Predicting expertise among novice programmers with prior knowledge on programming tasks

Zubair Ahsan* and Unaizah Obaidellah†
Dept of Artificial Intelligence
Faculty of Computer Science and Information Technology
University Malaya, Kuala Lumpur, Malaysia
*E-mail: woa180022@siswa.um.edu.my
†E-mail: unaizah@um.edu.my

Abstract— **The studies on program comprehension have seen developments over the years from the cognitive science perspective. As eye-tracking technology has proven to analyze visual attention and gaze-performance, it has then been largely used in the program comprehension studies to help understand the underlying cognitive processes among the participants. In this research work, we conducted an experiment using common fundamental programming questions on 66 undergraduate computer science students to study the gaze-behavior among the high and low-performing participants on programming comprehension. We aim to better understand the differences in the time taken by the individuals in terms of their performance with existing prior knowledge and use machine learning to predict their expertise. Findings from this study suggest that mental schemas do play a role as the high performers demonstrated less time taken to attempt the questions than the low performers and machine learning algorithms were able to successfully predict their expertise. The conclusions drawn are supported by eye-tracking metrics across individual- and group-levels.**

Keywords: novice programmers, eye-tracking, classification, machine learning, problem-solving

I. INTRODUCTION

The increased need for many individuals to understand and produce computer programs has made it necessary to improve the general understanding of programming [1]. This may produce better instructional design and teaching strategies.

Algorithmic problem solving requires a set of iterative process involving identifying the problems, defining solution strategies, references to the necessary subject matter knowledge, and relate problems to a familiar experience. Drawing on the Mental Schema Theory, several studies have investigated the effect of previous knowledge on programming success [2][3]. A basic concept stored in the memory represented by a data structure is called a Mental Schema. Therefore, from this viewpoint, understanding a program is through the evocation of these schemas [4]. Mental schema has two main aspects; knowledge to assist program understanding (declarative) and cognitive mechanism to use the knowledge (procedural) [5]. Thus, an individual will use prior knowledge to produce a solution plan to solve a specific programming problem. This helps individuals solve a problem quicker as the solution plan is formed quickly as compared to those individuals who have zero or less prior knowledge. Thus, the more organized the knowledge is in an individual's mind the more quickly they are likely to solve a programming problem. However, empirical evidence is required to support the effect of schemas on the individual's performance.

However, [6] emphasized that the inability of individuals to correctly establish analogies with past problems due to misunderstood deep-lying concepts makes it difficult for them to solve new problems after evoking the schemas. Therefore, a wide variety exists in the performance levels of the students. Other factors in varying performance levels, especially in a classroom, include personal traits as well as experience that influence the undertaken strategies to solve problems [7].

Eye-tracking technology allows for the assessment of an individual's reading behavior and empirically specifies their underlying cognitive processes [8]. Therefore, eye-tracking technology can help examine and study the individual's visual attention and mental schemas when attempting to solve programming questions. This type of eye movement assessment may inform about an individual's level of expertise based on the exhibited programming comprehension abilities. Eye-tracking metrics such as the total fixation duration may inform about an individual's problem-solving capability as they attempt programming questions. The total fixation duration is different from the total task duration as it only considers the time taken on certain areas of the stimulus that the participant gazes upon (see II. Related Work, Section D, Data Preparation). These certain areas are the crucial lines of code that the participants must gaze upon to attempt the question. This allows the narrowing down of the exact time taken by the participants as the time taken represented in the form of total fixation duration is the time that the participants actually take to solve a programming question. Previously, eye movement data analysis has been done through different algorithms, however, machine learning approaches to classify these detections are still new and in need of further investigation [9].

This work aims to classify novice programmers based on accuracy and total fixation duration using machine learning methods. Furthermore, individual's performance and their total fixation duration will be analyzed using statistical tests. This research work seeks to contribute an eye-tracking study to examine mental schemas and visual attention while using machine learning algorithms and statistical tests for analysis.

II. Related Work

Early works in programming comprehension emphasized psychological aspects [10]. This refers to how an individual cognitively processes information when attempting programming tasks. The study by [1] used the term "cognitive fit" and then was extended upon with another hypothesis as seen in [11] that introduced the terms "plan-like" and "unplan-like" for programs. Plan-like programs are text-book based programs which are more commonly taught in the educational institutes whereas unplan-like programs are although correct but are written differently from what is usually taught and presented. These studies were focused upon the structure of the questions and whether or not these questions correlate with the prior knowledge that individuals possess in their minds. [12] explored the problem-solving strategy taken by the individuals as they attempt the programming questions. It was concluded that individuals while learning (creating a schema), use a backward and bottom-up approach, but while retrieving (using the schema), use a top-down and forward approach. [4] conducted a study to see how experts (advanced students) use schemas to solve programming problems and found that only one strategy is often insufficient to solve a problem due to unstructured schemas. [5] used mental schema theory as a framework and with its findings, supported by the eye-tracking data of the participants, suggested that the prior knowledge is indeed helpful to the students when solving new problems. This provides room for further exploration, as a better understanding of how students comprehend and perform on the programming questions that they have previously attempted but represented in different formats could provide more insights in programming comprehension and problem-solving strategies domain. This research attempts to investigate this further, as it would help teaching, learning, and overall instructional design of program languages.

