather, B. A. Becker, M. Craig, P. Denny, D. Loksa, and L. Margulieux, "What Do We Think We Think We Are Doing? Metacognition and Selfregulation in Programming," in *Proceedings of the 2020 ACM Conference on International Computing Education Research*, 2020, 2–13.
- [23] R. S. Rist, "Schema Creation in Programming," *Cognitive Science*, 13(3), 1989, 1–26, https://doi.org/10.1207/s15516709cog1303_3.
- [24] B. Sharif and J. I. Maletic, "An Empirical Study on the Comprehension of Stereotyped UML Class Diagram Layouts," in *IEEE International Conference on Program Comprehension*, 2009, 268–72, <https://doi.org/10.1109/ICPC.2009.5090055>.
- [25] E. Soloway, "Learning to Program = Learning to Construct Mechanisms," *Communications of the ACM*, 29(9), 1986, 1–9.
- [26] E. Soloway, J. Bonar, and K. Ehrlich, "Cognitive Strategies and Looping Constructs: An Empirical Study," *Communications of the ACM*, 26(11), 1983, 1–8, <https://doi.org/10.1145/182.35843>.
- [27] E. Soloway and K. Ehrlich, "Empirical Studies of Programming Knowledge.pdf," in *IEEE Transactions on Software Engineering*, Vol. 10, 1984, 1–15.