Given the recent development of machine learning approaches in data analysis, adopting such methods on eye-movement data is promising. For example, [13] and [9] employed machine learning algorithms to detect events such as fixations and saccades from the eye-tracking data. Furthermore, [14] and [15] strongly suggest that machine learning algorithms can be used to classify the differences in eye-gaze patterns. [16] conducted a study to gather data from the participants using a combination of psycho-physiological devices and used machine learning to classify their emotions while attempting the questions. These studies use a variety of machine learning techniques and produce significant enough results that set a foundation for this research work to use similar machine learning techniques to analyze and classify the eye-tracking data.

A closer inspection of the literature indicated that investigation on areas related to programming comprehension, eye-tracking, and machine learning is rather meager, as this kind of work seems to be highly collaborative. [17] conducted a study that used eye-tracking technology as well as electroencephalography (EEG) to predict programmer expertise (novice/expert) and task difficulty (easy/difficult). The study was conducted on 38 programmers (novices and experts) who were required to perform 23 basic comprehension tasks, and the results were analyzed using a supervised machine learning algorithm; Support Vector Machine (SVM), and evaluated using F-measure and Cross-Validation. The analysis found that eye tracker predicted task difficulty with 63.3% precision and 66.4% recall while an EEG predicted with 60.8% precision and 66.4% recall and eye tracker predicted programmer expertise with 88.7% precision and 92.1% recall while EEG predicted with 94.2% precision and 91.1% recall [17]. Therefore, either device is enough in experiments of this nature (program comprehension) as they both provided similar prediction percentages using a machine learning algorithm. A similar study was conducted by [18] that incorporated the use of eye-tracking, electrodermal activity, and electroencephalography on the subjects, and machine learning to determine task difficulty among the subjects.

These studies further support how machine learning techniques can be used on eye-tracking data. However, it is worth noting that [17] and [18] did not incorporate mental schema theory as their theoretical framework. It is important to consider the effect of mental schemas on an individual's performance on programming questions as it would make it easier to understand where they struggle and what measures can be taken to improve their learning. Therefore, drawing on this limitation, the proposed research incorporates mental schema theory as its theoretical framework (see III. Methodology, Section B. Materials) to carry out an investigation on program comprehension using eye-tracking technology via machine learning assessments.

III. Methodology

The experiment presented in this study aims to answer the following research questions:

1. What groups can be classified from the students' performance?
2. How fast and accurately do students (high vs low performing) identify code mechanics that have different representations?
3. How well can the identified machine learning algorithm classify the participants?

A. Participants

Participants were recruited from a large public university in Asia. They were first-year undergraduate computer science students major. The students needed to have undertaken or presently registered for the Fundamentals of Programming course as a pre-requisite for participation eligibility at the time of the study. Hence, their expertise levels were determined by their performance of the course evaluated via formal assessments defined by the course. A total of 66 students took part in this study, male and female alike, who had normal or corrected to normal vision. The participants were between 18 and 25 years of age (M_{age} 19.3 years, SD_{age} = 0.57) and were all considered as novices as they are all

beginners. However, the data from 6 students were excluded due to poor recordings.

B. Materials

The stimuli contained a total of twelve questions that were all randomly presented to the participants. These questions were equally divided into two types namely Selection and Iteration, and each of these types involved two questions of each level of difficulty - easy, medium, and hard. The easy questions contain a simple and straightforward piece of short code with one main statement of selection or iteration (only for loops). The medium questions contain a more sophisticated code, by using two selection statements or while loops. The hard questions contain more sophistication using nested statements for selection and iteration question. Figure 1 shows an example of the stimulus. Each question consists of a problem statement and four code snippets, which became the Areas of Interest (AOIs) – invisible to the participants. The code snippets contain at least two correct answers (max four) where one is plan-like and the others are unplan-like. The questions were designed with an assumption that any piece of code that produces similar output would evoke the corresponding previous knowledge as the chosen participants were familiar with the presented questions. Indirectly, this informs about the nature of the network of concepts or mental schema possessed by a problem solver. Thus, a similar amount of visual attention could be exhibited. A locally hosted web-application was developed to present the stimuli to the students. The Tobii X2-30C eye-tracking device was used to capture the eye movements of the participants. Tobii Studio was used to collect the eye-gaze data.

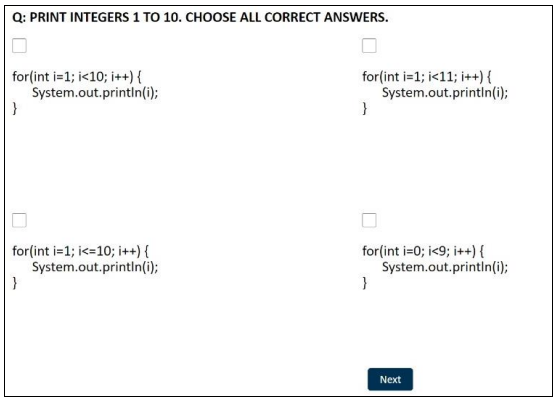

Figure 1 Example of an easy Iteration stimulus.

C. Procedure

The participants came to a dedicated lab space and attempted all twelve questions at a scheduled time. The eye-tracking device was adjusted according to their sitting positions. This was followed by a 5-point calibration. After that, participants were left to read the instructions, watched a short video about the task, and performed an example task before they began answering the 12 questions. The questions were randomized for every participant to avoid order effects. After each question, the participants were asked to rate their confidence level and difficulty level for each question. At the end of the session, the participants completed a post-survey questionnaire to indicate their demographic and programming experience. The participants were told to take as much time as they need, however, they were encouraged to attempt the questions to their best ability as quickly as possible. Each participant received a course or extra credit for their participation.

D. Proposed Analysis

To classify the participants into groups, machine learning algorithms are considered. These machine learning algorithms are later validated. Furthermore, statistical tests are utilized to analyze how fast and accurately do participants identify the same codes represented differently. This section proposes the analysis of the Total Fixation Duration (TFD) of the participants in terms of scores between the high and low performers. Total Fixation Duration is the sum of the duration of all fixations that participants make in each Area of Interest across all stimuli. Hence, it is considered the most appropriate eye-tracking metric for this section as it accurately represents the time taken by the students on the stimuli in the respective Areas of Interest (AOIs). The aim is to categorize and relate the Total Fixation Duration of the participants with their performance on the stimuli.

DATA PREPARATION

The data to be used for analysis consists of Total Fixation Duration from a total of 5 Areas of Interest (AOIs) that were drawn on each stimulus on Tobii Studio and the answers for each stimulus by each participant. The Total Fixation Duration (TFD) is an eye-tracking metric that represents the accumulated duration of all fixations that a participant has made with their eyes on the stimuli. In this case, the fixations were included from each of the five areas of interest drawn prior to the analysis. The unit for the Total Fixation Duration (time) is second. The correct answers, on the other hand, were tallied to devise the total score of the participants on the stimuli. Partially correct answers by the participants are considered as wrong answers. For example, if a participant chooses one right answer and the other wrong, their response will be considered as wrong. In the case of three right options, if the participants choose two right options and one wrong, their response will still be considered as wrong. A provided answer is only considered correct if it matches exactly those defined by the experimenter.

MACHINE LEARNING

A. Unsupervised Learning

The k-means cluster analysis was considered appropriate because it gave a quick overview by clustering the data into an optimal number of clusters using the elbow method. As for research question 1, the requirement for this analysis is to cluster different types of students based on their total fixation duration across all stimuli and their score (performance on the stimuli). It is expected that there shall be four clusters that compare the total fixation duration with performance levels, namely; Short Time-Low Performer (SL), Long Time-Low

Performer (LL), Short Time-High Performer (SH), and Long Time-High Performer (LH).

B. Supervised Learning

As described in III. Methodology, section A, the participants recruited for the study were computer science, first-year undergraduate students, from a university in Asia. The grading scheme adopted by the university was used to determine the participants' performance levels, as shown in Table 1.

Table 1 Grading Scheme

Range	Grade	Category
80-100	A	Distinction
75-79	A-	
70-74	B+	Good
65-69	B	
60-64	B-	
55-59	C+	Pass
50-54	C	
45-49	C-	Conditional Pass
40-44	D+	Fail
35-39	D	
00-34	F	

The dataset was divided into training and testing set with 60% and 40% of the total data respectively, which means that 36 participants' data was labeled manually and put in MATLAB's classification learner. The remaining 24 participants' data were predicted by the training model created and exported from within MATLAB. The classifiers used by MATLAB in the training were Tree (Fine), Support Vector Machine (Quadratic), KNN (Fine). Tree (Fine) selected as a decision tree classifier is a tree in which branches are labeled by features that were used in training the classifier. The decision tree classifier then uses its labeled branches to predict the testing data. Support Vector Machine (Quadratic) is selected as Support Vector Machine (SVM) is a binary classifier, however, the quadratic variant allows for the maximization of its margins. KNN (Fine) is selected for its simplicity as it uses Euclidean distance between two points for classification. As there are four desired groups tree classifier is expected to perform better than SVM and KNN, however, they are included for comparison. For validation, F1-accuracy will be calculated.

Statistical Tests

The statistical analysis will be carried out in three parts; 1) an Independent Samples t-test will be carried out using accumulated Total Fixation Duration (TFD) (mean) and Score (mean) of High and Low Performers, 2) an Independent Samples t-Test will be carried out using accumulated TFD (mean) in Stimuli Difficulty between High and Low Performers, 3) a Repeated Measures ANOVA will be carried out using accumulated TFD (mean) across Areas of Interest (AOIs) among all Stimulus.

IV. Results

A. Unsupervised Learning

Figure 2 describes the clusters produced by k-means cluster analysis and Figure 3 shows the elbow method used to find the optimal number of clusters. As shown in Figure 3, an optimal value of k=4 was acquired from the K-means Cluster analysis in Figure 2 on the data containing all participants' performance on the stimuli and the total fixation duration mean of each participant across all stimuli. The desired clusters for this data set would categorize the participants into four major clusters - Short Time Low Performance, Short Time High Performance, Long Time Low Performance, and Long Time High Performance. However, this was not the case with the unsupervised learning method. It is worth noting that the clusters, as they were based on Euclidean distance, display a clear difference between cluster 3 and cluster 4 as can be seen in Figure 2. It can be seen in Figure 2 that cluster 4 has a Total Fixation Duration Mean greater than 13 seconds, whereas cluster 3 has a Total Fixation Duration Mean less than 12 seconds. It can also be seen that the cluster 1 and 2 also have a Total Fixation Duration Mean less than 12 second. This highlights a gap between the students taking more time on average to attempt the questions and the students taking less time on average to attempt the questions.

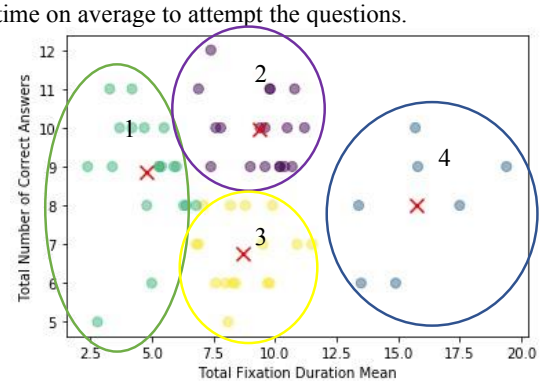

Figure 2 K-means Cluster Analysis. Each color represents a different cluster.

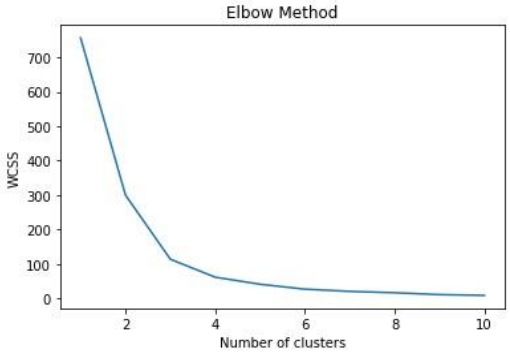

Figure 3 The Elbow Method

Given the inaccurate representation of clusters using the unsupervised learning approach, the focus of analysis shifted to the supervised method.

B. *Supervised Learning*

Using the grading scheme of the university where the data was collected, Table 2 was produced and it shows that 34 students from a total of 60 received an A- and above which is 75% or more correct answers out of 12 questions. These 34 students were then labeled as "High Performers" whereas the remaining students who received 8 correct answers or less were labeled as "Low Performers".

Table 2 Performance of the Participants on the Stimuli

No. Students	Correct answers	Percentage of 12	Grade	Category	Performance
2	5	41%	D+	Fail	Low Performers
9	6	50%	C	Pass	
5	7	58%	C+		
10	8	66%	B	Good	
16	9	75%	A-	Distinction	High Performers
11	10	83%	A		
6	11	91%	A		
1	12	100%	A		

Note. Percentage of 12 (Column 3) is the percentage calculated by dividing the correct number of answers (Column 2) by the total number of questions 12.

As the stimuli were designed to find the effect of the mental schema (see III. Methodology, Section B. Materials), it is expected at a shorter time taken on average to attempt and solve these problems, the participants potentially had well-structured mental schemas. As the majority of the participants took less time than 12 seconds as seen in the K means Cluster Analysis in Figure 2, we decided to label this majority of participants as "Short Time", whereas the rest that took longer than 13 seconds were labeled as "Long Time".

Determining an appropriate threshold to separate the high and low performers from the total fixation duration mean (accumulated mean of all stimuli) participants took to complete the tasks is subjective as it depends on the type of participants who are involved in the study, the kind of stimulus that is presented to the participants, and the structure of the mental schemas the participants possess. In this particular case, the participants were expected to have well-structured schemas as the stimuli were already taught to them during the semester the data was collected, and the participants were all novice programmers in the first year of their undergraduate computer science program. Therefore, it is assumed that participants who spent less time but scored high accuracy (i.e. 9 out of 12 correct answers), potentially possess a structured network of mental schema with good relations between concepts.

The classifiers used by MATLAB in the training were Tree (Fine), Support Vector Machine (Quadratic), and KNN (Fine) as mentioned in III. Methodology, section D. The accuracies of these classifiers ranged from 83.3% (KNN - Fine) to 94.4% (Tree - Fine), see Table 3.

Table 3 Accuracies of the Machine Learning Classifiers on the Training Set

Classifier	Accuracy
Tree (Fine)	**94.4%**
SVM (Quadratic)	86.1%
KNN (Fine)	83.3%

For validation, the F1-accuracy of the classifiers was calculated and can be seen in Table 4. As there are 4 classes in the model and each class did not contain the same number of participants, hence, weighted F1-accuracy is calculated.

Table 4 F1-Accuracy of the Selected Classifiers

Classifier	Weighted F1-Accuracy	Weighted Precision	Weighted Recall
Tree (Fine)	**93.7%**	**89.5%**	**93.1%**
SVM (Quadratic)	83.7%	82.5%	86.2%
KNN (Fine)	82.1%	81.3%	83.1%

In consequence, Tree (Fine), which showed the highest accuracy as well as highest Weighted F1-Accuracy, Weighted Precision, and Weight Recall, was trained as a classifier. The predictors used for this approach are Total Fixation Duration Mean and Total Number of Correct Answers. The training model is shown in Figure 4, and the prediction model can be seen in Figure 5.

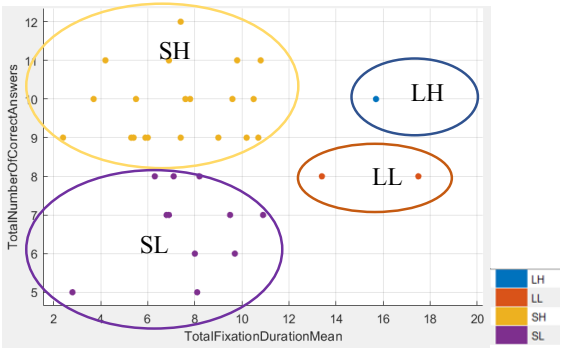

Figure 4 Training Model (SH:22, SL:11, LH:1, LL:2)

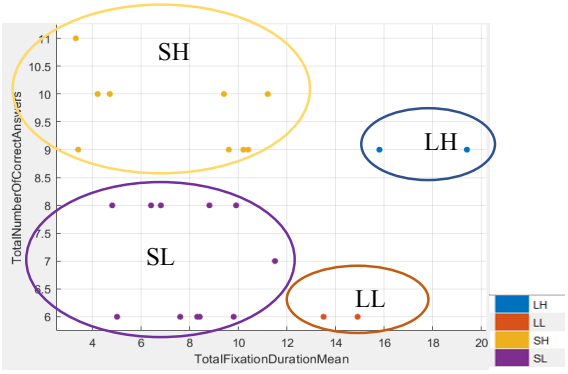

Figure 5 Prediction on Testing Set

To better understand the effect of previous knowledge in relation to correctly attempting the tasks and the duration to

attempt these tasks, the following section will discuss about Total Fixation Duration Mean and Total Number of Correct Answers Mean.

C. *Total Fixation Duration and Score*

In this subsection, Total Fixation Duration Mean and Total Number of Correct Answers Mean, both accumulated of high and low performers separately, are used to draw comparisons between high and low performers. To further support this, a statistical approach has been adopted.

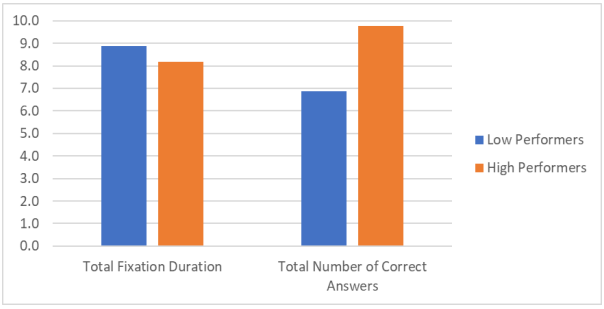

Figure 6 Aggregated Fixation Duration and Total Score of the High and Low performers

Figure 6 shows the aggregated Total Fixation Duration for low performers, which is almost 9, and for high performers, which is slightly higher than 8 across all stimuli. Furthermore, Figure 6 shows the Total Number of Correct Answers (mean) across all stimulus for low performers, which is at 7, and for high performers, which is almost 10.

An independent-samples t-test conducted to compare total fixation duration (TFD) did not show a significant difference between the low performers (M=8.836, SD=3.2650) and high performers (M=8.079, SD=3.8472); t(58)=0.806, p=0.424. This suggests that the amount of attention invested in the stimuli between these two groups was of no difference.

To further inspect the relation between the performance (high and low) and the total fixation duration, further steps were taken by comparing the total fixation duration mean between high and low performers in terms of stimuli difficulty. Based on the following analysis, it is expected that the high performers will spend less time (have a lower total fixation duration) as compared to low performers across all stimuli.

D. *Total Fixation Duration (TFD) in Stimuli Difficulty between High and Low Performers*

In this section, the mean of total fixation duration for high and low performers was calculated for each stimulus. There are two types of questions; Iteration and Selection, each involving two questions of each level of difficulty; easy, medium, and hard, as mentioned in III. Methodology, Section B. The comparison drawn in this section will be among high and low performers between two questions of the same type and the same difficulty i.e. Iteration (I) type; question Easy 1 (E1) and Easy 2 (E2), Medium 1 (M1) and Medium 2 (M2), Hard 1 (H1) and Hard 2 (H2), see Figure 7 and similarly for the Selection (S) type questions. This comparison will be followed by an independent-samples t-test to compare the total fixation duration (mean) for each stimulus between the high and low performers.

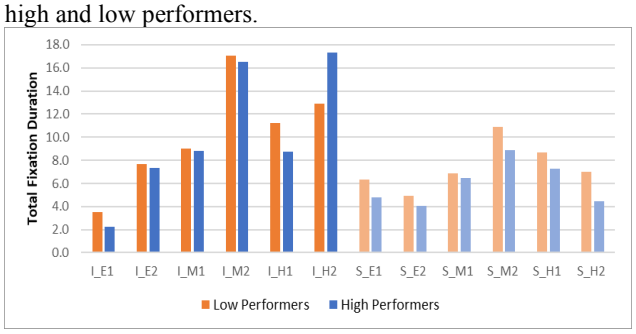

Figure 7 Accumulated TFD Mean Across Stimulus. Darker tones are for Iteration type questions; lighter tones are for Selection type questions.

An independent-samples t-test was conducted to compare the total fixation duration (TFD) between high performers and low performers for each stimulus separately. For I_E1, the independent-samples t-test showed a significant difference between the low performers (M=3.521, SD=2.9767) and high performers (M=2.268, SD=1.2055); t(58)=2.231, p=0.030. In the comparison for S_H2, a significant difference is also shown between the low performers (M=7.007, SD=3.5613) and high performers (M=4.486, SD=2.3181); t(58)= 3.315, p=0.002. The independent-samples t-test did not show a significant difference for the remaining stimuli.

E. *Total Fixation Duration (TFD) across Areas of Interest (AOIs) among all Stimulus*

As each stimulus contained 5 Areas of Interest (AOIs); Choice 1 (C1), Choice 2 (C2), Choice 3 (C3), Choice 4 (C4), and Problem Statement (PS), all of them for each respective stimulus are represented against their Total Fixation Duration Mean among all participants in Figure 8.

Figure 9 represents the Total Fixation Duration (mean) for high performers across all Areas of Interest (AOIs) of each stimulus and Figure 10 represents the Total Fixation Duration (mean) for low performers across all Areas of Interest (AOIs) of each stimulus. These two figures also highlight an anomalistic comparison between 4 AOIs between high and low performers.

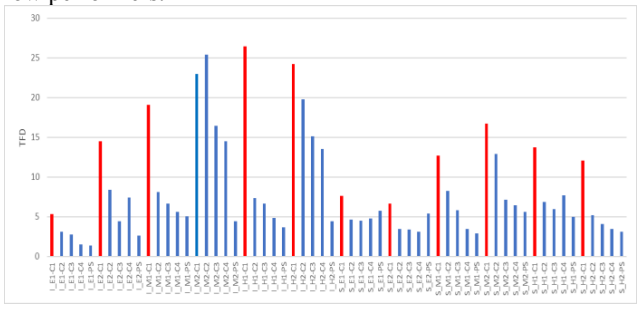

Figure 8 Comparison of TFD Mean Across all AOIs of Stimuli (bars in red color represent the highest TFD at C1 for all stimulus except I_M2)

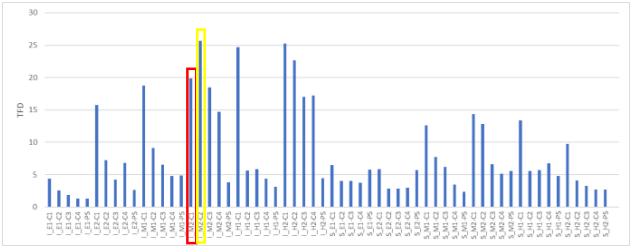

Figure 9 Mean High Performers Across all AOIs of Stimuli (the highlighted bars in red and yellow color represent the comparison between AOIs of Choice 1 and Choice 2 for I_M2)

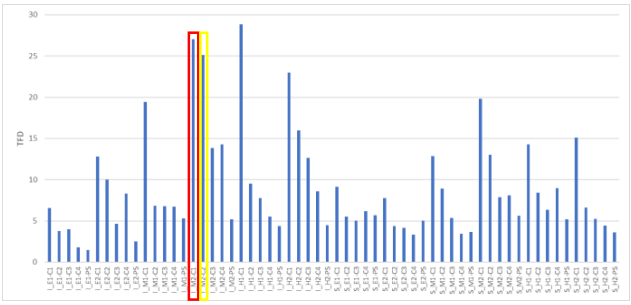

Figure 10 Mean Low Performers Across all AOIs of Stimuli (the highlighted bars in red and yellow color represent the comparison between AOIs of Choice 1 and Choice 2 for I_M2)

A repeated measures ANOVA was run to investigate if there were significant differences between the stimuli, their levels of difficulty and AOIs. Findings showed that Type differed significantly between Level 1 (Iteration) and Level 2 (Selection) $F (1, 59) = 65.645$, $p < .001$]. Furthermore, it showed that Difficulty also differed significantly for both Iteration and Selection between Level 1 (Easy 1) and Level 2 (Easy 2) $F (1, 59) = 14.837$, $p < .001$], between Level 3 (Medium 1) and Level 4 (Medium 2) $F (1, 59) = 51.514$, $p < .001$], but not significantly between Level 5 (Hard 1) and Level 6 (Hard 2) $F (1, 59) = 6.866$, $p > .001$]. Finally, ANOVA showed that AOI differed significantly between Level 1 (Choice 1) and Level 2 (Choice 2) $F (1, 59) = 116.255$, $p < .001$], between Level 2 (Choice 2) and Level 3 (Choice 3) $F (1, 59) = 45.618$, $p < .001$], but not significantly between Level 3 (Choice 3) and Level 4 (Choice 4) $F (1, 59) = 4.082$, $p > .001$].

V. DISCUSSION

A. Classifying Participants into Groups

As research question 1 is to classify the participants, the results found in IV. Analysis and Results, Section A Supervised Learning suggests that three students who took a long time but still managed a high number of correct answers (LH) can be described as students who may exhibit a better problem-solving ability; however, they could potentially possess less robust mental schema as suggested by the amount of the time taken to attempt and solve these questions. An alternative explanation is that the LH students may take time to revisit their solution strategies as a form of validating their answers. On the other hand, four students who took longer time and scored lower in the tasks (LL) can be considered to possess even less structured schemas, and therefore, it affects their problem-solving ability. Thirty-one students who took shorter time and performed higher (SH) may possess more structured schemas as the time taken is reduced considerably and can be considered skillful at solving problems. Twenty-two students who took short time and performed low (SL) could simply be disinterested and tried to quickly finish the experiment and hence, took less time and performed lower. Those who had a total of 8 correct answers are debatable, as they can be considered as average students with somewhat structured schemas enough to speed up the solution process but could be influenced by other factors such as anxiety to affect their performance. This claim needs further exploration.

B. Prior Knowledge affects Performance and Duration (TFD)

As research question 2 is about the performance and the duration of high and low performing students, the results found in IV. Results, section C suggests that low performers took a slightly longer time while high performers took less time in completing the task. This enforces the mental schema theory that high performers take less time to attempt problems due to potentially well-structured units of knowledge supported by stronger conceptual relations in the mental schema. However, an independent t-test of aggregated total fixation duration mean between high and low performers did not show a significant difference between the score of high and low performers.

The results were found in IV. Results, section D reveal that low performers had a higher total fixation duration on average as compared to the high performers among all iteration questions except for H2. This finding indicates that the low performers could possess limited knowledge (declarative) causing them to have a less structured schema which ultimately makes the access to the right information weaker (procedural) and hence, taking them a long time due to disjoint pieces of knowledge or concepts. On the contrary, high performers could possess adequate knowledge causing them to have a well-structured schema allowing their access to the right information stronger and faster. Hence, taking them a short amount of time due to structured and coherent pieces of knowledge. A potential reason why high performers took longer for H2 than low performers could be that in iteration questions there is an increased need for validation through the loops to ensure that the considered code statements produce correct results causing the students to spend longer time to analyze the codes. Here it can be noted that due to the higher difficulty of the question, even the high performers struggled, as out of the 60 participants' data that is being analyzed, only 28 of them answered correctly on Iteration H2.

Accordingly, in Selection type questions, a similar pattern is observed as in Iteration type questions among the low and high performers in terms of total fixation duration. As

expected, low performing students took longer time and high performing students took shorter time. It is to be noted that since Selection type questions would potentially require easier processing in identifying the logic of the codes in binary form, it generally takes both sets of performers to take less time as compared to when they attempt Iteration type questions. Both sets of participants showed the Total Fixation Duration Mean of about 10 seconds and almost reaching 17 seconds for the Iteration type questions with the exception for Iteration E1 that has a Total Fixation Duration Mean of less than 4 seconds for all performers. On the other hand, both sets of participants showed a Total Fixation Duration Mean of less than 10 seconds at most for Selection type questions. This suggests that selection type questions generally require less cognitive effort as compared to iteration type questions. An independent samples t-test showed a comparison of total fixation duration (TFD) between high performers and low performers for each stimulus separately. Out of all 12 stimuli, only two of them showed a significant difference; I_E1 (Iteration_Easy1) and S_H2 (Selection_Hard2).

The results found in IV. Results, section E indicate that all stimuli except for I_M2 have the highest Total Fixation Duration Mean for C1 (Choice 1). This means that all participants naturally inspected code in Choice 1 to match the desired outcome that they have formed by reading the problem statement. Therefore, for Choice 1, a long time was taken to analyze it, consequently, they only seemed to verify all the other choices with the first choice. By further investigating why I_M2 was different than the rest of the stimuli it is noted that high performers have spent slightly more time on C2 than C1 of I_M2 (Iteration Medium 2) whereas the low performers spent slightly more time on C1 than C2. However, the TFD for both choices are higher than C3 and C4 of the same question for high and low performers. This is due to the reason that for this particular stimulus, the correct choices are C2 and C3 but the reason for TFD on C1 still being considerably high is that they naturally went to C1 to compute the desired outcome as given to them in the problem statement (PS) but upon failure, they moved to C2 to analyze. Once they recognized that C2 is the desired outcome, C3 and C4 were only to be verified by C2, and therefore, both have a lower TFD.

A repeated-measures ANOVA was run to investigate if there were significant differences between the stimuli, their levels of difficulty, and AOIs. It showed that Type differed significantly between Level 1 (Iteration) and Level 2 (Selection). This means that the nature of the questions was presented well enough in the given stimuli. Furthermore, it showed that Difficulty also differed significantly for both Iteration and Selection between Level 1 (Easy 1) and Level 2 (Easy 2) between Level 3 (Medium 1) and Level 4 (Medium 2), but not significantly between Level 5 (Hard 1) and Level 6 (Hard 2). This suggests that the harder questions were considered somewhat equally difficult for Iteration and Selection questions. Finally, ANOVA showed that AOI differed significantly between Level 1 (Choice 1) and Level 2 (Choice 2), between Level 2 (Choice 2) and Level 3 (Choice 3), but not significantly between Level 3 (Choice 3) and Level 4 (Choice 4). Although this finding suggests that the amount of attention spent on each choice is generally different, the finding should be interpreted with care as the positioning of the correct codes were randomly positioned between the questions.

To answer research question 2, high performers take less time as they seem to possess more organized mental schemas, and hence, they fixate less on the stimulus and quickly attempt the question, whereas low performers take more time to attempt the questions as they seem to possess less organized mental schemas, and hence, they fixate more on the stimulus and spend more time on the question. This is consistent with the results found in [4], [5] in terms of the duration that the high and low performers take to solve programming problems.

C. Performance of Machine Learning Classifiers

As research question 3 is about the performance of the selected machine learning classifiers on the data used in this research, the results shown in IV. Results, section B suggest that all selected machine learning algorithms performed well in classifying the participants. The best among them, however, was Tree (Fine) and it classified the participants with a 94.4% accuracy, with 93.7% weight F1-accuracy, 89.5% weighted precision, and 93.1% weighted recall.

VI. CONCLUSION

A. Implications

Findings from this research demonstrate that the overall organization of information in an individual's mind seems to influence the time taken to perform algorithmic programming tasks. This is tested under several measures, namely, difficulty levels and AOIs to further understand the behavior and the effect of organized information in an individual's mind. This research also suggests that higher ability problem solvers who take less time may seem to possess more organized mental schemas is reflected in their ability to fixate less on the stimulus and quickly attempt the question. On the other hand, lower ability problem solvers who usually took more time to attempt the questions may seem to possess less organized mental schemas as they fixate more, and hence, spend more time on the questions. Furthermore, machine learning showed some potential to classify novices and experts based on their total fixation duration mean and the total number of correct answers they achieved. This further implies the relationship between the two features.

B. Limitations

Several aspects are not addressed in this research, and hence, become the limitations of this research work. The data that was collected and analyzed was taken from one university in one country and from 60 participants. Hence, the presented results and discussions are limited to that data set. Other aspects that limit this study include the questions that

were designed to be the stimuli and the predefined difficulty level assigned to them.

The analysis section involved the categorization of the participants into high and low performers, whereas there could have easily been an "average performers" group. However, the high and low performers' categories were opted for due to their large use in similar studies and for drawing simpler and more distinct comparisons between the two.

C. Future Work

The limitations outlined are in a way research gaps that can be fulfilled with further research. A different dataset can be used to verify and validate the findings of this study. Different questions can be designed to increase the cognitive load on the participants which may lead to different results. Accordingly, the average performers' group can be introduced as a middle category between high and low performers to understand the reading strategies and patterns of the individuals in more depth.

ACKNOWLEDGMENT

The authors would like to thank the study participants for their voluntary participation and the reviewers for their suggestions.

REFERENCES

[1] E. Soloway, J. Bonar, and K. Ehrlich, "Cognitive Strategies and Looping Constructs: An Empirical Study," *Commun. ACM*, vol. 26, no. 11, pp. 853–860, 1983.

[2] M. Andrzejewska and A. Stolińska, "Comparing the difficulty of tasks using eye tracking combined with subjective and behavioural criteria," *J. Eye Mov. Res.*, vol. 9, no. 3, pp. 1–16, 2016.

[3] U. Obaidellah and M. Al Haek, "Evaluating gender difference on algorithmic problems using eye-tracker," *Eye Track. Res. Appl. Symp.*, 2018.

[4] F. Détienne and E. Soloway, "An empirically-derived control structure for the process of program understanding," *Int. J. Man. Mach. Stud.*, vol. 33, no. 3, pp. 323–342, 1990.

[5] U. Obaidellah, M. Raschke, and T. Blascheck, "Classification of strategies for solving programming problems using AoI," *ETRA*, 2019.

[6] A. Gomes and A. J. Mendes, "Learning to program - difficulties and solutions," *Int. Conf. Eng. Educ.*, pp. 1–5, 2007.

[7] K. Sharma, M. Katerina, and T. Halvard, "Evidence for Programming Strategies in University Coding Exercises," in *European Conference on Technology Enhanced Learning*, 2018, pp. 326–339.

[8] L. Meng Lung *et al.*, "A review of using eye-tracking technology in exploring learning from 2000 to 2012," *Educ. Res. Rev.*, vol. 10, no. 88, pp. 90–115, 2013.

[9] R. Zemblys, D. C. Niehorster, O. Komogortsev, and K. Holmqvist, "Using machine learning to detect events in eye-tracking data," *Behav. Res. Methods*, vol. 50, no. 1, pp. 160–181, 2018.

[10] F. Detienne, *Software Design - Cognitive Aspects*. Springer-Verlag London Ltd., 2002.

[11] E. Soloway and K. Ehrlich, "Empirical studies of programming knowledge.pdf," *Ieee*, vol. 10, no. 5. pp. 595–520, 1984.

[12] R. S. Rist, "Schema creation in programming," *Cogn. Sci.*, vol. 13, no. 3, pp. 389–414, 1989.

[13] R. Zemblys, "Eye-movement event detection meets machine learning," *Biomed. Eng. (NY).*, no. November, pp. 98–101, 2016.

[14] K. W. Cho *et al.*, "Gaze-Wasserstein: A quantitative screening approach to autism spectrum disorders," *2016 IEEE Wirel. Heal. WH 2016*, pp. 14–21, 2016.

[15] K. A. Dalrymple, M. Jiang, Q. Zhao, and J. T. Elison, "Machine learning accurately classifies age of toddlers based on eye tracking," *Sci. Rep.*, vol. 9, no. 1, pp. 1–10, 2019.

[16] S. C. Müller and T. Fritz, "Stuck and frustrated or in flow and happy: Sensing developers' emotions and progress," *Proc. - Int. Conf. Softw. Eng.*, vol. 1, no. May, pp. 688–699, 2015.

[17] S. Lee, D. Hooshyar, H. Ji, K. Nam, and H. Lim, "Mining biometric data to predict programmer expertise and task difficulty," *Cluster Comput.*, pp. 1–11, 2017.

[18] T. Fritz, A. Begel, S. C. Müller, S. Yigit-elliott, and M. Züger, "Using Psycho-Physiological Measures to Assess Task Difficulty in Software Development," 2014.